frontiers | Research Topics

From batch-size 1 to serial production: Adaptive robots for scalable and flexible production systems

Edited by
Mohamad Bdiwi, Arvid Hellmich, Steffen Ihlenfeldt and Andreas Mueller

Published in
Frontiers in Robotics and AI

ISSN 1664-8714
ISBN 978-2-8325-2392-6
DOI 10.3389/978-2-8325-2392-6

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

From batch-size 1 to serial production: Adaptive robots for scalable and flexible production systems

Topic editors

Mohamad Bdiwi — Fraunhofer Institute for Machine Tools and Forming Technology (FHG), Germany
Arvid Hellmich — Fraunhofer Institute for Machine Tools and Forming Technology (FHG), Germany
Steffen Ihlenfeldt — Fraunhofer Institute for Machine Tools and Forming Technology (FHG), Germany
Andreas Mueller — Johannes Kepler University of Linz, Austria

Citation

Bdiwi, M., Hellmich, A., Ihlenfeldt, S., Mueller, A., eds. (2023). *From batch-size 1 to serial production: Adaptive robots for scalable and flexible production systems*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-2392-6

Table of contents

frontiers | Frontiers in Robotics and AI

TYPE Editorial
PUBLISHED 26 April 2023
DOI 10.3389/frobt.2023.1201488

Check for updates

OPEN ACCESS

EDITED AND REVIEWED BY
Kostas J. Kyriakopoulos,
National Technical University of Athens,
Greece

*CORRESPONDENCE
Mohamad Bdiwi,
mohamad.bdiwi@iwu.fraunhofer.de

RECEIVED 06 April 2023
ACCEPTED 17 April 2023
PUBLISHED 26 April 2023

CITATION
Bdiwi M and Ihlenfeldt S (2023), Editorial: From batch-size 1 to serial production: adaptive robots for scalable and flexible production systems.
Front. Robot. AI 10:1201488.
doi: 10.3389/frobt.2023.1201488

Editorial: From batch-size 1 to serial production: adaptive robots for scalable and flexible production systems

Mohamad Bdiwi* and Steffen Ihlenfeldt

Fraunhofer Institute for Machine Tools and Forming Technology (FHG), Chemnitz, Germany

KEYWORDS

agile manufacturing and industry 4.0, from simulation to real robot applications safety and virtual commissioning, flexible robotics for handiwork, AI and perception, robot control

Editorial on the Research Topic
From batch-size 1 to serial production: adaptive robots for scalable and flexible production systems

The first manuscript with the title “Halim et al.” proposes a no-code approach to programming industrial robots. The proposed method relies on a finite state machine with three layers of natural interactions based on hand gestures, finger gestures, and voice recognition. The results obtained from the experiments indicate the capability of this novel approach for real-world deployment in an industrial context. In a similar vein, however, on the controller level, the third manuscript “Wiese et al.” presents a skill-based approach as an abstract template class methodically for modularization of the assets in the control and parameterizable skills. An orchestration system is used to call the skills with the corresponding parameter set and combine them into automated process sequences. This approach provides a more flexible control, as operators can independently adapt and expand the automated process sequence without modifying the controller code.

In the second manuscript, “Bdiwi et al.” the authors propose a dynamic safety-related finite-state machine for safe transitions between various collaborative operation modes dynamically and adequately. In addition to that, the collaborative operation modes are grouped in different clusters and categorized at various levels systematically. The proposed approach is integrated into a new dynamic risk assessment tool as a promising solution toward a new safety horizon in line with Industry 4.0.

To enable the production of large components using industrial robots, the fourth manuscript “Schnellhardt et al.” presents a novel approach to segmented manufacturing. The proposed segmentation strategy divides the part into segments whose structural design is adapted to the capabilities of the field components available on the shop floor. The process planning step of each segment is automated by utilizing the similarity of the segments and the self-description of the corresponding field component. The result is a transformation of a batch size one production into an automated quasi-serial production of the segments. Moreover, the fifth manuscript with the title “Wabner et al.” proposes technological cooperation of industrial robots and machine tools to improve flexibility and efficiency in parts production. The approach results in a novel type of collaborative manufacturing equipment for matrix production that will improve the versatility, efficiency, and profitability of production. By enhancing machine tools with additional manufacturing technologies, a robot can beneficially support workpiece machining.

Regarding shop floor management, modern manufacturing objectives such as automation, mass customization, self-organization, and smart factories require intelligent control approaches. The sixth manuscript "Bahrpeyma and Reichelt" presents a review (MARL) as an effective approach to handling uncertainties due to the dynamic nature of the environment. It has been demonstrated how different aspects of smart factories match the objectives and capabilities of MARL and suggested a mapping from smart factory features to the equivalent concepts in MARL, indicating how MARL provides an appropriate solution to provide almost all the required features in the smart factory at once. In terms of robot path planning, most motion planners generate low-level control inputs, leading to geometric and temporal deviations between the executed and planned motions of the robot. To solve this challenge, the work in the seventh manuscript "Hou et al." proposed a new approach using neural networks. This approach uses realistic collision-free trajectories to simultaneously learn high-level motion commands and robot dynamics, generating trajectories that can be executed directly by the robot control system. The proposed approach has significantly reduced geometric and temporal deviation between the executed and planned motions and generates new collision-free trajectories up to ten times faster than benchmark motion planners.

The last two manuscripts discuss innovative approaches to improve industrial human-robot collaboration using intelligent sensor systems with a focus on safety and efficiency. The eighth manuscript with the title "Krusche et al." proposes a novel approach for automatic annotation of human actions in 3D point clouds using various DNN classifiers, an intuitive GUI, and a methodology for automatic sequence matching. The proposed framework was evaluated in an industrial use case and was shown to accelerate the annotation process by 5.2 times through automation. This approach can save time and resources while improving human-robot collaboration by recognizing, analyzing, and modeling human actions. The last manuscript with the title "Rashid et al." presents flexible and efficient local and global sensing using cameras and LiDAR. The proposed methodology incorporates a local 3D sensor on the robot body and formulates occlusion due to the robot body, which ensures minimum occlusion in the robot workspace. The resulting system enables high robot velocities while providing flexibility and safety with heavy-duty industrial robots. The proposed approach aims to have a minimum scalable sensor concept and adjust it according to the process requirements.

Author contributions

MB: main author SI: Review.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Robotics and AI

TYPE Brief Research Report
PUBLISHED 29 September 2022
DOI 10.3389/frobt.2022.1014476

Flexible skill-based control for robot cells in manufacturing

Torben Wiese*†, Johannes Abicht†, Christian Friedrich, Arvid Hellmich and Steffen Ihlenfeldt

IIoT Controls and Technical Cybernetics, Fraunhofer Institute for Machine Tools and Forming Technology, Dresden, Germany

Check for updates

OPEN ACCESS

EDITED BY
Khoshnam Shojaei,
Islamic Azad University of Najafabad, Iran

REVIEWED BY
ALireza Izadbakhsh,
Islamic Azad University of Garmsar, Iran
Omid Elhaki,
Islamic Azad University of Najafabad, Iran

*CORRESPONDENCE
Torben Wiese,
torben.wiese@iwu.fraunhofer.de

†These authors contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 08 August 2022
ACCEPTED 16 September 2022
PUBLISHED 29 September 2022

CITATION
Wiese T, Abicht J, Friedrich C, Hellmich A and Ihlenfeldt S (2022), Flexible skill-based control for robot cells in manufacturing.
Front. Robot. AI 9:1014476.
doi: 10.3389/frobt.2022.1014476

Decreasing batch sizes lead to an increasing demand for flexible automation systems in manufacturing industries. Robot cells are one solution for automating manufacturing tasks more flexibly. Besides the ongoing unifications in the hardware components, the controllers are still programmed application specifically and non-uniform. Only specialized experts can reconfigure and reprogram the controllers when process changes occur. To provide a more flexible control, this paper presents a new method for programming flexible skill-based controls for robot cells. In comparison to the common programming in logic controllers, operators independently adapt and expand the automated process sequence without modifying the controller code. For a high flexibility, the paper summarizes the software requirements in terms of an extensibility, flexible usability, configurability, and reusability of the control. Therefore, the skill-based control introduces a modularization of the assets in the control and parameterizable skills as abstract template class methodically. An orchestration system is used to call the skills with the corresponding parameter set and combine them into automated process sequences. A mobile flexible robot cell is used for the validation of the skill-based control architecture. Finally, the main benefits and limitations of the concept are discussed and future challenges of flexible skill-based controls for robot cells are provided.

KEYWORDS

skill-based control, flexible control systems, robot cells, modular automation, robot skills

1 Introduction: Current challenges of controls for robot cells

In the manufacturing industry, robots offer a productive and flexible solution to automate manufacturing processes. Due to their serial design and uniform mechanical interfaces, robots are used as manipulators for variable, repetitive and high-precision tasks (Arents and Greitans, 2022). Typical applications include basic handling applications (e.g., for parts, pallets) as well as more complex processes, such as welding or the assembly of parts (Siciliano and Khatib, 2016). To add the necessary skills to the robot, assets, like grippers, sensors, and actuators, are applied to form task-specific static or mobile robot cells (Lienenluke et al., 2018 - 2018; Wojtynek et al., 2019; Sanneman et al., 2020).

In mass production, e.g., the automotive industry, such robotic cells are common. However, the increasing number of product variants requires more flexible robot cells in hardware and software to adapt them to the current processes (Dorofeev and Wenger, 2019 - 2019; Saukkoriipi et al., 2020). Therefore, the retooling and reconfiguration of robot cells are key challenges for the current research (Jörgen Frohm et al., 2006; T. Dietz, 2012). In hardware, there exist standardized mechanical, electrical, and data interfaces for modular assets to enable flexibility (Radanovic et al., 2021 - 2021). This is known as Plug and Produce concept (Pfrommer et al., 2015; Wojtynek et al., 2019; Falkowski et al., 2020). In software, configuring and teaching robot controllers (RC) and programmable logic controllers (PLC) is still a non-uniform, time-consuming and skill-demanding bottleneck (Sanneman et al., 2020; Zhou et al., 2020). An expert with control programming knowledge is necessary to reconfigure the robot cell PLC. Operators with basic process knowledge are not able to adjust the control software effortless. Therefore, programming experts must define all possible process changes in software that limits the flexibility to serval static case clauses (Deutschmann et al., 2020). In addition, the monolithic programming of controllers, non-uniform interfaces and static graphical user interfaces further decrease the software flexibility in robot cell controls. Hence, the standardization of communication interfaces and more abstract, task-oriented programming becomes very important to increase the software flexibility of robot cells (Saukkoriipi et al., 2020; Heimann and Guhl, 2020 - 2020; Sanneman et al., 2020). As a result, the following requirements for flexible controls of robot cells are defined:

- Extensibility: To be able to adapt a robot cell to changing processes, it must be possible to extend it with adapted or new assets to be able to use their manufacturing functions for the process sequences. Besides the hardware connectors, the extensibility must be ensured in terms of software. The control architecture, therefore, must deal with real-time capabilities, computing power, and communication interfaces of the control systems of the assets.
- Flexible usability: The individual manufacturing functions of the assets must be flexibly usable for the operator. To ensure flexibility, each asset should provide its functions independently of other assets to combine them independently into sequences. By defining automated sequences, the operator assembles the functions into more complex process steps.
- Configurability: The control of flexible robot cells must enable a configurability of the automated sequences to the operator based on his detailed manufacturing process knowledge. The individual functions of the assets must be configurable *via* changeable parameters to be able to adapt them to specific process steps. This allows the operator to configurate sequences with differently parameterized function calls of the assets without having control programming expertise.
- Reusability: Already defined functions and sequences should be reusable to reduce reprogramming and increase commissioning time. In this way, the operator can access already working process steps and generate new process sequences without having to adapt individual functions. Process steps can also be exchanged and reused between different robot cells with the same functionalities.

One promising approach to fulfilling the requirements of a flexible robot cell control is the skill-based control architecture (SBC) (Dorofeev and Wenger, 2019 - 2019). The SBC uses an abstraction concept by composing single manufacturing tasks through parameterizable, component-specific skills. An orchestration layer manages the task-specific arrangement and call of the skills. Each component offers its skills *via* uniform software interfaces for data communication (Pfrommer et al., 2014 - 2014). Beside other self-describing component modelling approaches according to the Industry-4.0 concept, OPC-UA is commonly used as a universal communication interface (Zimmermann et al., 2019 - 2019). Not only for combining skills but also for unified and flexible multi-system orchestration, SBC together with OPC-UA enables immense benefits in software implementation and reconfiguration (Profanter et al., 2019). In SkillPro (Brandenbourger and Durand, 2018 - 2018), RAZER (Steinmetz et al., 2018), and other projects (Saukkoriipi et al., 2020), the successful implementation has been validated. The VDI/VDE has published the first standardizations of skills in a guideline in field of process industry. This guideline focuses on modularization, the service interfaces, parametrization, state machines, and behavior models (Deutsches Institut fur Normung, 2020). Today, main deficits are:

- Despite the increasing efforts in standardization and tests, SBCs are not widely used in the manufacturing industry, compared to established monolithically programmed control systems.
- Unified models for components and skills for manufacturing processes are missing (Malakuti et al., 2018 - 2018).
- Control systems are only programmed by experts. New concepts need to simplify control programming for non-experts (Pedersen et al., 2016).

This paper presents an approach of a skill-based control for flexible robot cells for manufacturing. Therefore, the approach proposes a control architecture that fulfils the requirements of

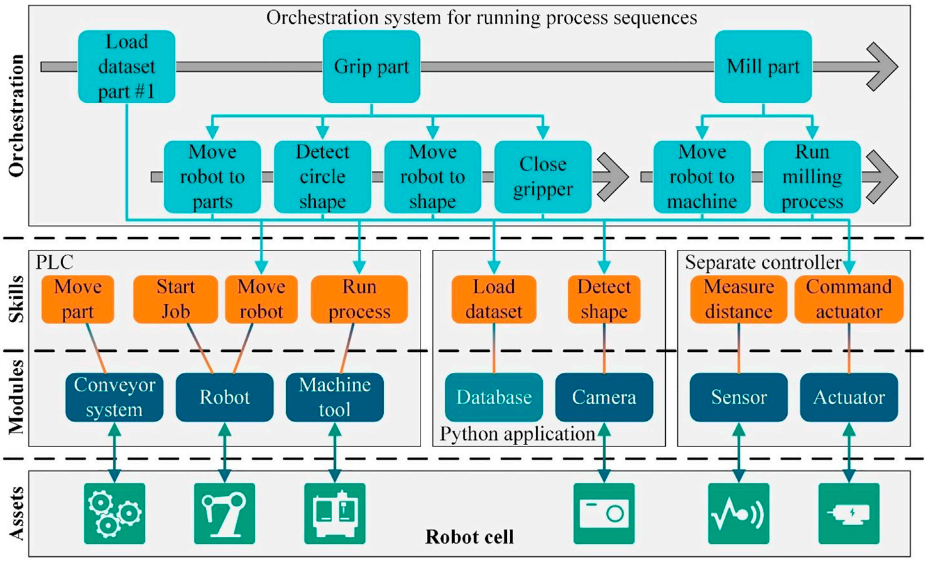


FIGURE 1
Concept of a robot cell automation by a skill-based control architecture.

extensibility, flexible usability, configurability, and reusability. The verification of requirements is analysed on a flexible robot cell for machine tool automation.

2 Method: Development of a skill-based control for flexible robot cells

The development of the SBC divides into three methodical subgoals of the control software. The order of the subgoals represents the workflow during implementation on the robot cell controller. First, all assets are modularized, followed by assigning the functions of the assets to the modules as parameterizable skills. Finally, the SBC is extended by an orchestration system of the skills to automate the skills into process sequences. In the following section, the subgoals are presented.

Modularization starts by dividing the assets of a robot cell into functionally separable subsystems that work and are controlled independently. Therefore, object-oriented programming ensures uniform states and interfaces. In the SBC, a superclass as a template for a unified asset module is defined. New asset modules are thus created by inheritance of the template module. This approach enables consistent handling, monitoring, state control, and error management of all the different modules in a robot cell.

Figure 1 illustrates the linking of assets and their corresponding software modules in different control systems of the robot cell. Depending on the controller architecture, module controllers can also run in different controllers or applications as long as the communication and linking with the module handling is realized. Beneficially, the specific requirements for asset controls in terms of necessary real-time capability, hardware connectivity, and computing performance can be considered and implemented individually. This allows the decentralized allocation of control tasks to performance-specific, separated controllers which reduces hardware costs. Modules can also be arranged hierarchically at different levels and consist of different sub-modules to consider the physical linking of assets in the controller. The communication between the modules of different controllers is realized *via* various manufacturer- and programming language-independent interfaces, such as OPC-UA. The modularization of controls for all assets enables the extensibility of the robot cell at the software level. New assets and their control modules can be integrated *via* uniform interfaces through template inheritance.

To provide the asset functions, such as “move” of a robot or “close” of a gripper, parameterizable skills for modules are defined, as shown in Figure 1. In programming, the bottom-up approach can be used to implement the available functions for each asset as skills in the module control. The control programmer should implement not only the asset functions that are necessary for the overall automation solution, but also the functions that the asset can perform independently of other assets. This guarantees the flexible usability of all functionalities of the assets. Another important aspect is the possibility to parameterize the skills to adapt the individual asset functions to different process tasks. For example, the target position can be specified as a parameter for a robot movement to enable configurability by the operator.

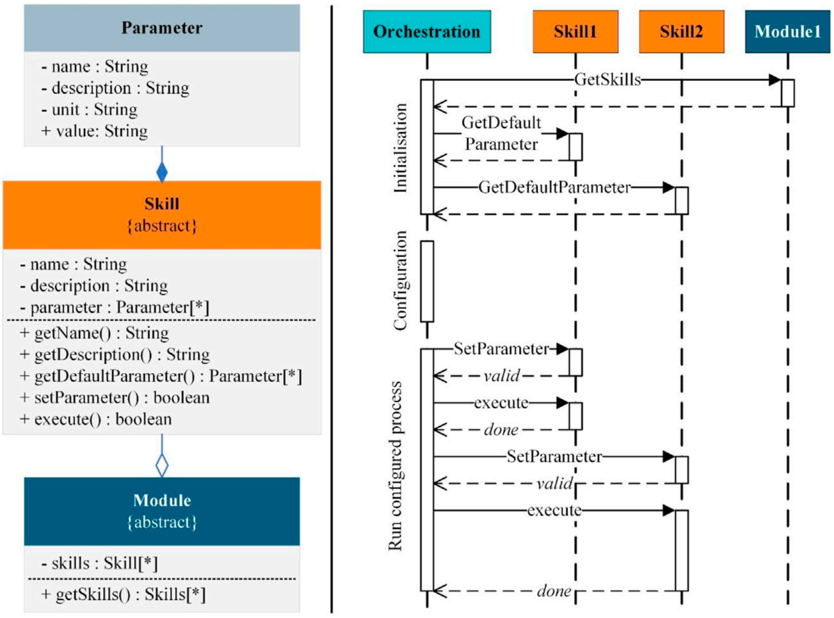


FIGURE 2
Class diagram for the implementation of parameterizable skills in modules (left) and sequence diagram for their orchestration (right).

Modules and their skills represent the basic functions of the various assets in a robot cell. To combine them into an automated process sequence, e.g., a machine tool tending process, an orchestration system in the master controller is required. The orchestration system defines, parameterizes, and controls the process sequence. Due to the modularization, the orchestration system can communicate with all available module skills using different communication interfaces like both vendor-specific and vendor-independent ones. Complex process sequences can be configured by parameterizing and combining skills into reusable steps, whereby the operator can flexibly change individual parameters of skills or entire steps at any time. When creating automated process sequences, the top-down approach is ideal to generate detailed sub-steps in various abstraction levels. The operator can use his detailed knowledge of the manufacturing process and first create abstract process steps, which are then specified in further sub-steps and finally call up the individual skills with configured parameters. The orchestration level in Figure 1 shows an exemplary process flow with abstract steps, which in turn contain more concrete sub-steps. Reusability is ensured by storing the sequences and steps in data lists that contain the information about the skill connection and associated parameters. With a suitable human machine interface (HMI), the operator can configure and parameterize the process sequences without programming, which means that no knowledge of programming in the controls is required.

For the communication between skills and the orchestration system or operator, each skill provides the necessary meta information about itself, such as its name, description, and the associated asset as well as the information about its adjustable parameters. The left side of Figure 2 presents an exemplary class diagram of the abstract skill class with the necessary parameters and methods for implementation in control systems. Every skill deriving from the abstract class can be connected in the same way, by accessing properties and using methods providing the information and control options. To be able to parameterize the skills uniformly, self-describing parameters using a generalized definition structure are introduced. The orchestration system can use the methods to retrieve the default parameters, set new parameter values as well as to execute the skill that triggers the associated function of the asset with the specified parameters. After the configuration and parameterization of a process sequence, the steps can be processed *via* the unified interfaces to the skill and thus an automated flow can be accomplished. The right side of Figure 2 shows a sequence diagram for the exemplary execution of a process sequence by an orchestration system.

3 Results: Verification of the flexibility of the skill-based control on a mobile robot cell

To evaluate the proposed SBC in terms of the flexibility requirements, a test platform has been selected. Therefore, the SBC was implemented on the Robo Operator© (RO) using a

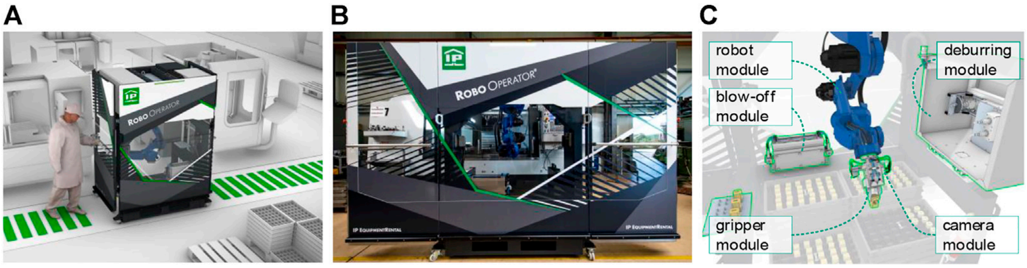
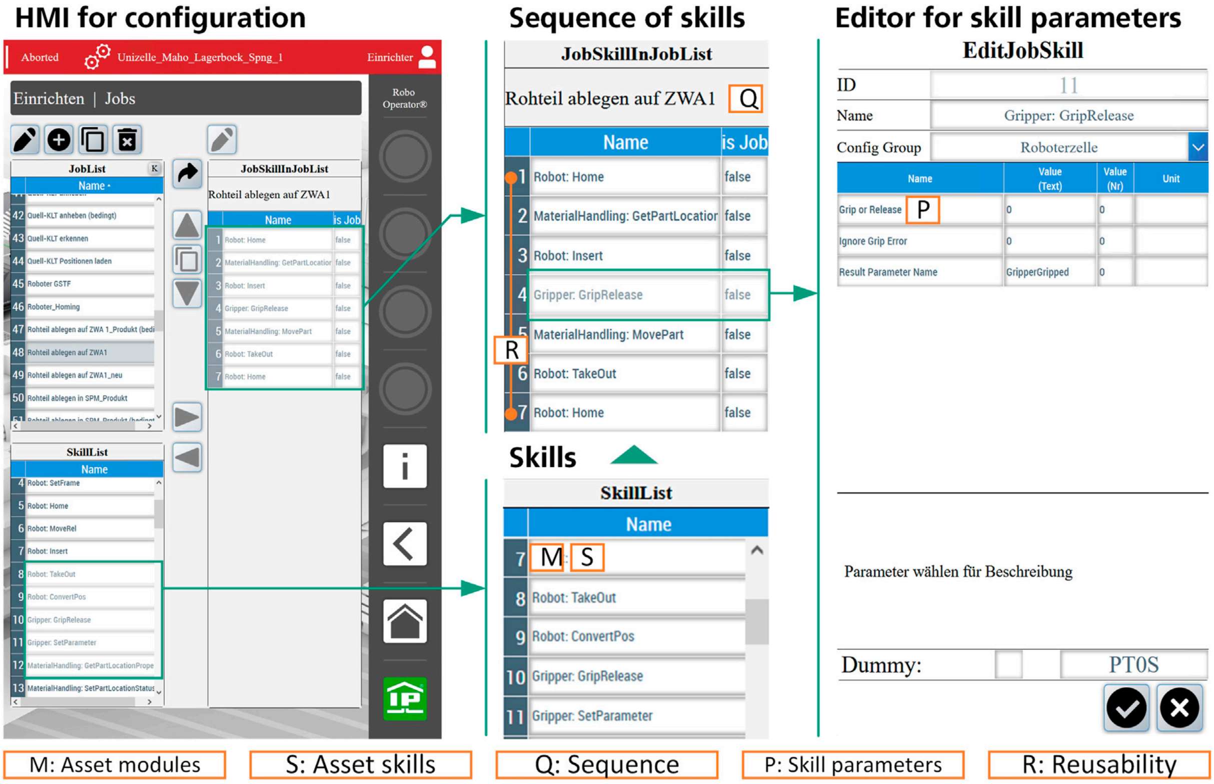


FIGURE 3
(A–C): The Robo Operator © automates operator tasks on machine tools by skills.



Aborted Unizelle_Maho_Lagerbock_Spng_1 Einrichter

Einrichten | Jobs

Robo Operator®

JobList

	Name
42	Quell-KLT anheben (bedingt)
43	Quell-KLT erkennen
44	Quell-KLT Positionen laden
45	Roboter GSTF
46	Roboter_Homing
47	Rohteil ablegen auf ZWA 1_Produkt (bedi
48	Rohteil ablegen auf ZWA1
49	Rohteil ablegen auf ZWA1_neu
50	Rohteil ablegen in SPM_Produkt

SkillList

	Name
4	Robot: SetFrame
5	Robot: Home
6	Robot: MoveRel
7	Robot: Insert
8	Robot: TakeOut
9	Robot: ConvertPos
10	Gripper: GripRelease
11	Gripper: SetParameter
12	MaterialHandling: GetPartLocationPrope
13	MaterialHandling: SetPartLocationStatus

JobSkillInJobList

Rohteil ablegen auf ZWA1

	Name	is Job
1	Robot: Home	false
2	MaterialHandling: GetPartLocatior	false
3	Robot: Insert	false
4	Gripper: GripRelease	false
5	MaterialHandling: MovePart	false
6	Robot: TakeOut	false
7	Robot: Home	false

SkillList

	Name
7	M S
8	Robot: TakeOut
9	Robot: ConvertPos
10	Gripper: GripRelease
11	Gripper: SetParameter

EditJobSkill

ID	11
Name	Gripper: GripRelease
Config Group	Roboterzelle

Name	Value (Text)	Value (Nr)	Unit
Grip or Release	0	0	
Ignore Grip Error	0	0	
Result Parameter Name	GripperGripped	0	

Parameter wählen für Beschreibung

Dummy: PT0S

FIGURE 4
Composition of the module skills to sequences and skill parameterization through the configuration HMI.

TwinCAT PLC. The RO is a mobile robot cell, as shown in Figure 3 (A-C), that was developed in a research project between Fraunhofer IWU and Industrie-Partner GmbH (Abicht et al., 2021). As flexible automation solution, it automates tasks of operators on machine tools. Therefore, an industrial robot (Yaskawa GP12) with a 2-jaw gripper and a smart camera (Intel Realsense D435i) enable the RO to move parts, to open and close doors, and to start the machine tool by control panel interaction. The smart camera provides the position information of all relevant objects. Applicable asset-modules, e.g., deburring or blow-off modules, extend the workflow with new skills for the manufacturing process, see Figure 3C.

With the given structure, the RO represents a flexible robot cell in manufacturing. The flexibility of hardware is reached by standardized Han® connectors and the capsuled design of the asset modules. In the following section, the paper analyses how the methods of the proposed SBC architecture realize the flexibility in the control. Therefore, the paper discusses the implementation based on the four aspects from chapter 1. The human machine interface (HMI) of the RO visualizes the

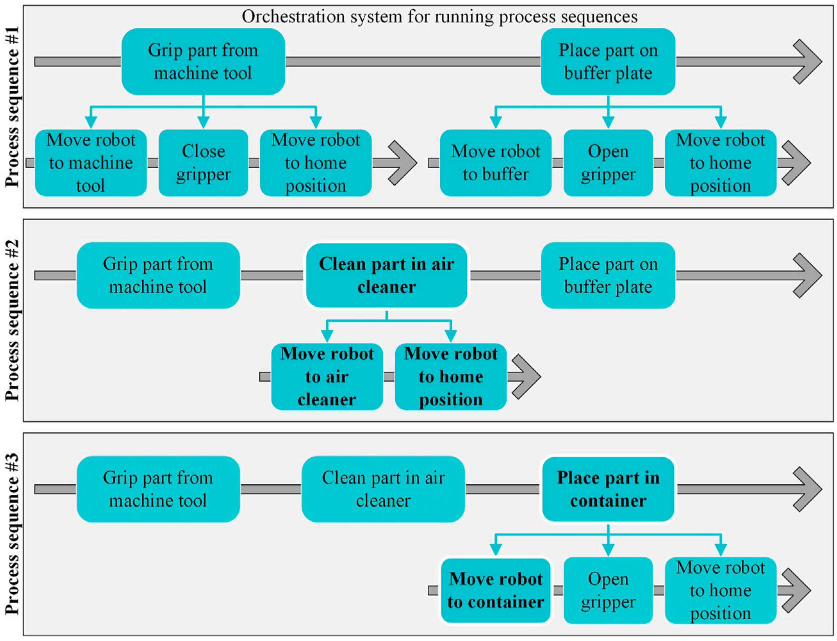

Orchestration system for running process sequences
Process sequence #1
Grip part from machine tool
Place part on buffer plate
Move robot to machine tool
Close gripper
Move robot to home position
Move robot to buffer
Open gripper
Move robot to home position
Process sequence #2
Grip part from machine tool
Clean part in air cleaner
Place part on buffer plate
Move robot to air cleaner
Move robot to home position
Process sequence #3
Grip part from machine tool
Clean part in air cleaner
Place part in container
Move robot to container
Open gripper
Move robot to home position

FIGURE 5
Example for modifying an automated process sequence of a robot cell with skill-based control.

achieved results, see Figure 4. At the HMI, operators configure the process through the composition of skills into sequences.

While implementing the SBC into the RO, all assets get an own functionally separated software module. As shown in Figure 4, e.g., the asset modules (M) "Robot" for the Yaskawa GP12 and "Gripper" for the 2-jaw gripper are programmed. Other modules, such as the camera, blow-off and, deburring module, could be created by inheritance of the template module easily to verify the extensibility of the software. Based on the template, other modules can be integrated into the process in this way. Due to the TwinCAT EtherCAT Hot-Connect functionality, the modules are initialized automatically when the corresponding assets are plugged by Han® connectors. A state machine was implemented to manage the module states, such as Idle, Error, or Resetting, while automating the process. The communication between the modules is based on TwinCAT ADS (Automation Device Specification) as universal software interface.

The definition of skills for all asset modules guarantees the flexible usability of the software inside the master control system, see the asset skills (S) in Figure 4. To achieve a specific goal or sub-task of a manufacturing process, the skills are combined into a sequence of skills (Q). For example, the sequence "Place raw part on buffer plate" (in Figure 4: "Rohteil ablegen auf ZWA1") places a gripped raw part on a buffer plate to determine the position of the part more accurately. Physically, the RO inserts the gripped part from a home position of the robot to the buffer plate position, opens the gripper and returns to the home position safely. In the HMI, the operator combines the skills Home, Insert, and TakeOut from the robot module with a GripRelease skill from the gripper module from the SkillList, as shown in Figure 4.

To use the module skills for different processes in a flexible way, they must fulfil the configurability requirement. In the skill editor in Figure 4, the operator adjusts the skills to the current requirements of the process by specific skill parameters (P). For example, operators must configure if the gripper should be opened or closed at a grip or release position. By inheriting the skill template class from Figure 2, the GripRelease-Skill is defined with the skill-specific parameters Grip or Release and additional parameters that configure the skill in the RO control to the current task without changing PLC code. The orchestration module executes every skill with the configured skill parameters while automating the complete process sequence.

Predefined sequences can also be used as steps in a higher-level sequence to reduce the configuration effort. This allows an operator without detailed knowledge of single sequence steps to configurate a sequence for the RO. For example, a simple process sequence automating the move of one part from a machine tool to a buffer plate, as shown by sequence #1 in Figure 5, can be extended with an intermediate step to clean the part with a blow-off module. For this, the operator only has to insert the necessary process step in the orchestration system, which contains the parameterized skills calls (see Sequence #2 in Figure 5). Furthermore, the operator can change the already existing step for placing the part by a new destination, such as a container. The operator can do this by adjusting the parameterization of the skill that makes the robot move to a specific position, in this case by changing the target position from “buffer” to “container” (see Sequence #3 in Figure 4).

The combination of parameterized skills and pre-defined sub-sequences to new sequences enables a high reusability. In Figure 4 the “Place raw part on buffer plate” sequence uses more than one Home-Skill (R). To save the configuration in the HMI, the SBC of the RO has an additional data base module to save and reload all the information about the skills and parameters as well as the sequences itself.

4 Conclusion

The present paper proposes a method for implementing a flexible control architecture in robot cells called skill-based control. In chapter 1, four requirements were defined for developing the method for implementing the skill-based control. Main reasons for developing this method are missing guidelines or unifications for flexible control architectures for manufacturing purposes and the wide range of programmer expertise levels that must be conducted.

Methodically, the skill-based controls consist of a software modularization of all assets, definition of capsuled asset functions in skills and an orchestration system for skill management and calling. Based on object-oriented programming, template classes have been implemented for the asset modules and skills. For an extension of an automation, the templates can be used to easily create specific modules. Thus, the communication structure of the modules is unified. As communication protocols, OPC-UA or similar universal manufacturer-independent standardizations are proposed. To adapt the skills to the current situation, each skill call be individualized by parameters defined by the skill developer.

The results of the paper show that the skill-based control fulfils all requirements of a flexible control for robot cells. For verification of the methods, the skill-based control was successfully implemented on a mobile robot cell. The implementation shows the fast programming through the reusability of the software components in the human machine interface. Furthermore, the programming of the software is reduced to the combination of skills and process steps to sequences on a non-programming-level that is potentially less time-consuming than static programming.

As the method was used to implement the skill-based control for the RO, it quickly becomes clear that the flexibility available to the operator depends primarily on the type and amount of provided skills and their parameters. For high flexibility, many skills and adjustable parameters are needed, requiring a longer development time. To reduce the resulting complexity, the operator must be offered predefined process steps that combine frequently used skill combinations and their parameters. Furthermore, dependencies between skills that may not be known to the operator must be represented in the orchestration system. Knowledge about the dependencies of the skills and process steps is crucial for the configuration of fault-free process sequences. Therefore, a critical development goal is to further reduce the expertise required in the use of skill-based controls mainly by expanding the orchestration system.

For further studies, the interoperability of the software modules on different master controllers and their corresponding programming languages must be conducted. Because the flexible usability, configurability and reusability of skills and sequences depends mainly on the usability of the HMI, so more research in HMI design and layout is necessary. Therefore, the intuitiveness, modularity, uniformity, security, and robustness must be considered. Finally, it must be researched how suitable the proposed skill-based control fits to larger production lines or matrix production systems.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

Conceptualization and methodology, TW and CF; software implementation and verification, TW and JA; writing and drafts, TW and JA; reviewing and editing, CF and JA; supervision, CF, AH and SI; funding acquisition, AH, JA and SI. All authors have read and agreed to the published version of the manuscript.

Funding

The results of the paper were funded by the European Fund for Regional Development (EFRE) under grants from the Sächsisches Staatsministerium für Wissenschaft, Kultur und Tourismus (SMWK) and the Sächsische Aufbaubank (SAB).

Acknowledgments

The authors want to thank the editors and reviewers for their helpful comments and constructive suggestions with regard to the revision of the paper.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abicht, J., Wiese, T., Hellmich, A., and Ihlenfeldt, S. (2021). "Interface-free connection of mobile robot cells to machine tools using a camera system," in *Advances in automotive production technology – theory and application*. Editors P. Weißgraeber, F. Heieck, and C. Ackermann (Berlin, Heidelberg: Springer Berlin Heidelberg), 468–477.

Arents, J., and Greitans, M. (2022). Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.* 12, 937. doi:10.3390/app12020937

Brandenbourger, B., and Durand, F. (2018). Design pattern for decomposition or aggregation of automation systems into hierarchy levels. in 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Turin, Italy, 04-07 September 2018, IEEE, 895–901.

Deutsches Institut fur Normung, E. V. (2020). *VDI/VDE/NAMUR 2658 BLATT4: Automatisierungstechnisches engineering modularer anlagen in der Prozessindustrie - modellierung von Moduldiensten*. USA: Beuth.

Deutschmann, B., Ghofrani, J., and Reichelt, D. (2020). Cognitive production systems: A mapping study. in 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, United Kingdom, 20-23 July 2020, IEEE.

Dietz, U. (2012). Programming system for efficient use of industrial robots for deburring in SME environments. in ROBOTIK 2012; 7th Ger-man Conference on Robotics, Munich, Germany, 21-22 May 2012, VDE.

Dorofeev, K., and Wenger, M. (2019). Evaluating skill-based control architecture for flexible automation systems. in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation, Zaragoza, Spain, 10-13 September 2019, IEEE, 1077–1084.

Falkowski, P., Smater, M., Koper, J., Myśliwiec, A., and Mackiewicz, T. (2020). "An approach towards high-precision docking of the mobile robots for industrial purposes," in *Automation 2020: Towards industry of the future*. Editors R. Szewczyk, C. Zieliński, and M. Kaliczyńska (Cham: Springer International Publishing), 239–247.

Frohm, J., Lindström, V., Winroth, M., and Stahre, J. (2006). The industry's view on automation in manufacturing. *IFAC Proc. Vol.* 39, 453–458. doi:10.3182/20060522-3-FR-2904.00073

Heimann, O., and Guhl, J. (2020). Industrial robot programming methods: A scoping review. in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, Vienna, Austria, 04-07 September 2018, IEEE, 696–703.

Lienenluke, L., Grundel, L., Storms, S., Herfs, W., Konigs, M., and Servos, M. (2018). Temporal and flexible automation of machine tools. in 2018 IEEE 22nd International Conference on Intelligent Engineering Systems, Palmas de Gran Canaria, Spain, 21-23 June 2018, IEEE, 335–340.

Malakuti, S., Bock, J., Weser, M., Venet, P., Zimmermann, P., Wiegand, M., et al. (2018). Challenges in skill-based engineering of industrial automation systems. in 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Turin, Italy, 04-07 September 2018, IEEE, 67–74.

Pedersen, M. R., Nalpantidis, L., Andersen, R. S., Schou, C., Bøgh, S., Krüger, V., et al. (2016). Robot skills for manufacturing: From concept to industrial deployment. *Robotics Computer-Integrated Manuf.* 37, 282–291. doi:10.1016/j.rcim.2015.04.002

Pfrommer, J., Stogl, D., Aleksandrov, K., Escaida Navarro, S., Hein, B., and Beyerer, J. (2015). Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. *A. T. - Autom.* 63, 790–800. doi:10.1515/auto-2014-1157

Pfrommer, J., Stogl, D., Aleksandrov, K., Schubert, V., and Hein, B. (2014). Modelling and orchestration of service-based manufacturing systems via skills. in Proceedings of the 2014 IEEE Emerging Technology and Factory Automation, Barcelona, Spain, 16-19 September 2014, IEEE, 1–4.

Profanter, S., Breitkreuz, A., Rickert, M., and Knoll, A. 2019 "A hardware-agnostic OPC UA skill model for robot manipulators and tools, " in 2019 24th IEEE International Conference. Zaragoza, Spain, 10-13 September 2019, IEEE, 1061–1068.

Radanovic, P., Jereb, J., Kovac, I., and Ude, A. (2021). Design of a modular robotic workcell platform enabled by Plug & produce connectors2021 20th International Conference on Advanced Robotics, Ljubljana, Slovenia, 06-10 December 2021, IEEE, 304–309.

Sanneman, L., Fourie, C., and Shah, J. A. (2020). *The state of industrial robotics: Emerging technologies, challenges, and key research directions*. Cambridge: Massachusetts Institute of Technology.

Saukkoriipi, J., Heikkilä, T., Ahola, J. M., Seppälä, T., and Isto, P. (2020). Programming and control for skill-based robots. *Open Eng.* 10, 368–376. doi:10.1515/eng-2020-0037

Siciliano, B., and Khatib, O. (2016). *Springer handbook of robotics*. New York: Cham: Springer International Publishing.

Steinmetz, F., Wollschlager, A., and Weitschat, R. (2018). RAZER—a HRI for visual task-level programming and intuitive skill parameterization. *IEEE Robot. Autom. Lett.* 3, 1362–1369. doi:10.1109/LRA.2018.2798300

Wojtynek, M., Steil, J. J., and Wrede, S. (2019). Plug, plan and produce as enabler for easy workcell setup and collaborative robot programming in smart factories. *Kunstl. Intell.* 33, 151–161. doi:10.1007/s13218-019-00595-0

Zhou, Z., Xiong, R., Wang, Y., and Zhang, J. (2020). Advanced robot programming: A review. *Curr. Robot. Rep.* 1, 251–258. doi:10.1007/s43154-020-00023-4

Zimmermann, P., Axmann, E., Brandenbourger, B., Dorofeev, K., Mankowski, A., and Zanini, P. (2019). Skill-based engineering and control on field-device-level with OPC UA. in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation, Zaragoza, Spain, 10-13 September 2019, IEEE, 1101–1108.

frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 30 September 2022
DOI 10.3389/frobt.2022.1002226

Check for updates

OPEN ACCESS

EDITED BY
Khoshnam Shojaei,
Islamic Azad University of Najafabad, Iran

REVIEWED BY
Omid Elhaki,
Islamic Azad University of Najafabad, Iran
Amir Naderolasli,
Islamic Azad University of Najafabad, Iran

*CORRESPONDENCE
Mohamad Bdiwi,
Mohamad.bdiwi@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems, a section of the journal Frontiers in Robotics and AI

RECEIVED 24 July 2022
ACCEPTED 19 August 2022
PUBLISHED 30 September 2022

CITATION
Bdiwi M, Al Naser I, Halim J, Bauer S, Eichler P and Ihlenfeldt S (2022), Towards safety4.0: A novel approach for flexible human-robot-interaction based on safety-related dynamic finite-state machine with multilayer operation modes. *Front. Robot. AI* 9:1002226. doi: 10.3389/frobt.2022.1002226

Towards safety4.0: A novel approach for flexible human-robot-interaction based on safety-related dynamic finite-state machine with multilayer operation modes

Mohamad Bdiwi[1]*, Ibrahim Al Naser[1], Jayanto Halim[1], Sophie Bauer[1], Paul Eichler[1] and Steffen Ihlenfeldt[2]

[1]Department of Cognitive Human-Machine-Systems, Fraunhofer Institute for Machine-Tools and Forming-Technology, Chemnitz, Germany, [2]Department of Production System and Factory Automation, Fraunhofer Institute for Machine-Tools and Forming-Technology, Chemnitz, Germany

In the era of Industry 4.0 and agile manufacturing, the conventional methodologies for risk assessment, risk reduction, and safety procedures may not fulfill the End-User requirements, especially the SMEs with their product diversity and changeable production lines and processes. This work proposes a novel approach for planning and implementing safe and flexible Human-Robot-Interaction (HRI) workspaces using multilayer HRI operation modes. The collaborative operation modes are grouped in different clusters and categorized at various levels systematically. In addition to that, this work proposes a safety-related finite-state machine for describing the transitions between these modes dynamically and properly. The proposed approach is integrated into a new dynamic risk assessment tool as a promising solution toward a new safety horizon in line with industry 4.0.

KEYWORDS

dynamic risk analysis, human-robot collaboration, human safety, industry 4.0, finite state machine

1 Introduction

The CE conformity declaration (CE-marking according to the Machinery Directive 2006/42/EC) is the final mandatory step in Europe, which indicates that the machinery (e.g. the robot cell) meets European Union standards for health, safety, and environmental protection. Figure 1 shows the safety-related standards for the implementation of robotic systems according to 2006/42/EC. In general, the standards are divided into three categories: 1) Type A standards: describe the general principles of machinery design principles, 2) Type B standards: describe the generic safety standards covering safety aspects and safeguard across a wide range of machinery and 3) Type C standards: describe the safety standards for a specific machine group. ISO 12100:2010 (ISO 12100:2010, 2016)

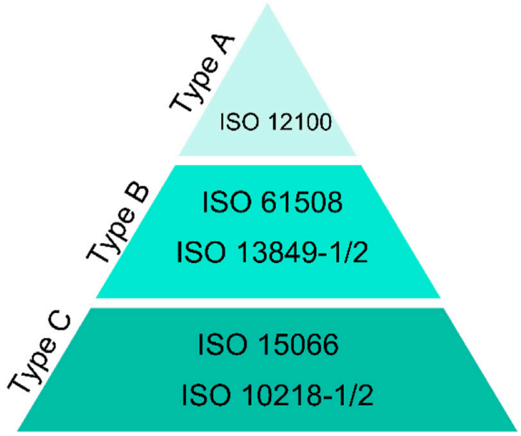


FIGURE 1
ISO types.

as basic safety standards (type A) define the basic terminologies and principles for achieving safety in the design of machinery. Furthermore, it specifies a methodology for risk assessment and risk reduction. The Type B standards are also generic safety standards. However, they cover specific safety aspects or one type of safeguard which can be used within a wide range of machinery. E.g. ISO 13849-1:2015 (ISO 13849-1:2015, 2015) provides the safety requirements and guidelines for designing and integrating safety-related parts of control systems. The ISO 13849-2:2012 (ISO 13849-2:2012, 2012) defines the procedures and conditions for validating the designed safety functions according to the ISO 13849-1. In the standards "Type C", the safety requirements for a particular machine or group of machines are addressed in detail, e.g. ISO 10218-1:2021 (ISO 10218-1:2021, 2021) specifies requirements and guidelines for the inherent safe design, protective measures, and information for the use of industrial robots, while the ISO 10218-2:2021 (ISO 10218-2:2021, 2021) specifies safety requirements for the integration of industrial robots and industrial robot systems. ISO/TS 15066:2016 (ISO/TS 15066:2016, 2016) as technical specification defines safety requirements for collaborative industrial robot systems and the work environment. Revisions to the robotics-related standard are under process by the technical committee "ISO/TC 299".

According to the ISO/TS 15066:2016, "Safety-Rated Monitored Stop" (SRMS), "Speed and Separation Monitoring" (SSM), "Hand Guiding" (HG), and "Power and Force Limiting" (PFL) are the main four collaborative methods for collaborative operation (HRC). Figure 2 illustrates these operation modes and techniques for collaborative operation for industrial robot systems. In "SRMS", the safety sensors directly stop the robot's operation when a human enters the work cell respectively collaborative workspace. "SSM" mode allows the worker to be in the collaborative workspace while the robot moves by maintaining the protective distance between worker and robot. This method for collaborative operation ensures to stop the robot before any collision with the worker may occur. The safeguarded space should be monitored with external safety sensors (ISO 13855:2010, 2010) provides the basis for positioning the safeguards taking into account the speed of parts of the human body. Various approaches for "SSM" considering dynamic speed regulation of robots are presented in (Byner et al., 2019), (Kolbeinsson et al., 2018), and (Michalos et al., 2015). In another work (Rashid et al., 2020), the minimum protective distance between worker and robot has been addressed in details. Furthermore, the system performance in scenario with heavy-duty robot has been systematically investigated.

As is shown in Figure 2, during "HG" collaborative mode: the worker can use a hand-operated device "Control@Flange" or even gesture "GestureControl" to control the motion of the robot. HG requires additional PFL or SRMS safety functions. Most of the cobots possess integrated hand-guiding functionalities. However, the industrial robot systems require an external safe force/torque sensor or hand-guided system to measure the worker's applied forces and torques. In "PFL", the robot system may come into direct contact either intentionally or accidentally with the worker. However, the contact power and force should be limited to a safe level to reduce risk and avoid harming humans. Most of the implemented applications with PFL operation mode are currently realized based on lightweight "cobots", being industrial robots constructed for it and equipped with PFL safety functions. PFL in high payload HRC applications is still difficult to implement due to the high inertial mass of the robotic itself and potentially dangerous collisions. In general, the safety requirements for these methods are still under development for the upcoming revision of ISO 10218 (to be published in 2022).

Some industrial robot systems offer two manual operation modes for teaching or commissioning processes. The teaching process in the first operation mode can be performed without any barriers but with reduced speed (max. 250 mm/s). While in the second operation mode, the worker should stay outside the cell to let the robot moves at full speed (e.g. 2 m/s). During the teaching phase, the cobots usually work with the reduced manual operation mode. The operation modes "4" and "5" are typically used in the fully-automated tasks. In the fifth mode, an external control system, "e.g. Programmable-Logic-Controller (PLC)", is used as a master control unit.

The operation mode is usually the primary key element for further steps during risk analysis, risk reduction, and defining the required safety sensors/functions. The Safety-related sensors/functions significantly impact the design of HRC applications in terms of efficiency and flexibility. Any change in the process, workflow, product, layout, etc., requires a new identification of possible hazards. Furthermore, the whole procedure for getting the CE marking could be repeated from scratch. Such kinds of

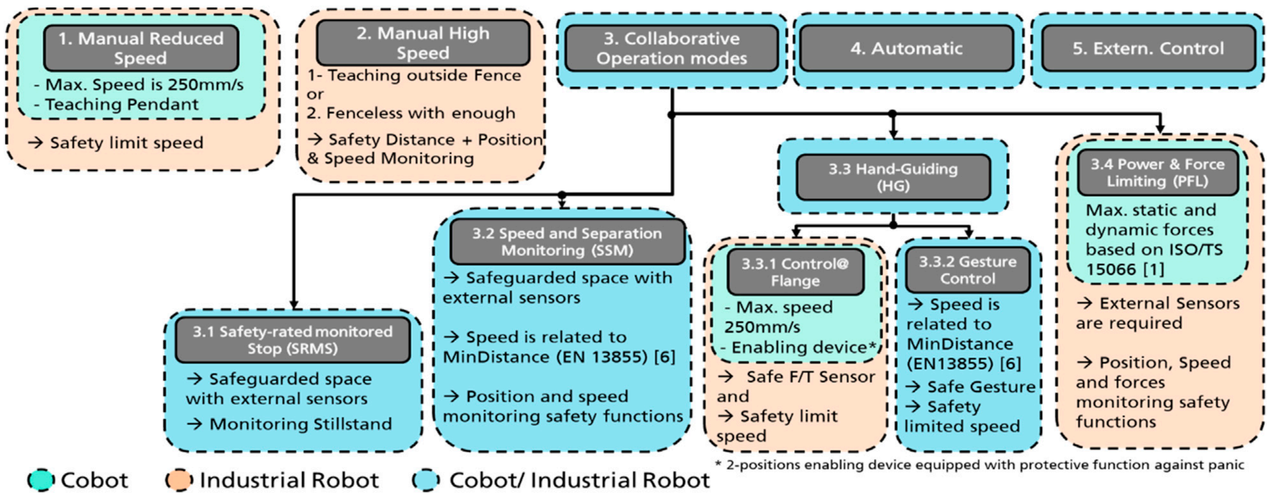

1. Manual Reduced Speed
- Max. Speed is 250mm/s
- Teaching Pendant
→ Safety limit speed
2. Manual High Speed
1- Teaching outside Fence
or
2. Fenceless with enough
→ Safety Distance + Position & Speed Monitoring
3. Collaborative Operation modes
4. Automatic
5. Extern. Control
3.1 Safety-rated monitored Stop (SRMS)
→ Safeguarded space with external sensors
→ Monitoring Stillstand
3.2 Speed and Separation Monitoring (SSM)
→ Safeguarded space with external sensors
→ Speed is related to MinDistance (EN 13855) [6]
→ Position and speed monitoring safety functions
3.3 Hand-Guiding (HG)
3.3.1 Control@ Flange
- Max. speed 250mm/s
- Enabling device*
→ Safe F/T Sensor and
→ Safety limit speed
3.3.2 Gesture Control
→ Speed is related to MinDistance (EN13855) [6]
→ Safe Gesture
→ Safety limited speed
3.4 Power & Force Limiting (PFL)
Max. static and dynamic forces based on ISO/TS 15066 [1]
→ External Sensors are required
→ Position, Speed and forces monitoring safety functions
Cobot
Industrial Robot
Cobot/ Industrial Robot
* 2-positions enabling device equipped with protective function against panic

FIGURE 2
Possible Operation modes related to cobots and industrial robots.

policies are very cumbersome for both systems integrator and operator. They are almost no longer possible in the era of Industry 4.0, where agile manufacturing systems, flexible layout, dynamic processes, and customized product features are the main characteristics. In addition, myriad applications could require multiple sequential operation modes on the same cell to efficiently fulfill the required task, which is currently not feasible due to the restricted safety producers and lack of acceptable safety-related solutions. This work proposes a new approach for multilayer HRI operation modes merged with various process-related human-robot-interaction levels systematically. The proposed system is implemented using a dynamic finite state machine architecture. The following section will present an overview of the state of the art. Section 3 will illustrate the proposed approach in detail, while Section 4 explains one use case as an example for presenting the advantages of the proposed approach in reality. Section 5 will focus on integrating the proposed methodology in a newly developed risk assessment tool as a final result of this work. Finally, the proposed work will be concluded in the last section.

2 State of the art

As mentioned, the collaborative operation modes describe the interaction between humans and robots from a safety point of view. This focus makes executing the collaborative operation modes in the applications very tricky. In other words, there is a large gap between process-related and safety-related functionalities during the design phase of the collaborative tasks. Furthermore, the complexity of safety design forces the safety planner to consider the operation modes individually for fixed tasks. Any changes in the operation modes or even the process could require new certification procedures. In addition, when various interactions between humans and robots are desirable, implementing multiple operation modes is laborious. This research problem has initiated researchers worldwide to exploit the approaches to ensure HRI at the implementation level (from the process point of view). A general insight of the current framework and state of the art in the implementation level for safety in industrial robotic environments has been reviewed by S. Robla-Gomez et al. (Robla-Gomez et al., 2017). Even though this review shows many possibilities to implement safeguarding with sensors, it does not mention any standards or metrics at the implementation level that could bridge the safety operation modes with the functional safety of the machinery. Some quantitative metrics have been introduced in several works (Galin et al., 2020), (Kolbeinsson et al., 2018), (Marvel et al., 2020). Kolbeinsson et al. have suggested a metric by visualizing interaction level in HRI based on human and robot efforts (Kolbeinsson et al., 2018). Marvel et al. and Aaltonen et al., propose other metrics as quantiative measurements in the design of the HRI process (Marvel et al., 2020) and (Aaltonen et al., 2018). Galin and Mescheryakov proposed the quantified process parameters for HRI regarding efficiency (Galin et al., 2020). Although these approaches have covered the quantification and validation process to benchmark the HRI design and process, the metrics are still far from implementation. An approach to validate safety in HRI has been introduced by Valori et al. by introducing safety skills and their validation protocols (Valori et al., 2021). Although the safety skills have been tested under strict validation protocols to reduce specific risks, the safety skills are limited to simple tasks. The protocols are more into validation measurements for testing purposes than implementation purposes. Michalos et al. have introduced and suggested an approach to ensure safety in HRI by considering and combining highlighted functional safety, safety operation modes, and machinery directive based on the shared tasks and workspace at the implementation level (Michalos et al., 2015). Unfortunately, there is no general overview of which complementary functional safety, safety operation mode, and machinery directive are required for different tasks or interactions between humans and robots. This open point can also be found in Askarpour et al.'s method, which uses a complex non-deterministic formal model resulting from human errors (Askarpour et al., 2017). In the latest work of Gualtieri et al. (2022), guidelines to develop and validate safety aspects in the HRI workspace have been suggested. This method covers a quantification methodology for the possible mechanical hazards in the design of the HRI workspace and suggests risk assessment strategies based on the standards. Although the methodology has a good scope to cover the development of safety measures in HRI for non-expert users, the method only focuses on the assembly process.

The approach's overview and classification methodology of this work has been exploited in a previous paper (Bdiwi et al., 2017) by introducing a new classification methodology for HRI applications using four interaction levels. *Via* the proposed four levels of interaction, most of the possible HRI applications in the industry could be classified. Furthermore, the safety procedures and safe zones could be derived based on single or even clustered safety operation modes for each interaction level. Even though work has focused on improving the safety procedures for HRI; a general system architecture was missing for quantifying the complex safety functionalities in the level of interactions. The finite state machine (FSM) is one effective method to implement a technical system with a minor development curve. FSM offers a very effective method in the implementation of complex robot behaviors in comparison to monolithic programming. The system's sequential, deterministic, and causal behaviours ease the implemented robotic system for debugging, modification, and enhancement. Balogh and Obdrzalek have introduced these benefits (Balogh et al., 2018). Although FSM would offer many simplifications in developing and implementing the robotics system, design errors, cognitively complex human decision-making errors, and other failures are challenging to observe. Another possible approach to identify these new causal factors is using a top-down analysis called system theoretic process analysis (STPA), a method used in a technical system to show its behavior and address component interactionfailures. Integrating between STPA and FSM could assist the safety analysis in identifying the dysfunctional behavior of a system (Abdulkhaleq et al., 2013). However, the proposed approach aims to couple the process-related classification (Interaction-levels) with the safety-related classification (operation modes) to reduce the effort and complexity during risk analysis and mitigation. A general comparison between STPA and FSM is shown in Figure 3. Figure 3 left shows the intermodule communication in a SRMS collaborative operation. In this figure, all of the data flows can be seen in the STPA diagram. When failures occur, the failure analysis can be performed by checking the causality in the control structure. Figure 3 right depicts the FSM diagram for the same use-case. In contrast to STPA, FSM method is focused on the causality of the system state. Each state (action) will be triggered by a deterministic signal. Hence, it can be concluded that STPA has a general focus on analyzing failure in the hardware and communication level. In contrast to STPA, FSM focuses the debugging level in the functionality of the system. Hence, this work focuses on the FSM more than STPA for hardware integration. This approach allows the user also to plan and implement complex and agile collaborative tasks flexibly and intuitively.

3 The proposed approach

Figure 4 illustrates the main structure of the proposed approach. The first layer "Level-Planner", facilitates the classification of the proposed application according to the interaction level (Bdiwi et al., 2017). The second layer presents a combination of possible clustered operation modes to fulfill the described task in the level-planner. Every cluster consists of various operation modes shown in the third layer and modeled using a finite state machine. The fourth layer contains all the required safety functions for every operation mode. All layers are systematically coupled to ensure the system's legibility and the user's comprehend-ability. They will be described in detail in the following sections.

3.1 Level-planer

As (Bdiwi et al., 2017) explained, the main difference between interaction level 1 and interaction level 2 is that the human presence is not intended during the normal process of level 1. In other words, there is no shared task in level 1, and the human enters the robot cell only, e.g. in emergencies. In level 2, the robot and humans work on a shared task. However, the cooperation is still low. At this level, the robot can move toward a predefined position during the human presence, taking into account the safety distance and type of collaborative operation modes. Level 3 requires active robot control during the shared task, where the robot can change its position and react to human movements. The possible shared tasks in this level are, e.g. Handing-over, Gesture-based control, or even automatic path planning. In path planning, the robot can automatically change the working height based on human anthropometry or regenerate a new path during the shared task to avoid any collision with the human. In Level 4, physical HRI is necessary to fulfill the task. For instance, the

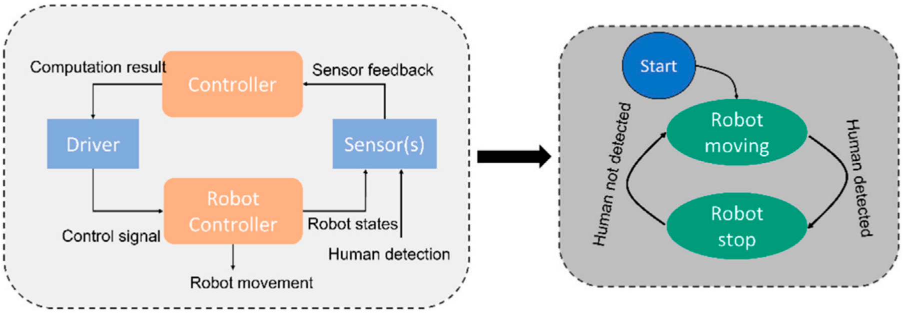


FIGURE 3
Comparison between an STPA diagram (left) and FSM diagram (right) for a fictional SRMS use case.

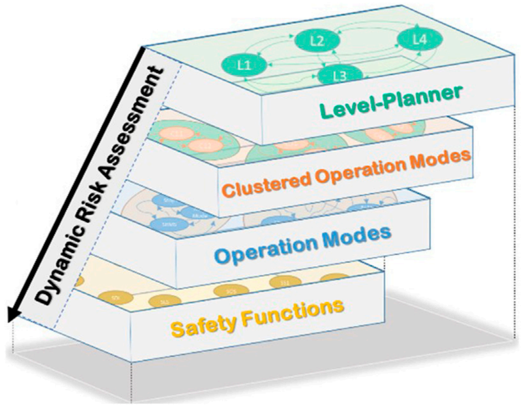
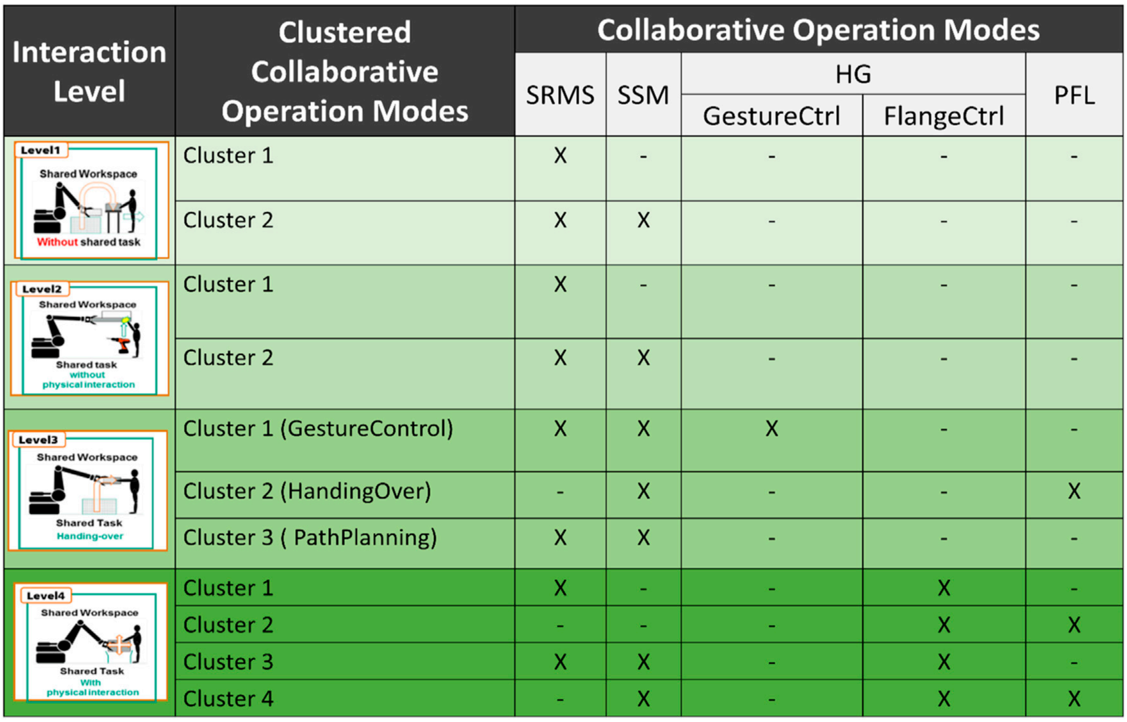


FIGURE 4
Structure of the proposed approach.

robot can bring heavy components to the human or a predefined position near the assembly line; then, the human can guide the robot to a final position by using human forces (Handguiding-System). In this work, the proposed interaction levels are focused on the implementation of HRC with serial manipulators. It is also possible to implement the proposed method for mobile manipulation by extending the safety functions with safety related standards and control system e.g. (Elhaki and Shojaei, 2020) and (Elhaki and Shojaei, 2022) for mobile robotic.

Figure 5 illustrates the possible collaborative operation modes at each interaction level. **Level 1** can contain, e.g. two clusters. **Cluster 1**: SRMS-Standalone: the whole workspace and work process will be performed in SRMS mode. **Cluster 2**: SRMS + SSM: this cluster allows the robot to change its speed according to the human position in some process or even defined zones of the workspaces. As is already mentioned, the human enters the robot cell only, e.g. in emergencies. Hence, when the human enters any dangerous area monitored by the safety sensors, the robot will have to stop in stop-category 1. Therefore, user confirmation is required for activating the robot again. Cluster 2 allows the user to plan different collaborative operation modes in different zones (e.g. SSM for the entrance areas besides the footpaths to avoid any unnecessary robot stops, while the SRMS can cover more critical areas). Each cluster has its safety functions which will be described more in the next section. **Level 2** could also consist of the same clusters. However, the safety functions are different, as is shown in Supplementary Appendix S2. At this level, the human is intended to participate in the process during the shared task. The robot should be able to restart automatically without an additional confirmation by the worker. Hence, the robot will have to stop in stop-category 2 when the human enters the danger areas. Thestop-category 2 will increase the efficiency and availability of the facility. Figure 5 also presents three possible clusters of collaborative operation modes in **level 3**. **Cluster 1** “GestureCtrl” gives the users the possibility to control the robot based on their gestures. The user can perform these procedures within the Handguiding “HG” (GestureControl in Figure 2 and as a shortcut GestureCtrl in Figure 4) operation mode. Usually, gesture control is performed during a specific task during the whole process or even in a defined and restricted area. The cluster that is mixing SSM, and SRMS with HG-GestureCtrl could be very useful for increasing the facility’s efficiency and flexibility on the one hand. On the other hand, it ensures the safety of the human during all processes with the required safety functions, no more, no less. **The second cluster** “Handing-Over” could be used in any process which contains handing-over tasks between humans and robots. This cluster can combine, e.g. SSM with PFL, to ensure that the maximum possible impact forces/torques during the collision will not increase the maximum described values in ISO/TS 15066. **The third cluster**, “PathPlanning”, requires additional safety functions for monitoring the paths during the shared task. In **level 4**, four possible clusters are presented. In the cobots applications, the HG-FlangCntrl can be combined with PFL **(e.g. cluster2)** or with modes, SSM, and PFL



Interaction Level	Clustered Collaborative Operation Modes	Collaborative Operation Modes				
		SRMS	SSM	HG		PFL
				GestureCtrl	FlangeCtrl	
Level1 Shared Workspace Without shared task	Cluster 1	X	-	-	-	-
	Cluster 2	X	X	-	-	-
Level2 Shared Workspace Shared task without physical interaction	Cluster 1	X	-	-	-	-
	Cluster 2	X	X	-	-	-
Level3 Shared Workspace Shared Task Handing-over	Cluster 1 (GestureControl)	X	X	X	-	-
	Cluster 2 (HandingOver)	-	X	-	-	X
	Cluster 3 (PathPlanning)	X	X	-	-	-
Level4 Shared Workspace Shared Task With physical interaction	Cluster 1	X	-	-	X	-
	Cluster 2	-	-	-	X	X
	Cluster 3	X	X	-	X	-
	Cluster 4	-	X	-	X	X

FIGURE 5
Some possible examples for clustered collaborative operation modes in every.

(e.g. cluster4). The HG-FlangCntrl can be combined with SRMS **(e.g. cluster 1)** or with SRMS and SSR **(e.g. cluster3)** in the heavy-duty application. PFL in high payload HRC applications is still difficult to implement due to the high inertial mass of the robotic system and potentially dangerous collisions. The clusters mentioned previously are built based on interaction levels (Bdiwi et al., 2017). These clusters were chosen as examples of common scenarios. However, unusual scenarios or special requirements may be realized based on the level of interaction with the new cluster. The new cluster defines the state graph and which safety functions could be involved.

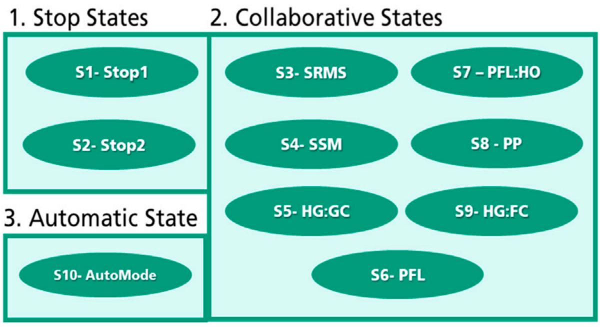


FIGURE 6
Machine states.

3.2 Multilayer collaborative operation modes

In the proposed approach, every collaborative operation mode will represent one machine state to build a safety-related finite state machine properly. These collaborative states are; S3-SRMS, S4-SSM, S5-HG: GC, S6-PFL, S7-PFL: HO, S8-PP, and S9-HG: FC. S3 (SRMS) and S4 (SSM) are already well explained in the first section. S5 (HG: GC) represents the state of Handguiding using gesture control in level 3, while the S9 (HG: FC) represents the hand guiding use, e.g. hand guiding device mounted on the robot flanch. S7 (PFL: HO) is a special collaborative state for handing-over tasks based on PFL operation modes in the third interaction level. S8 (PP) is also a particular collaborative state during the third interaction level when the robot can modify its paths. All the possible proposed machine states are shown in Figure 6. In addition to that, two states represent the stop categories “S1 (Stop1) and S2 (Stop 2)”, and one state represents the automatic mode “S10 (AutoMode)”. All the previously mentioned collaborative operation modes can

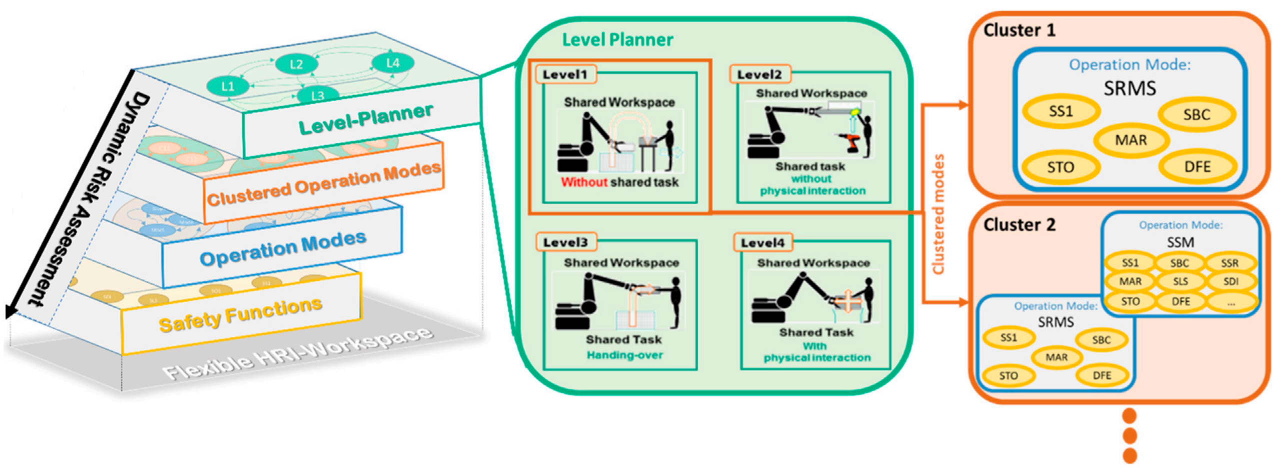


FIGURE 7
Example of the possible collaborative operation mode clusters of level 1 with their safety functions.

be integrated to create variable clusters concerning the interaction levels.

Figure 7 illustrates an example of the possible clusters in level 1. Furthermore, it presents the required safety functions of each cluster. All the safety functions and their acronyms are described in Supplementary Appendix S1. The transition and state conditions between the operation modes can be coupled with the safety functions using the finite state machine structure. More details will come in the next two sections.

3.3 Enhanced safety functions

As is already in section 3.1 presented, each level has various clusters of operation modes consisting of a bundle of safety functions. The basic safety functions are listed in DIN EN IEC 61800-5-2:2017 (DIN EN IEC 61800-5-2:2017-11, 2017). Figure 8 shows the main categories of the required safety functions in robotics applications. The functionalities of all safety functions in these categories are described in Supplementary Appendix S1. The first category is Safe Standstill. It contains all the safety functions for controlling the monitoring the robot during the standstill. These functions are; 1. Safety Stop1 (SS1), 2. Safe Brake Control (SBC), 3. Safe Torque off (STO), 4. Safe Stop2 (SS2) and 5. Safe Operation Stop (SOS). The safety functions in the second category "Safe Motion" are responsible for controlling and monitoring the robot's motion, e.g. 1. Safety Limited Speed (SLS) 2. Safe Speed Monitoring (SSM), 3. Safety Range Speed (SRS), 4. Safe Direction (SDI). The third category of safety function takes care of the robot positions, e.g. 1. Safety Limited Position (SLP) and 2. Safe Cam (SCA). The safety function in the fourth category is the Safe Limited Torque (SLT) for preventing the actuator from exceeding the torque limit. This work proposed also enhanced safety functions within the fifth category, "Safe Guarding", for controlling and monitoring the interaction between the human and the robot. e.g. 1. Dangerfield-Entry (DFE), 2. MinDistance (MiD), 3. Coopfield_Entry (CFE1), 4. ManualRes (MAR) 5. OperationRes (OPR), 6. GestureCntrlStart (GCS), 7. GestureCntrlEnd (GCE) etc. The whole list of the enhanced safety functions is listed in Supplementary Appendix S1. The structure of the proposed approach is dynamic. Hence, it allows the user to extend the bundle of the safety functions if it is required.

3.4 Safety-related finite-state machine for collaborative applications

After presenting the main safety functions and the possible states of the collaborative operation modes, this section will show the main concept of the safety-related finite-state machine for collaborative applications. A transition T_m^n presents the transition from the start state S_n to the end state S_m. Every transition T_m^n consists of a couple of conditions representing the relation between safety functions and the machine state to switch from the start state to the end state. The transition, which contains a set of safety functions with the AND ($\wedge$) operator, is true if all its safety functions are true. While the transition with Or ($\vee$) operator is true if only one of its safety functions is true. For better illustration, the finite state machine of both collaborative clusters in the first level of interaction will be explained in this section. As is shown in Figure 9, there are two states, S1 (Stop1) and S3 (SRMS), in cluster 1 of level 1. There are two transitions, T_3^1 and T_1^3, between S1 and S3. By supposing

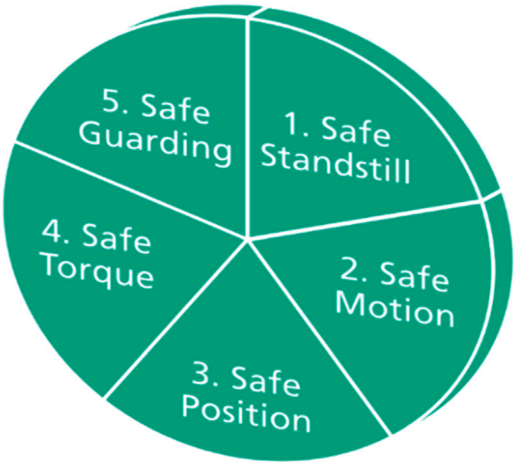


FIGURE 8
Overview of main categories of the safety functions.

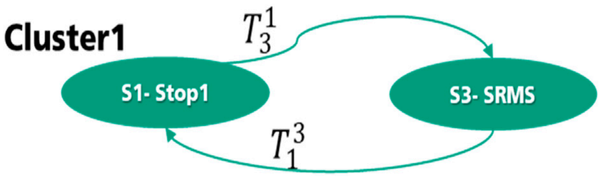


FIGURE 9
States graph of cluster 1—level 1.

that S3 is the start state, the robot goes in S1, when the human enters the danger area for an emergency situation. This happens when the safety function "Dangerfield_Entry *DFE*" is active. Hence, the safety functions SS1 for maintaining the position of the actuator, STO for disabling the torque in the actuator, and SBC for supplying a safe output signal to drive an external brake system should also be active to transfer the machine from the S3-state to the S1-state. These transition conditions are presented in Eq. 1.

$$T_1^3 \rightarrow (DFE \wedge SS1 \wedge SBC \wedge STO) \quad (1)$$

The transition condition from S1-state to S3-State consists of two functions: 1. $\overline{DFE}$ this means that the Dangerfield_Entry DFE is deactivated, and the human has left the danger area. Besides that, a user confirmation through MAR "Manual restart button" is necessary for the Stop1 to ensure that the process could be continued after the inexistence of the human in the danger area, as shown in Eq. 2.

$$T_3^1 \rightarrow (\overline{DFE} \wedge MAR) \quad (2)$$

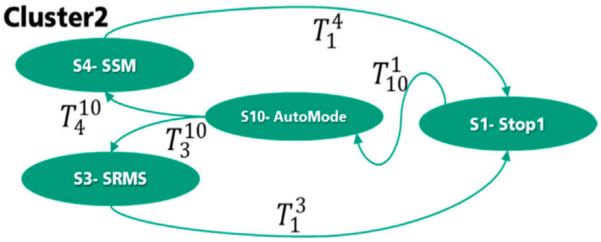


FIGURE 10
State graph of Cluster 2—Level 1.

As is shown in Figure 10, the second cluster of Level 1 has a more complex state graph which consists of 4 states. In this cluster, two other states have been added. S10 (AutoMode) represents the automation mode which can be the start mode. T_3^{10} and T_4^{10} represent two transitions from the automatic mode to both collaborative mode SSM and SRMS. SSM can start when the human enters the collaboration areas by activating the safety functions (Collaborative_Field_Entry CFE). Various speed limitations can be defined in different collaborative fields (e.g. from CFE1 until CFEX). The safety functions, which are listed in Supplementary Appendix S1 in the safe motion, should also be active, as is shown in Eq. 3.

$$T_4^{10} \rightarrow ((CFE1 \vee CFE2 \ldots CFEX) \wedge SLS \wedge SSM \wedge SSR \wedge SDI) \quad (3)$$

The transitions T_1^4 and T_1^3 from S4 and S3 states to the S1-state happen when the DFE is activated, and all the required safety functions of Stop1 are also activated, as is shown in Eqs 4, 5. By transition T_1^4, if one of the safe motion functions is not active (e.g. the actuator speed exceeds the maximum speed limit), the robot goes to S1-state, as is shown in Eq. 5.

$$T_1^3 \rightarrow (DFE) \wedge SS1 \wedge SBC \wedge STO \quad (4)$$

$$T_1^4 \rightarrow (DFE) \vee \left(\overline{SLS} \vee \overline{SSM} \vee \overline{SSR} \vee \overline{SDI}\right) \wedge SS1 \wedge SBC \wedge STO \quad (5)$$

The transition T_{10}^1 from the S1-Stop1 to the S10 automatic mode happens when nobody is inside the danger area. The user confirms it through the safety function operation restart (OPR), as shown in Eq. 6.

$$T_{10}^1 \rightarrow (\overline{DFE} \wedge OPR) \quad (6)$$

When the transition to the automatic mode happens successfully, the user can confirm further through MAR "Manual restart button" to start the SRMS operation mode, as shown in Eq. 7.

$$T_3^{10} \rightarrow (\overline{DFE} \wedge MAR) \quad (7)$$

The manual confirmations at this level are necessary because the human can enter only in emergencies, and it is not a part of the process. In the second level of interaction, these manual confirmations are not required. However, the same principle of the safety-related finite-state machine can be generalized to all other clusters at all levels of interaction. The state graphs and their related transitions of all other clusters are presented in Supplementary Appendix S3. The state graphs in this novel approach represent the clustered collaborative operation modes from the proposed method. The states are selected by considering the defined collaborative operations (ISO 15066) and related safety functions (IEC/ISO 61508). When additional or customized safety functions are integrated in the cluster, the states of the state machines must be extended to represent the additional extension.

4 Case-study

This section will present the application of the collaborative mode clusters through a practical example. Furthermore, it illustrates how the proposed approach can increase the efficiency of the process and ensure the safety of humans in every process. In this use case, handling and machining of a car engine (**collaborative machine tending**) are performed using a CNC machine EMAG VMC 300 MT integrated with a heavy-duty robot Kuka KR-180 Prime 2,900. The robot transfers the engine to different places as it is a heavy task for a human. Figure 11 illustrates the cell layout with the locations of the robot and machine. Additionally, a convey and two tables exist for the quality checking process. A storage place is needed on the left side of the cell where the finished items are placed there.

The scenario is as follows:

1- The storage place has unfinished pieces that the robot is programmed to take from the storage place in a known sequence.
2- A robot is responsible for moving the item from the storage place to the CNC machine. The robot places the item slowly, and the machine can fix the item to process.
3- The machine processes the item for creating screw holes and, it takes 10 min for each item.
4- After the process, the machine opens the safety door while robots move toward the machine. The robot grabs the item from the machine.
5- The quality process is a cooperation between humans and robots. The robot brings the item to the middle table, and the human starts checking the quality of the CNC process. Based on a command from the human, the robot moves instead to the storage place or towards the convey for rechecking and maintenance station.
6- The robot moves from the conveyor point to bring a new item from the storage place, or the robot will be there already if the quality checking is fine. Finally, the process repeats itself till the items are finished.

Such a scenario incorporates the human and the robot together to perform the process effectively. Furthermore, the human risk in such a cell will be high compared to a normal cell where human skill is not required. Hence, a proper risk assessment procedure could reduce the risk to humans in this collaborative environment.

In this use case, the interaction between humans and robots can be clustered under Level 3 or Level 4. It depends on the type of hand-guiding concept during the quality process. Here, the human can control and rotate the robot through camera-based neutral gestures (level 3) or through a hand-guiding device on the robot flange (level 4) to check the engine's quality from different perspectives. As shown in Figure 12, the robot can work under SRMS operation mode while picking the item from storage, transporting it to the CNC, and waiting for the machining process. When the quality process starts, the robot can switch to the HandGuiding operation mode. During the final process, the robot can work under SSM while transporting the item to the maintenance station or under SRMS when the robot should transport the item to the storage back if the item's quality is fine. Choosing the robot type and operation mode in such applications depends on many factors (Schneider et al., 2020), e.g. batch size, processing time by the machine, transporting time by the robot, required time for the manual tasks performed by humans. Using the proposed approach, can the user design the safety procedures of such agile and dynamic environments flexibly and adequately.

5 Dynamic risk assessment

In order to reduce the risks in HRI robot cells, three management strategies include 1) Inherently safe design, 2) Guards and protective devices, and 3) administrative information, as are written in the ISO 12100. The most related paper (Realyvásquez-Vargas et al., 2019) in risks assessment for HRI addressed the first both strategies of risk management. These strategies are only implementable on their predefined use-case. Otherwise, other works give only a theoretical overview and metrics for risk assessment (Franklin et al., 2020) and (Hanna et al., 2020). Compared to these works, the proposed approach is generic and it can be implemented in any safety control device. Furthermore, it simplifies for the end-user the derivation of technical risk management in HRI-context by classifying HRI based on shared workspaces and tasks. Each cluster represents a use-case scenario regarding the requirement of the user. Hence, the functional safety inside the cluster can be customised depending on the user's requirements. The recent study of Hornung und Wurll (Hornung and Wurll, 2021) addresses that lack of know-how and skills are the biggest hindrances in implementing collaborative robot systems. Figure 13 illustrates the integration of the proposed approach within the risk assessment methodology derived from ISO 12100 and ISO 13849-1. It extends the typical risk assessment as presented in grey color. The proposed approach

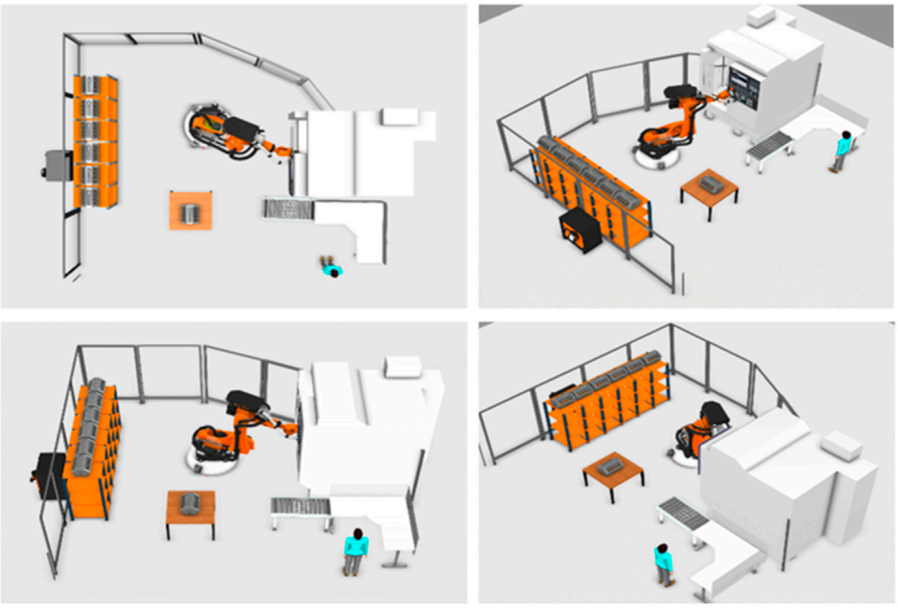


FIGURE 11
Fenceless manufacturing (Human, Machine and Robot).

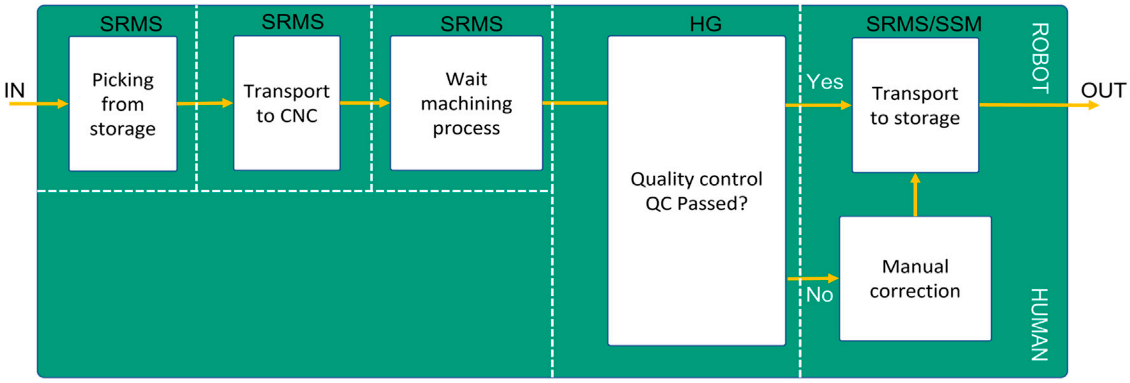


FIGURE 12
Use-Case Process-Flow concerning the operation mode.

begins with the determination of the machine limits, taking into account all the phases of the machinery life to fit the requirements of agile and flexible human-robot applications. The first step describes the machine's characteristics, performances, and limits in an integrated process. Using the proposed level-planner, the user can classify all the planned processes according to the level of interaction with that machine. This procedure allows the user to easily estimate all possible hazards and all access points during that level of interaction. With the help of the clustered operation modes (zone-based, time-based), the user can define the machine movement range, space for people interacting with machines (operators, maintainers), the required time for each process, and the relation between this information and the operation modes properly. To assess the initial risk for each access point in the form of a risk score and to calculate the required performance level of each safety function, one needs to know the number of persons involved affected by the hazard, the duration, and frequency of the hazard exposure, the probability of

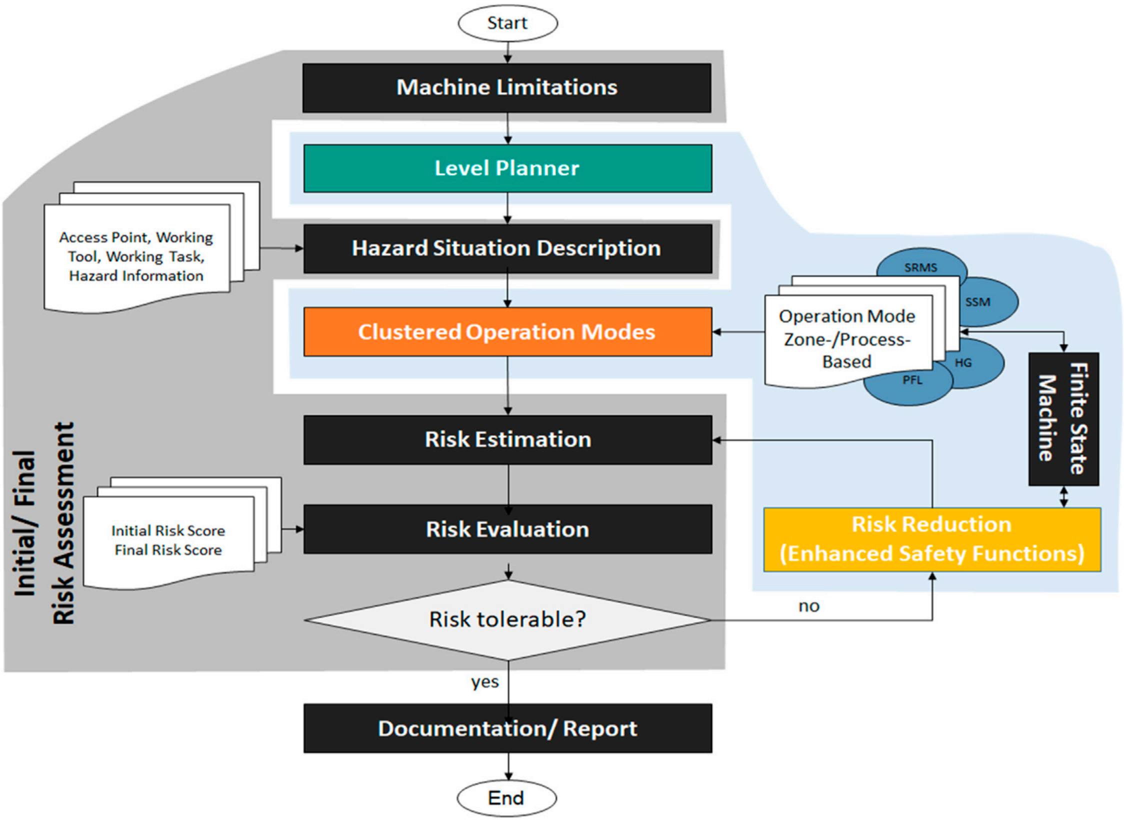

Start
Machine Limitations
Level Planner
Hazard Situation Description
Access Point, Working Tool, Working Task, Hazard Information
Clustered Operation Modes
Operation Mode Zone-/Process-Based
SRMS
SSM
HG
PFL
Finite State Machine
Initial/ Final Risk Assessment
Risk Estimation
Risk Evaluation
Initial Risk Score Final Risk Score
Risk Reduction (Enhanced Safety Functions)
Risk tolerable?
no
yes
Documentation/ Report
End

FIGURE 13
Integration of finite state machine with dynamic risk assessment methodology (Steps of the proposed risk assessment based on ISO 13849-1 and ISO 12100).

occurrence, and the injury severity of the possible human injury.

After evaluating all initial risks, the risk reduction procedures start with its typical three steps, as mentioned previously. The innovative part of this approach focuses on the second step of risk reduction. In other words, it integrates the developed safety-related finite state machine with the safety functions of the guards and protective devices and with the collaborative operations. In general, the limitation of safety sensors should be considered during the risk management. Finally, if the risk is tolerable, the user can automatically generate all the required technical documentation for the declaration of conformity. The user can anytime reconfigure the safety design and risk assessment according to changes in the process, layout, or product.

6 Conclusion

As is already presented, the typical risk assessment procedures are very complex, and take a lot of time and effort. Hence, an expert must follow the implementation of the risk-assessment rules as written in the safety standards and guidelines. The expert should have a broad knowledge of the available safety functions and safe workspace methods. The complex, rigid and static procedures could lead to mistakes even by experts, and the state-of-the-art methodologies prevent them from planning and realizing agile production systems. The proposed approach establishes a dynamic and safe relation between the interaction levels, operation modes, and risk reduction procedures (safety functions). A safety-related finite-state machine has been illustrated for the transitions between these modes dynamically and adequately. Collaborative machine-tending has been described as a use case. Finally, the proposed approach has been integrated into a new dynamic risk assessment methodology as a promising solution toward a new safety horizon in line with industry 4.0.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

MB: Corresponding author, Conceptualization, Methodology, IA: Software and Validation. JH: Visualization and Creation of models, SB: Formal analysis and Data curation: PE: Investigation and Validation. SI: Reviewing and Supervision.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2022.1002226/full#supplementary-material

References

Aaltonen, I., Salmi, T., and Marstio, I. (2018). Refining levels of collaboration to support the design and evaluation of human-robot interaction in the manufacturing industry. *Procedia CIRP* 72, 93–98. doi:10.1016/j.procir.2018.03.214

Abdulkhaleq, A., and Wagner, S. (2013). "Integrating state machine analysis with system-theoretic process analysis," in *Software engineering 2013*. Editors S. Wagner, H. Lichter, and Hrsg (Bonn: Gesellschaft für Informatik e.V.), 501–514.

Askarpour, M., Mandrioli, D., Rossi, M., and Vicentini, F. (2017). "Modeling operator behavior in the safety analysis of collaborative robotic applications," in *Computer safety, reliability, and security. SAFECOMP 2017. Lecture notes in computer science*. Editors S. Tonetta, E. Schoitsch, and F. Bitsch (Cham: Springer), 10488. doi:10.1007/978-3-319-66266-4_6

Balogh, R., and Obdrzalek, D. (2018). "Using finite state machines in introductory robotics: Methods and applications for teaching and learning," in *Robotics in education*. Editors W. Lepuschitz, M. Merdan, G. Koppensteiner, R. Balogh, and D. Obdržálek (Cham: Springer), 829, 85–91. doi:10.1007/978-3-319-97085-1_9

Bdiwi, M., Pfeifer, M., and Sterzing, A. (2017). A new strategy for ensuring human safety during various levels of interaction with industrial robots. *CIRP Ann.* 66, 453–456. doi:10.1016/j.cirp.2017.04.009

Byner, C., Matthias, B., and Ding, H. (2019). Dynamic speed and separation monitoring for collaborative robot applications – concepts and performance. *Robot. Comput. Integr. Manuf.* 58, 239–252. doi:10.1016/j.rcim.2018.11.002

DIN EN IEC 61800-5-2:2017-11 (2017). *Adjustable speed electrical power drive sys–ems - Part 5-2: Safety requirem–nts - Functional*. Berlin: German Institute for Standardization. (IEC 61800-5-2:2016).

Elhaki, O., and Shojaei, K. (2020). Observer-based neural adaptive control of a platoon of autonomous tractor–trailer vehicles with uncertain dynamics. *IET Control Theory & Appl.* 14 (114), 1898–1911. doi:10.1049/iet-cta.2019.1403

Elhaki, O., and Shojaei, K. (2022). Output-feedback robust saturated actor–critic multi-layer neural network controller for multi-body electrically driven tractors with n-trailer guaranteeing prescribed output constraints. *Robotics Aut. Syst.* 154, 104106. doi:10.1016/j.robot.2022.104106

Franklin, C. S., Dominguez, E. G., Fryman, J. D., and Lewandowski, M. L. (2020). Collaborative robotics: New era of human–robot cooperation in the workplace. *J. Saf. Res.* 74, 153–160. doi:10.1016/j.jsr.2020.06.013

Galin, R. R., and Meshcheryakov, R. V. (2020). "Human-robot interaction efficiency and human-robot collaboration," in *Robotics: Industry 4.0 issues & new intelligent control paradigms*. Editor A. Kravets (Cham: Springer), 272. Studies in Systems, Decision and Control. doi:10.1007/978-3-030-37841-7_5

Gualtieri, L., Rauch, E., and Vidoni, R. (2022). Development and validation of guidelines for safety in human-robot collaborative assembly systems. *Comput. Industrial Eng.* 163, 107801. doi:10.1016/j.cie.2021.107801

Hanna, A., Bengtsson, K., Götvall, P. L., and Ekström, M. (2020). "Towards safe human robot collaboration-Risk assessment of intelligent automation," in *25th IEEE international conference on emerging technologies and factory automation (ETFA)* (Vienna, Austria, 1, 424–431.

Hornung, L., and Wurll, C. (2021). *Human-robot collaboration: A survey on the state of the art focusing on risk assessment", robotix-academy conference for industrial robotics (RACIR)*. Germany: Hoppstädten-Weiersbach, 10–17.

ISO 10218-1:2021 (2021). *Robots and robotic devices—safety requirements for industrial robots—Part 1: Robots.* Geneva, Switzerland: International Organization for Standardization.

ISO 10218-2:2021 (2021). *Robots and robotic devices—safety requirements for industrial robots—Part 2: Robot systems and integration*. Geneva, Switzerland: International Organization for Standardization.

ISO 12100:2010 (2016). *Safety of machinery — general principles for design — risk assessment and risk reduction*. Geneva, Switzerland: International Organization for Standardization.

ISO 13849-1:2015 (2015). *Safety of machinery — safety-related parts of control systems — Part 1: General principles for design*. Geneva, Switzerland: International Organization for Standardization.

ISO 13849-2:2012 (2012). *Safety of machinery — safety-related parts of control systems — Part 1: General principles for design*. Geneva, Switzerland: International Organization for Standardization.

ISO 13855:2010 (2010). *Safety of machinery—positioning of safeguards with respect to the approach speeds of parts of the human body*. Geneva, Switzerland: International Organization for Standardization.

ISO/TS 15066:2016 (2016). *Robots and robotic devices—collaborative robots*. Geneva, Switzerland: International Organization for Standardization.

Kolbeinsson, A., Lagerstedt, E., and Lindblom, J. (2018). Classification of collaboration levels for human-robot cooperation in manufacturing. *Adv. Transdiscipl. Eng.* 7, 448–471. doi:10.1080/21693277.2019.1645628

Lacevic, B., Zanchettin, A. M., and Rocco, P. (2020). "Towards the exact solution for speed and separation monitoring for improved human-robot collaboration," in *The 29th IEEE international conference on robot & human interactive communication (RO-MAN)* (IEEE). doi:10.1109/RO-MAN47096.2020.9223342

Marvel, J. A., Bagchi, S., Zimmerman, M., and Antonishek, B. (2020). Towards effective interface designs for collaborative HRI in manufacturing: Metrics and measures, U.S. National institute of standards and technology IEEE transactions on human-machine systems. *ACM Trans. Hum. Robot. Interact.* 9 (4), 1–55. Article 25. doi:10.1145/3385009

Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovrakis, D., and Chryssolouris, G. (2015). Design considerations for safe human–robot collaborative work places. *Procedia CIRP* 37, 248–253. doi:10.1016/j.procir.2015.08.014

Rashid, A., Peesapati, K., Bdiwi, M., Krusche, S., Hardt, W., and Putz, M. (2020). "Local and global sensors for collision avoidance," in *2020 IEEE international conference on multisensor fusion and integration for intelligent system* (IEEE), 354–359. doi:10.1109/MFI49285.2020.9235223

Realyvásquez-Vargas, A., Arredondo-Soto, K. C., García-Alcaraz, J. L., Márquez-Lobato, B. Y., and Cruz-García, J. (2019). Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company. *Robotics Computer-Integrated Manuf.* 57, 315–328. doi:10.1016/j.rcim.2018.12.015

Robla-Gomez, S., Becerra, V. M., Llata, J. R., Gonzalez-Sarabia, E., Torre-Ferrero, C., and Perez-Oria, J. (2017). Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access* 5, 26754–26773. doi:10.1109/ACCESS.2017.2773127

Schneider, C., Hernandez, F. J., Suchanek, T., Bdiwi, M., Mironovova, M. H., and Putz, M. (2020). "Hybrid workstations: A simultaneous planning method for economic-oriented selection between industrial and collaborative robots," in *Conference: Robotix-Academy conference for industrial robotics (RACIR 2020)* (Saarbrücken, Germany.

Valori, M., Scibilia, A., Fassi, I., Saenz, J., Behrens, R., Herbster, S., et al. (2021). Validating safety in human–robot collaboration: Standards and new perspectives. *Robotics* 202110 (2), 65. Article 65. doi:10.3390/robotics10020065

TYPE Original Research
PUBLISHED 04 October 2022
DOI 10.3389/frobt.2022.1001955

No-code robotic programming for agile production: A new markerless-approach for multimodal natural interaction in a human-robot collaboration context

Jayanto Halim[1]*, Paul Eichler[1], Sebastian Krusche[1], Mohamad Bdiwi[1] and Steffen Ihlenfeldt[2]

[1]Departement of Cognitive Human-Machine System, Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany, [2]Departement of Production System and Factory Automation, Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany

OPEN ACCESS

EDITED BY
Khoshnam Shojaei,
Islamic Azad University of Najafabad, Iran

REVIEWED BY
Abbas Chatraei,
Islamic Azad University of Najafabad, Iran
Amir Naderolasli,
Islamic Azad University of Najafabad, Iran

*CORRESPONDENCE
Jayanto Halim,
jayanto.halim@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 24 July 2022
ACCEPTED 17 August 2022
PUBLISHED 04 October 2022

CITATION
Halim J, Eichler P, Krusche S, Bdiwi M and Ihlenfeldt S (2022), No-code robotic programming for agile production: A new markerless-approach for multimodal natural interaction in a human-robot collaboration context.
Front. Robot. AI 9:1001955.
doi: 10.3389/frobt.2022.1001955

Industrial robots and cobots are widely deployed in most industrial sectors. However, robotic programming still needs a lot of time and effort in small batch sizes, and it demands specific expertise and special training, especially when various robotic platforms are required. Actual low-code or no-code robotic programming solutions are exorbitant and meager. This work proposes a novel approach for no-code robotic programming for end-users with adequate or no expertise in industrial robotic. The proposed method ensures intuitive and fast robotic programming by utilizing a finite state machine with three layers of natural interactions based on hand gesture, finger gesture, and voice recognition. The implemented system combines intelligent computer vision and voice control capabilities. Using a vision system, the human could transfer spatial information of a 3D point, lines, and trajectories using hand and finger gestures. The voice recognition system will assist the user in parametrizing robot parameters and interacting with the robot's state machine. Furthermore, the proposed method will be validated and compared with state-of-the-art "Hand-Guiding" cobot devices within real-world experiments. The results obtained are auspicious, and indicate the capability of this novel approach for real-world deployment in an industrial context.

KEYWORDS

intuitive robot programming, multimodal intraction, learning from demonstration, human robot collaboration, no-code robotic teaching

1 Introduction

Human-Robot Collaboration (HRC) has been a prevalent concept in the industry. Compared to the fully automated solution in serial production, HRC offers flexibility to meet the market's demand for high product variability, diversity, and even batch size 1 as dictated in the current trend of agile production concept (Chryssolouris et al. (2012)). However, reconfiguring and reprogramming the production plan with industrial robots are technical bottlenecks for end-users without or with adequate expertise in robotic programming. Variety and specific domains in robotic programming languages are currently serious impediments to robotic system (re-)deployment in industrial context. Even if an offline programming method is used, refinement in the robot program is required and will cost time until the program is ready to be deployed. An actual survey from state-of-the-art indicated that the lack of HRC know-how, experiences and deployment skills are inhibitors in the deployment of HRC systems. Even though the participants of this survey are *de facto* robotic experts with years experience in the deployment of HRC systems, the results reveal that (re-)configuration of robotic with conventional programming methods is tedious, complex, abstruse and time-consuming (Hornung and Wurll (2022)). Consequently, it triggers a deficiency on productivity and cost efficiency.

Traditionally, robotic programming is categorized in online programming methods, such as traditional lead-trough and walk-trough and offline robotic programming methods, using software tools as the replacement of the real robot system [(Hägele et al. (2016)]. In order to achieve simplification in robotic programming, low- or no-code robotic programming systems are developed. Different novel approaches based on various sensor technologies e.g. 3D tracking system, Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (XR) and motion capture systems, have emerged over the years. Hence, human natural communication modalities substitute prior knowledge of syntaxes and semantics in robotic programming. This concept is known as Programming by Demonstration (PbD) (Billard et al. (2008)) and is also known as Learning from Demonstration (Argall et al. (2009); Lee (2017); Ravichandar et al. (2020)). This approach aims to enable non-robotic experts to teach their robots by demonstrating the desired robots' behavior or movement in the context of the production process.

Since no expertise to understand a specific robotic programming language is required from the end-user side, robot learning algorithms or strategies are developed to enable the robotic system to understand natural human communication modalities. Thus, it is essential to consider the technological aspects and human-centric issues such as usability and intuitiveness of the interaction between the human and the system. In order to capture, interpret, and understand human instructions accurately and robustly in the context of industrial processes, a novel approach for no-code programming by combining voice and hand gestures is proposed in this work. This combination enables a natural way for humans to interact with the robotic system. As a result, the robotic program can be deployed fast and agile in different industrial scenarios with different robotic systems by applying the proposed architecture in this work. The following section will present an overview of the state-of-the-art. Section 3 will introduce the proposed approach in detail, while section 4 will discuss the implementation of the proposed system. Section 5 will focus on the analysis of the implemented system. Finally, the last section will focus on the conclusion and a short outlook on potential future work.

2 Related works

The programming process entails providing a robot with a new ability to understand the state of the environment and perform actions that advance the system towards a process context. Conventionally, the online programming methods use a teach pendant to move a robot through the desired motion profile by jogging. The robot movement is stored in the robot controller and can be retrieved later. Even though the method seems to be simple and demands less expertise, online programming is suitable for simple repetitive tasks, e.g. industrial processes with simple movement profiles and geometric workpieces. When changes occur, adaptation to the robotic program is required. Hence, this approach is only suitable for production with large lot sizes. The frequent reconfiguration is tedious, unaffordable and time-consuming for small and medium enterprises with smaller batch sizes (Dietz et al. (2012)).

Offline robotic programming methods are deployed to replace the online robotic programming methods (Neto and Mendes (2013)). In offline programming methods, a virtual environment representing the robot work cell is created to program the robot's behaviour and motion. The robot programmer can generate a robot program off-site *via* offline programming methods. Hence production downtime can be avoided during the programming phase. Extendable functions for robotic programming, e.g. path planning and control system for complex production processes, are embedded in most offline programming tools (Beck et al. (2021); Funes-Lora et al. (2021)). A virtual robot controller (VRC) simulates the exact robot behaviour for a specific robot platform in the virtual environment. In many cases, the virtual environment mismatches the environment. For high-precision applications, adjustments in the robotic program must be performed to eliminate the deviations in transferring the robot program to the actual robot controller (Angelidis and Vosniakos (2014)).

With the rise of collaborative robots, the perspective of robotic programming shifted in the last decade. Safety and ease of use are crucial factors in developing collaborative robot systems. In many collaborative robot systems, hand-guiding control methods are deployed to accelerate robotic teaching compared to traditional methods (Massa et al. (2015)). In the PbD context, teaching *via* hand-guiding control is used to demonstrate the robot behaviour using a kinesthetic teaching process. Hand-guiding control is specified in actual standards of industrial robotic systems (DIN ISO/TS 15066 (2017); DIN EN ISO 10218-1 (2021); DIN EN ISO 10218-2 (2012)). In recent years, hand-guiding controls have been implemented in many industrial applications, e.g. robotic gluing (Iturrate et al. (2021)), assembly (Liu et al. (2021)), polishing (Kana et al. (2021)), welding (Zhang et al. (2019)), surface cleaning (Elliott et al. (2017)), Pick-and-Place or manipulation (Peng et al. (2018)). Despite the ease of hand-guiding teaching methods, these hand-guiding demands medium to high physical workload to move the robot joints. To improve users' ergonomics, algorithms, e.g. gravity compensation and variable stiffness, are developed to reduce the workload in kinesthetic teaching (Infante and Kyrki (2011); Wrede et al. (2013); Tykal et al. (2016)). The compensation algorithms mentioned above utilize dynamic parameters of the robotic system. In the implementation, this information is inaccessible to the robot manufacturers. The accuracy of the taught robotic path *via* kinesthetic teaching depends on the dexterity of the end-user. Hand tremor and lack of force in programming affect the quality and the precision of the robot path (Massa et al. (2015)). In order to compromise the physical workload in the kinesthetic teaching process, the teleoperation concepts are introduced where the users can manipulate the robot in real-time by using their gestures or body movements. In general, the teleoperation approaches are performed by utilizing different type of haptic sensors such as mid-air haptic devices (Du and Zhang (2014)), electroencephalograms (EEGs) (Yang et al. (2018a)) and joysticks (Sanchez-Diaz et al. (2019)).

Strategies such as teleoperation, observation and imitation are used to transfer human knowledge into robotic platforms. Vision-based systems, speech recognition systems, AR, VR and XR technologies are developed to accelerate low-code or no-code robotic programming methods (El Zaatari et al. (2019); Villani et al. (2018)). In low-code programming methods, adequate know-how in a robot programming language is still required. As a result, the reconfiguration of the robot program is time consuming. Compared to low-code programming, no-code robotic programming eliminates the barriers by allowing the user to interact with or move the robot using natural interactions, e.g., voice, gesture or haptic. In recent works from state-of-the-art, vision-based systems are exploited in many intuitive programming methods due to the capabilities of vision systems in environment recognition, object recognition and gesture recognition. In (Zhang et al. (2020c)), a novel approach for robot path teaching is developed using a marker-based vision system with a single RGB-D camera. The movement of the marker is tracked with the RGB-D camera and transferred into a motion planner. In the recent works (van Delden et al. (2012); Akkaladevi et al. (2019, 2020); Ajaykumar et al. (2021)), several works address intuitive programming approaches *via* vision systems for specific processes such as Pick-and-Place and assembly. In (van Delden et al. (2012)), a multimodal teaching approach *via* gesture and voice is developed for the Pick-and-Place application. This approach allows the user to select the objects and target position for the manipulation process by using a deictic finger gesture. Hence, a voice command is given to the robot to pick or place the object. An intuitive programming approach by demonstration is developed in (Akkaladevi et al. (2020)). This approach uses a multi-camera setup to track the assembly tasks performed by the user. The human actions and assembly objects will be tracked and used to build a knowledge representation of the assembly tasks, which will be sent to the robot system. In (Ajaykumar et al. (2021)), a marker-based programming strategy is developed by using objects with markers for the Pick-and-Place scenario. The robot path is created by manipulating the objects. The object movement will be tracked and converted as a robot program.

The emergence of AR/XR/VR technologies has influenced the programming strategies in HRC. In Akkaladevi et al. (2019), lighthouse sensors are used to demonstrate the user movement in a complex assembly process with screwing actions. A programming device is created by combining the lighthouse sensors for spatial tracking and force and torque sensors to measure the required torques for the screwing process. A combination of a vision-based system with augmented reality technology is introduced in (Lambrecht et al. (2013)). The augmented reality system allowed the teaching of robot paths by manipulating spatial objects with hand gestures. Other approaches with augmented reality technology are developed in (Soares et al. (2021); Blankemeyer et al. (2018); Bolano et al. (2020)). In (Soares et al. (2021)), a Microsoft HoloLens 2 1 is to develop an augmented reality environment. This environment enables the users to interact with the robot by drawing the robot path with their fingers. Afterwards the teaching process, the robot path is transferred into the robot system. In (Blankemeyer et al. (2018)), an intuitive programming approach for the assembly process is performed in an augmented reality environment. A representation of the assembled object is built in the virtual environment and the assembly process with the virtual object is demonstrated. Hence, this information will be transferred to the robot to execute the assembly task. In (Bolano et al. (2020)), an offline programming method in a virtual reality environment is developed. The robot trajectory can be generated by manipulating the virtual robot. Hence, the trajectory will be sent to a graphic interface to be executed in a real robot. *Via* the graphic interface, the movement sequence can be configured.

Besides using one modality to perform intuitive robot programming, more interactions can be used to increase the acceptance and comprehensibility of the teaching process. In (Liu et al. (2020)), a programming approach with the combination of

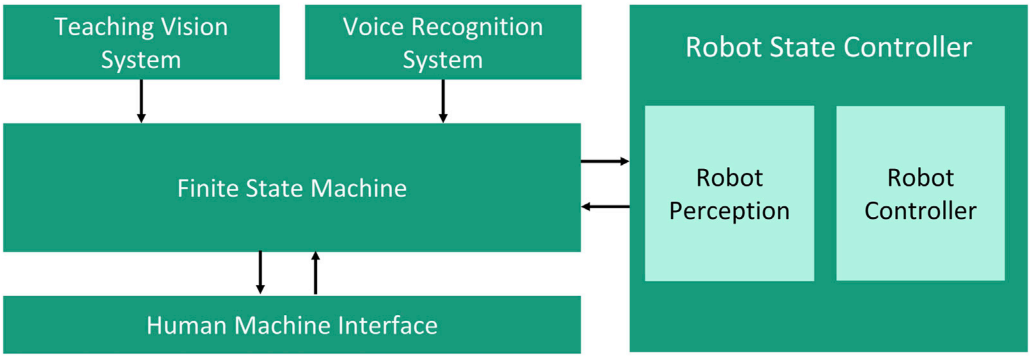


FIGURE 1
Proposed system architecture for multimodal no-code robotic programming.

sensorless haptic interaction, voice instructions, and hand gesture commands is used in an assembly scenario. The voice system helps the user to move the robot TCP. The hand gesture can perform the fine adjustment of the robot's position. Hence, the defined function blocks for the assembly and manipulation system can be triggered *via* voice instructions. In (Tirmizi et al. (2019)), a multimodal programming approach with a voice and vision system is developed for the Pick-and-Place scenario. The voice recognition system is utilized to control the system state. A vision-based object recognition system tracks the objects and delivers their coordinates that can be used for the manipulation process. In (Strazdas et al. (2022)), a multimodal system with a gesture, speech and gaze recognition system is developed for the Pick-and-Place scenario. The face and gaze recognition system monitors the interaction context with the system. The voice recognition system is used to control the robot's state. *Via* deictic gestures, the interaction objects can be chosen. In the recent multimodal programming approaches, a voice recognition system is integrated to navigate and control the system state. A recent study proved that a voice input system could accelerate robot programming up to two times in comparison to using traditional input devices (e.g., keyboards, teach pendants) (Ionescu and Schlund (2021)).

3 Methods

3.1 Proposed architecture

3.1.1 System architecture

The proposed system architecture consists of five modules which are depicted in Figure 1. The modular system design allows each functionality to be encapsulated as a subsystem. As a result, the highest degree of flexibility can be achieved in the system. The modular system architecture allows a better comprehensibility of the source codes, the simplification of the problem solving and the fast integration of new functionalities (Zirkelbach et al. (2019)).

A combination of hand- and finger-gestures with speech is proposed in the system architecture to allow a natural interaction in the teaching process of the robotic system. In comparison to low-code programming, no-code robotic programming method *via* multimodal interaction allows the user to create a robot program without particular expertise in robotic programming language. The robot program can be (re-)configured just by using interaction modalities that human does to communicate with each other. In this work, the proposed no-code programming is implemented by recognizing the hand- and finger-gestures *via* teaching vision system and recognizing user input *via* voice in the speech recognition system.

A camera-based vision system is developed to track and recognize the user's hand- and finger gestures in the teaching phase. The coordinates of the hand- and finger gestures are tracked and processed with computer vision algorithms to estimate the spatial pose in defined coordinate system. The coordinates of the hand or finger will be recorded based on the given commands and will be used to generate a robot path after the teaching process. This information will be converted into a specific robotic programming language before being transfer into the robotic system. The robotic system is equipped with a camera system as a perception module for executing the given robot path. Camera systems are considered in the proposed approach due to their benefits in comparison to other motion capture technologies such as (e.g: IMU- and VR systems). In general camera systems are markerless, easy to use, easy to set up, and affordable. In recent years, many reliable algorithms have been developed and shown potential to improve the camera system's performance, even compensating for their drawbacks (El Zaatari et al. (2019)).

The voice recognition system works as a complement to the teaching vision system to configure the system states and

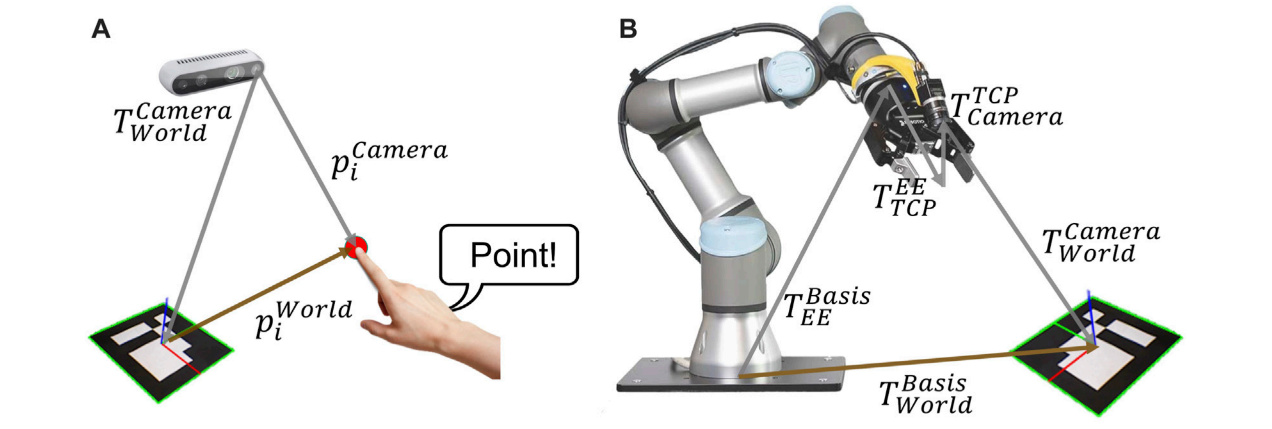


FIGURE 2
(A) transformation chain for the *i*th point of the robot path from programming process related to the target coordinate, **(B)** transformation chain for robotic perception system from robot base to target coordinate system.

parameters. In this work, the speech recognition system will process the user voice into text *via* Text-To Speech (TTS). Hence, the articulation of the voice command will trigger a deterministic action in the finite state machine. When a user says "take point," the actual coordinate of the finger will be extracted in the robot path. *Via* voice recognition system, efficiency in robotics programming is achieved by eliminating unnecessary user interactions *via* traditional human-machine interfaces (HMIs), e.g. buttons, keyboards, and mouse clicks. A recent study showed the potential of a speech recognition system to improve time efficiency in human-computer interface up to three times (Ruan et al. (2016)). A graphical HMI is developed to give the user visual feedback of the system. The HMI can be used as a redundant input system when the speech recognition system fails due to transient environmental noises.

3.1.2 System requirement

The system requirements for the proposed approach are depicted in Tables 5, 6. These system requirements must be fulfilled to enable fluent, stable and satisfactory interactions in the proposed robotic teaching process.

3.2 Teaching vision system

A vision-based teaching system is proposed for the main interaction modality of the novel teaching method. In Figure 2, the transformation chain for the programming process and robotic perception system are shown. For the proposed programming method, the world or target coordinate system is implemented by using an ArUco marker (Garrido-Jurado et al. (2014)). In comparison to other fiducial markers, e.g. ARTag, STag. ArUco marker guarantees high-precision position detection even in the noisy environments and utilizes low-computational power (Zakiev et al. (2020); Kalaitzakis et al. (2020)).

Figure 2A shows the transformation chain of the actual index finger's coordinates in the teaching process. The finger coordinates are captured from the camera system in the pixel coordinates. Hence, the finger coordinates are transformed in Cartesian coordinate with respect to the target coordinate system by using direct linear transformation. As a result, the target coordinate $\mathbf{p}_i^{Target}$ can be expressed with Eq. 1.

$$\mathbf{p}_i^{Target} = \left(\mathbf{T}_{Target}^{Camera}\right)^{-1} \mathbf{p}_i^{Camera} \tag{1}$$

Figure 2B shows the transformation chain for the homogenous transformation from base to target coordinate system $\mathbf{T}_{Base}^{Target}$ for the robot path. This transformation chain can be mathematically formulated using the equation in (2) and will be discussed in 3.2.1.3.

$$\mathbf{T}_{Base}^{Target} = \mathbf{T}_{Base}^{EE}\,\mathbf{T}_{EE}^{TCP}\mathbf{T}_{TCP}^{Camera}\mathbf{T}_{Camera}^{Target} \tag{2}$$

3.2.1 Hand- and finger-gesture recognition system

3.2.1.1 Hand- and finger-tracking

From the state-of-the-art, machine learning based hand- and finger-tracking SDKs are MediaPipe (Zhang et al. (2020b)), OpenPose (Simon et al. (2017)), AWR for hand 3d pose (Huang et al. (2020)) and MMPose (MMPose-Contributors (2020)). The mentioned SDKs allow hand- and finger-tracking by using RGB-image as input. Compared to the traditional computer vision-based algorithms, machine learning-based hand- and finger-tracking algorithms deliver better performance tracking under different lighting conditions, reflections, skin colours, and transitions over background objects with colour as human skin. The traditional computer vision tracking algorithm generally converts the input RGB image into another colour space. Classification is performed by defining the tracking colour constraints concerning the tracked object characteristics. As a result, unexpected objects will not be recognized. For example, a hand-gesture recognition system based on HSV colour space was implemented for an automatic handing-over system between heavy-duty and human co-workers (Bdiwi et al. (2013b)). This computer vision-based algorithm showed limits when tracking hand over reflective objects or objects with colour as human skin.

The main essential aspects for choosing the hand- and finger tracking SDK are the tracking performance based on the frame rate (FPS) and robustness under different light conditions. Besides, the specific hand model and its key points (landmarks) are considered for this proposed method. In experiments, MediaPipe constantly delivered 30 FPS with CPU computing. On the other hand, OpenPose delivered only 5 FPS with CPU computing. Even though the 2× up to 3× frame rate can be reached using GPU, it was not sufficient to provide fluent interaction for the proposed method. MediaPipe utilizes a hand model with 21 key points as shown in Figure 13. The index finger's tip (landmark 8) is tracked and used as a reference for the position in the teaching process. The finger's orientation is derived by calculating a Rodrigues vector between two landmarks in the index finger (landmarks 8 and 7). As a result, a robot path can be created by drawing splines or depicting singular points in the teaching process. It should be taken into account that the inaccuracies of the finger orientation calculation can occur due to the camera's limited field of view and perspective.

3.2.1.2 Pose estimation of the finger landmark

Assuming that the camera is a pinhole model, a direct linear transformation is used to obtain a projection of a point of interest in the target coordinate system (3D) into the pixel coordinate system (2D) or vice versa. Eq. 4 describes the transformation for rectified image. In this equation, s is the scaling factor, u and v are the coordinates of a point of interest in pixel coordinate. The intrinsic parameters of the camera are characterized by f_x, f_y, c_x, and c_y. f_x and f_y are the x- and y-axis focal length of the camera in a pixel unit. c_x, and c_y are the x- and y-axis optical center of the camera in a pixel unit. X_c, Y_c and Z_c are the coordinates of the point of interest in the camera coordinate system. By using a homogenous transformation matrix between the camera and target $\mathbf{T}^{Camera}_{Target\,(4x4)}$, the coordinates of the point of interest in the camera coordinate system are decomposed into coordinate points in the target coordinate system (X_w, Y_w and Z_w). The transformation matrix between camera and target is mathematically formulated with Eq. 3.

$$\mathbf{T}^{Camera}_{Target\,(4x4)} = \left[\mathbf{R}^{Camera}_{Target\,(3x3)} \| \mathbf{t}^{Camera}_{Target\,(3x3)}\right] \tag{3}$$

with $\mathbf{R}^{Camera}_{Target\,(3x3)}$ the rotation matrix and $\mathbf{t}^{Camera}_{Target\,(3x1)}$ the translation vector. The rotation matrix and translation vector represent the extrensic parameters of the camera.

The target coordinate system in this teaching process is represented by ArUco marker. All the points taken in the robot path will be transformed into the target coordinate system. In general, the 3D-coordinate points of the landmark (finger) relative to the ArUco marker is calculated by solving (4) in target coordinate points. Assuming that the finger is moving in different planes in 3D, the scaling factor s in (4) is varied according to the current plane parallel to the camera sensor. Hence, s is equal to the depth information of the finger in the camera coordinate system z_{finger}. This information can be derived directly from the depth image of the camera. The spatial information of the finger on x- and y-axis of the camera coordinate are calculated by using the intrinsic parameters f_x, f_y, c_x and c_y as shown in (5). Since diagonal elements of the transformation matrix between camera and target $R^{Camera^{-1}}_{Target\,(3x3)}$ is always not equal to zero the inverse of this matrix can be performed normally.

$$\begin{aligned} s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \\ &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left[\mathbf{R}^{Camera}_{Target\,(3x3)} \| \mathbf{t}^{Camera}_{Target\,(3x1)}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \end{aligned} \tag{4}$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \mathbf{R}^{Camera^{-1}}_{Target\,(3x3)} \left(\begin{bmatrix} \frac{u_{finger} - c_x}{f_x} z_{finger} \\ \frac{v_{finger} - c_y}{f_y} z_{finger} \\ z_{finger} \end{bmatrix} - \mathbf{t}^{Camera}_{Target\,(3x1)} \right) \tag{5}$$

In this work, the camera image is already rectified and the intrinsic parameters are accessible from the SDK of the camera. Otherwise intrinsic calibration can be performed by using function in OpenCV (Qiao et al. (2013)) or another tool like MATLAB. The rotation matrix and translation vector with respect to the marker is calculated *via* extrinsic calibration. The calculation of the rotation matrix and translation vector can be performed by using Perspective-n-Point (PnP) pose computation using approach (Marchand et al. (2016)) or OpenCV function for estimating pose of the single ArUco marker.

3.2.1.3 Image processing of spatial information of the finger landmark

With the advent of the computer vision algorithm, significant improvements in the accuracy of the teaching system can be achieved by implementing proposed algorithms, which are shown in Figure 3. Since the resolution of the RGB and depth image are not the same, it is necessary to synchronize the depth

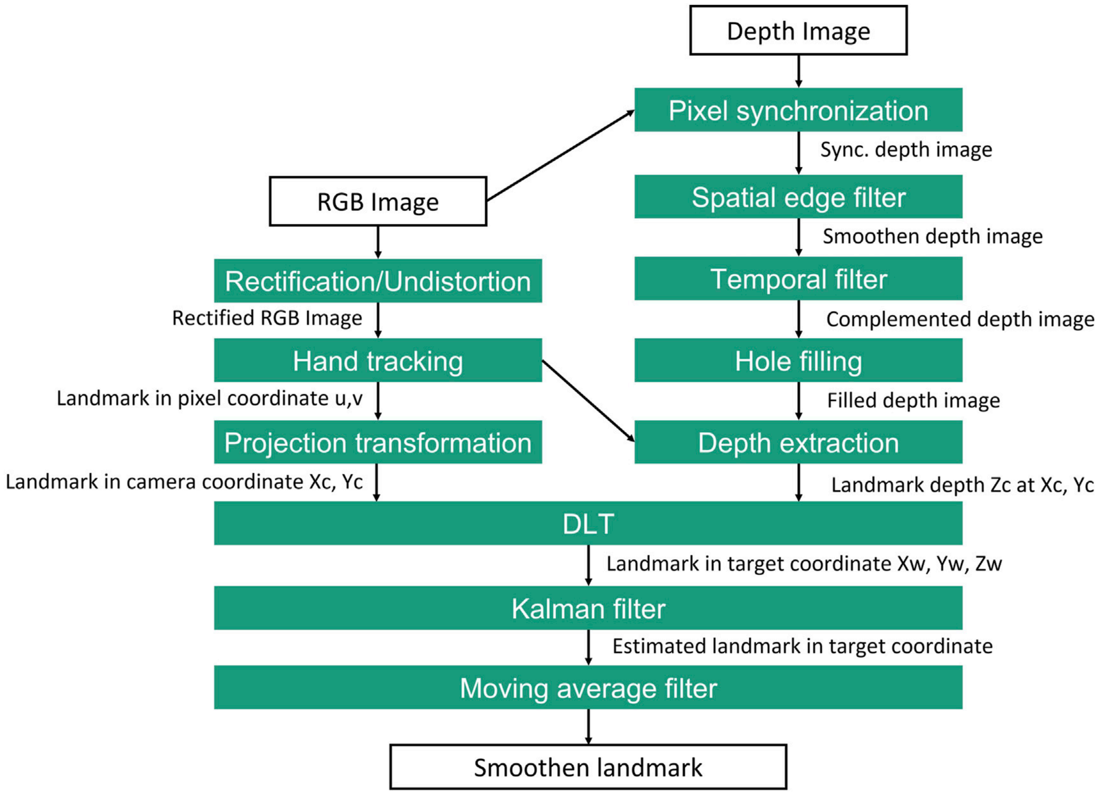


FIGURE 3
Proposed image processing method for extraction 3D coordinate of landmark for programming process.

image with the RGB image. Hence, the RGB image is rectified to correct the distortion in the image. The depth image processing is executed in parallel to the RGB-image processing. The spatial edge filter is used to enhance the smoothness of the depth reconstructed data by performing a series of 1D horizontal and vertical passes or iterations (Gastal and Oliveira (2011)).

A temporal filter is implemented to add the missing depth information when the pixel unit is missing or invalid. The data is processed in a single pass to adjust the depth values based on previous frames in this procedure. Hence, a hole-filling filter can fill the missing depth values using iteration based on the nearest pixel neighbours (Cho et al. (2020)). In the following step, the hand tracking method described in 3.2.1.1 is performed to obtain pixel coordinate u, v of the finger landmark. Simultaneously the transformation of the pixel coordinate into camera coordinate X_c, Y_c, and depth information Z_c extraction for the respected pixel unit of the finger landmark are performed. Then the landmark coordinate based on camera is fused and transformed into target coordinate X_w, Y_w and Z_w by using (5). Since the frame rate of the tracking system is limited to 30 FPS, stable hand tracking may not be available due to the fast movement of the hand. Therefore a Kalman filter is used to estimate the landmark position when tracking is missing or invalid in a short period. The kalman filter function from the OpenCV is utilized in this work. Finally, a moving average filter is implemented to smoothen the landmark position. The window size should be parameterized so that the filter does not cause any frame rate loss.

3.3 Voice recognition system

As already mentioned in 3.1.1, the voice recognition system is used to assist the end-user in changing the system state and parameter. The end user’s speech commands are extracted as text *via* Text-To Speech (TTS). After the feature extraction, the text is matched and proved with Natural Language Understanding (NLU) algorithm. In comparison to the traditional voice recognition system, NLU-based voice recognition system can deliver better performance and eliminate outliers with different voice characteristics (e.g., accents and voice profiles). In traditional voice recognition systems, the recognizer is built based on three models: *1*) acoustic models represent the acoustic signals of the voice, *2*) language models represent the grammars and semantics of the languages, *3*) lexicon models represent the phonemes and phonetics of word (Karpagavalli and Chandra (2016)). These models must be developed manually and

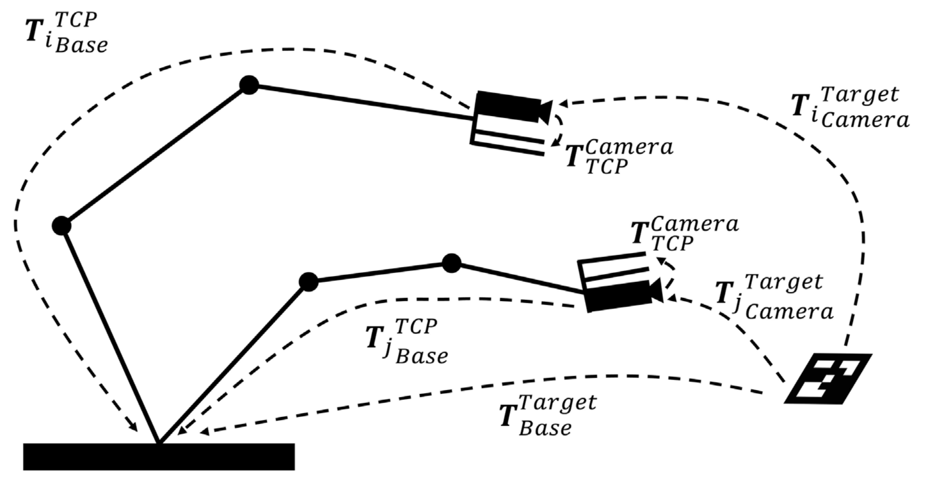


FIGURE 4
Hand-eye calibration problem: solving $\mathbf{T}_{Camera}^{TCP}$ using relative TCP and camera movements.

it is impossible to create a general model that can cover heterogeneous voice profiles of the speakers. NLU-based voice recognition systems use deep learning models based on trained data sets. With this approach, a better performance and more generic solution for voice recognition can be achieved.

3.4 Robot state controller

The robot state controller controls the behavior of the robot after receiving the generated robot path from the teaching process. The robot path from the teaching process is transformed to target coordinate system. The robot controller takes Cartesian coordinates at the robot base as reference for the robot movement. Therefore a coordinate transformation between the robot base and the target is performed With the assistance of a vision-based perception system.

It is sufficient to use the perception system to detect the target and apply the transformation with the target as the reference coordinate system for the robot. In other words, the robot movement is executed relative to the marker after the coordinate system transformation is performed. The transformation problem of the robot trajectory between robot base coordinate system and target coordinate system is accomplished by solving the equation of the transformation chain in (6).

$$\mathbf{T}_{Base}^{Target} = \mathbf{T}_{Base}^{EE}\mathbf{T}_{EE}^{TCP}\mathbf{T}_{TCP}^{Camera}\mathbf{T}_{Camera}^{Target} \tag{6}$$

The homogeneous transformation matrix from Base to EE $\mathbf{T}_{Base}^{EE}$ and transformation matrix from EE to TCP $\mathbf{T}_{Base}^{EE}$ is determined known by converting the TCP position from the robot interface into a 4 × 4 matrix. In order to obtain the transformation between the camera and TCP $\mathbf{T}_{TCP}^{Camera}$ the hand-eye calibration problem has to be solved by moving the robot into several positions. The resulting movements of the eye (camera) are observed as shown in Figure 4.

At this moment, the transformation matrix between the base and target $\mathbf{T}_{Base}^{Target}$ should be equal in each relative movement of the robot as mathematically formulated in (7).

$$\mathbf{T}_{iBase}^{TCP}\mathbf{T}_{TCP}^{Camera}\mathbf{T}_{iCamera}^{Target} = \mathbf{T}_{jBase}^{TCP}\mathbf{T}_{TCP}^{Camera}\mathbf{T}_{jCamera}^{Target} \tag{7}$$

By converting the (7) into (8), the transformation matrix of the target to the camera $\mathbf{T}_{Camera}^{Target}$ can be obtained using the pose estimating method (PnP) as described in 3.2.1.2.

$$\left(\mathbf{T}_{jBase}^{TCP}\right)^{-1}\mathbf{T}_{jBase}^{TCP}\mathbf{T}_{TCP}^{Camera} = \mathbf{T}_{TCP}^{Camera}\mathbf{T}_{jCamera}^{Target}\left(\mathbf{T}_{iCamera}^{Target}\right)^{-1} \tag{8}$$

In this work, numerical approach provided in OpenCV function is used to solve the hand-eye calibration problem. OpenCV provides five different calibration methods that differ in the order in which orientation and translation are estimated. In the following they will named after their authors and in line with the OpenCV documentation : Tsai (Tsai and Lenz (1989)), Park (Park and Martin (1994)), Horaud (Horaud and Dornaika (1995)), Andreff (Andreff et al. (1999)) and Daniilidis (Daniilidis (1999)). The results of our hand-eye calibration by applying the five mentioned algorithms above were converged. It means that the algorithms delivered the same results with minor offsets from each other.

3.5 Finite state machine

The finite state machine works as the main controller of the system. The speech commands are used as transition signals for the state machine. As a result, a deterministic action will be executed depending on the defined states in the state machine.

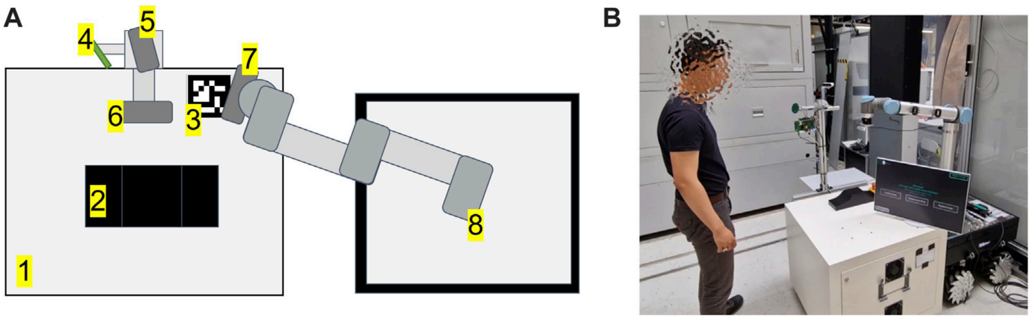


FIGURE 5
(A) setup of the proposed system. From 1 to 8: working table, workpiece, ArUco marker, Microphone array, Intel RealSense D415 - parallel to working table, Intel RealSense D435 - frontal to the user, Intel RealSense D415 - robot vision, Universal Robot UR10 CB3-Series, **(B)** setup in the real world.

Explicitly, the implementation of the finite state machine will be discussed more in detail in 4.3.

3.6 Human machine interface

To provide the user with feedback, a graphical user interface (GUI) was implemented. Information such as videos from the teaching and robot perception vision system, given speech commands, system parameters and statuses is represented in the GUI. The user interface serves not only as feedback, but also as a redundant input system. This is intended, for example, when the speech recognition system is not usable due to too intense ambient noise. Actual research showed that the relevance of user interfaces in hybrid human-robot systems can improve user acceptance and reduce mental workload (Bdiwi et al. (2021)).

4 Implementations

4.1 Setup

Figure 5 shows the experimental setup for the proposed multimodal programming approach in this work.

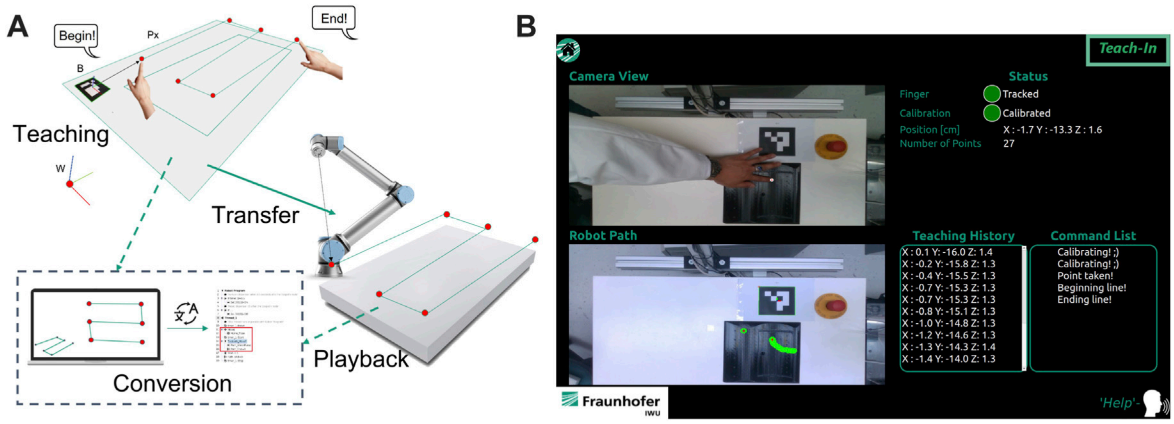
The hardware used in this setup has been fulfilled the system requirements suggested in Appendix I - system requirements. An Universal Robot UR10 CB-Series is used as the robotic platform (Robots (2015)). UR RTDE 2 is used as communication interface between an industrial PC and the UR10. Three Intel RealSense D400 Series cameras are used for the interaction process (Intel (2015)). One Intel RealSense D415 camera is placed parallel to the surface of the working table is used to capture the spatial information of the gesture during the teaching process, as mentioned in 3.2.1. The camera is located 64 cm above the table surface, delivering a 48 cm × 32 cm field of view. Since the field of view has linear correlations with camera height, putting the camera at a higher height would increase the field of view. All of the camera positioning is flexible and can be adapted depending on the required field of view. The second Intel RealSense D415 camera is mounted and calibrated with hand-eye calibration. This camera is used for robotic perception, as mentioned in 3.2.1.3. Finally, an Intel Realsense D435 camera is mounted facing the user frontally and used for teleoperation of the robot TCP *via* hand movements (gesture control). An ArUco marker is used as a reference for the finger-based teaching approach mentioned in 3.2.1. A NLU-based speech recognition module from voice INTER connect GmbH is used (voice INTER connect GmbH (2022)). This speech recognition module supports voice recognition with different languages, voice profiles (e.g: masculine or feminine), accents. It should be taken into account that all of the mentioned hardware devices are only tentative. The setup is flexible and may be changed depending on certain use case requirements. Different robotic platforms, cameras, and speech recognition systems should be compatible with the proposed approach, as the system is modular and uses standard interfaces.

4.2 Operation modes

Three operation modes have been implemented based on the proposed architecture mentioned in 3.1. These operation modes are:

1. Teaching mode
2. Teleoperation mode
3. Playback mode

In the teaching mode, the robotic program can be created by using index finger's gesture and voice recognition system. Teleoperation mode supports remote control of the robot by utilizing hand gesture and voice recognition system. The playback mode is used to replay the programmed robot path in the teaching mode. A graphical user interface is utilized to give



Camera View

Status

Finger ● Tracked
Calibration ● Calibrated
Position [cm] X : -1.7 Y : -13.3 Z : 1.6
Number of Points 27

Robot Path

Teaching History

X : 0.1 Y: -16.0 Z: 1.4
X : -0.2 Y: -15.8 Z: 1.3
X : -0.4 Y: -15.5 Z: 1.3
X : -0.7 Y: -15.3 Z: 1.3
X : -0.7 Y: -15.3 Z: 1.3
X : -0.8 Y: -15.1 Z: 1.3
X : -1.0 Y: -14.8 Z: 1.3
X : -1.2 Y: -14.6 Z: 1.3
X : -1.3 Y: -14.3 Z: 1.4
X : -1.4 Y: -14.0 Z: 1.3

Command List

Calibrating! ;)
Calibrating! ;)
Point taken!
Beginning line!
Ending line!

FIGURE 6
(A) illustration of the teaching process in teaching mode, **(B)** user interface of teaching mode.

feedback and instructions to the user, manually check system status and set system parameters.

4.2.1 Teaching mode

In teaching mode, index finger's gesture is utilized to create a robot path. By using the proposed algorithm in 3.2.1.3, the pose of the pointing finger in the teaching process can be estimated and recorded after the command is given. The voice recognition system is linked to the finite state machine and will trigger a defined action, if the command matches with the database in the context manager. As an example, command "take" triggers the state machine to extract the current pose of the finger as single robot path point. In Figure 6A, the teaching pipeline for the teaching mode and the implemented user interface are illustrated. After the teaching process is finished, the captured points are ready to be converted into robot paths in playback mode.

The implemented user interface provides real-time camera view for the teaching process and information regarding the created robot path. Additionally, information such as number of taken points, actual state of state machine, tracking status, calibration status and actual position of pointing finger are also provided *via* graphical user interface. Before the user interface of the selected operation mode is initialized, a tutorial video is played to explain to the user how the system works. If the user requires further assistance to use the system, a command list is accessible by giving a voice command "help." The implemented actions and voice commands for the teaching mode are:

- **Calibrate**: triggers the calibration process of the individual finger profile. It should be taken into account that finger profile of each user is varied. To compensate the ground truth effect, a calibration is performed in a defined time interval. Hence, the finger profile is registered as the offset in the pose estimation mentioned in 3.2.1.2.
- **Get**: triggers the extraction of the actual position of the index finger as a single point into the currently recorded robot path.
- **Begin**: initializes the extraction of a spline. The spline is created by demonstrating the path *via* the index finger's movement. Finger coordinates in each cycle time are extracted into the robot path until the stop command (**End**) is given. The recording process will be interrupted when the finger tracking is lost, and the taken points will not be registered in the robot path.
- **End**: ends the recording process of the spline.
- **Delete**: triggers the system to delete the latest taken object from the robot path. In this context, the object can be a single point or a spline.
- **Help**: triggers the system to show a command list for all available commands and their definitions.
- **Home**: stops the teaching mode and initialize the main menu (idle).

4.2.2 Teleoperation mode

In the teleoperation mode, the user can teleoperate the robot using hand gestures. A voice command is used to start the interaction. After initialization the initial position of the hand is registered and a bounding box is displayed on the feedback interface, representing the initial position of the user's hand. The relative position of the hand to the initial position (bounding box) is calculated and used to manipulate the robot TCP in 3D. Additionally, manipulation of the robot arm's single or multiple axes is possible. Figure 7 shows the interaction workflow, and graphical user interface for teleoperation mode.

The registered commands for teleoperation mode are:

- **Lock**: triggers the system to register the initial position of the user's hand for the TCP manipulation.

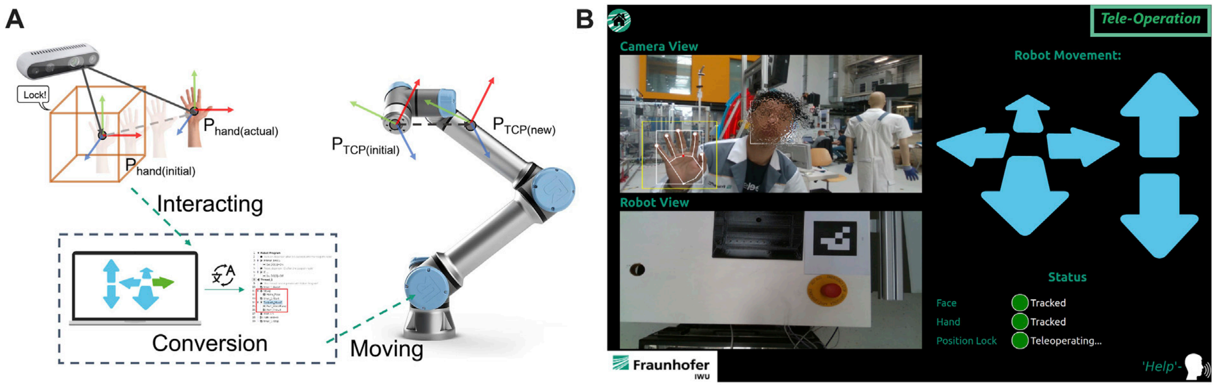


FIGURE 7
(A) illustration of the interaction process in teleoperation mode, **(B)** user interface of teaching mode.

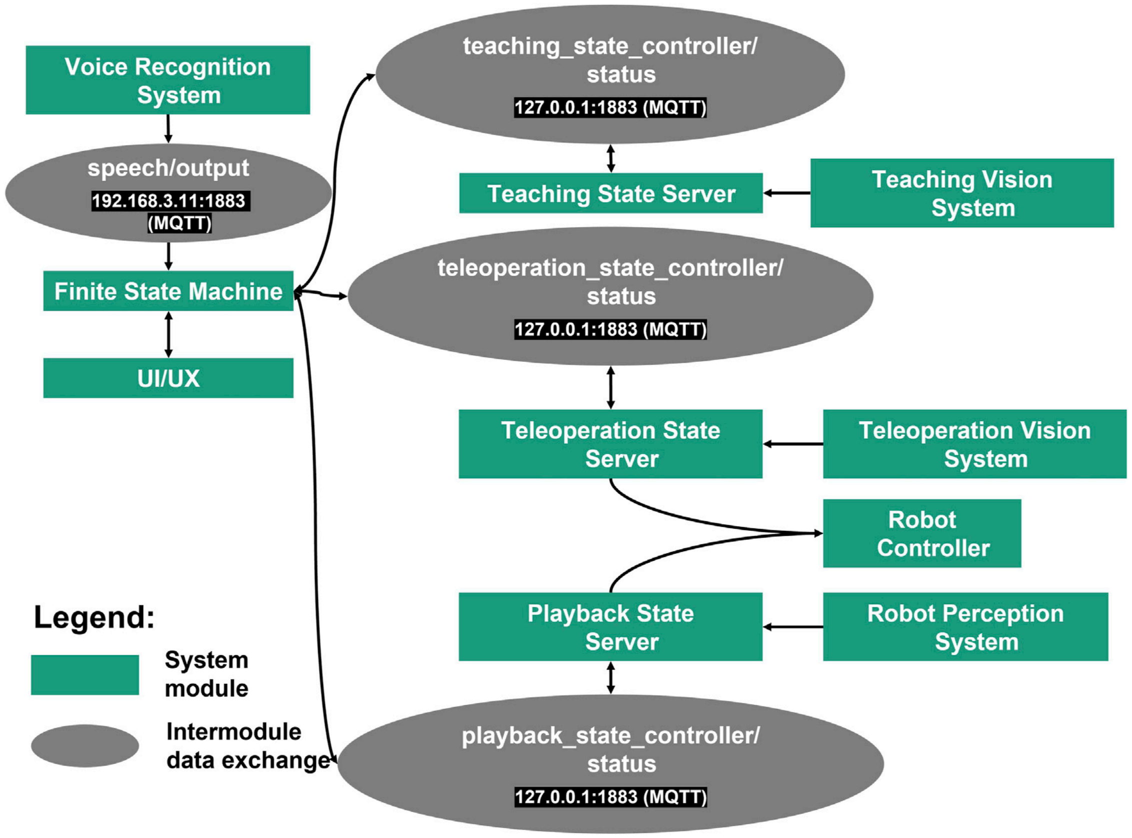


FIGURE 8
Implemented system diagram.

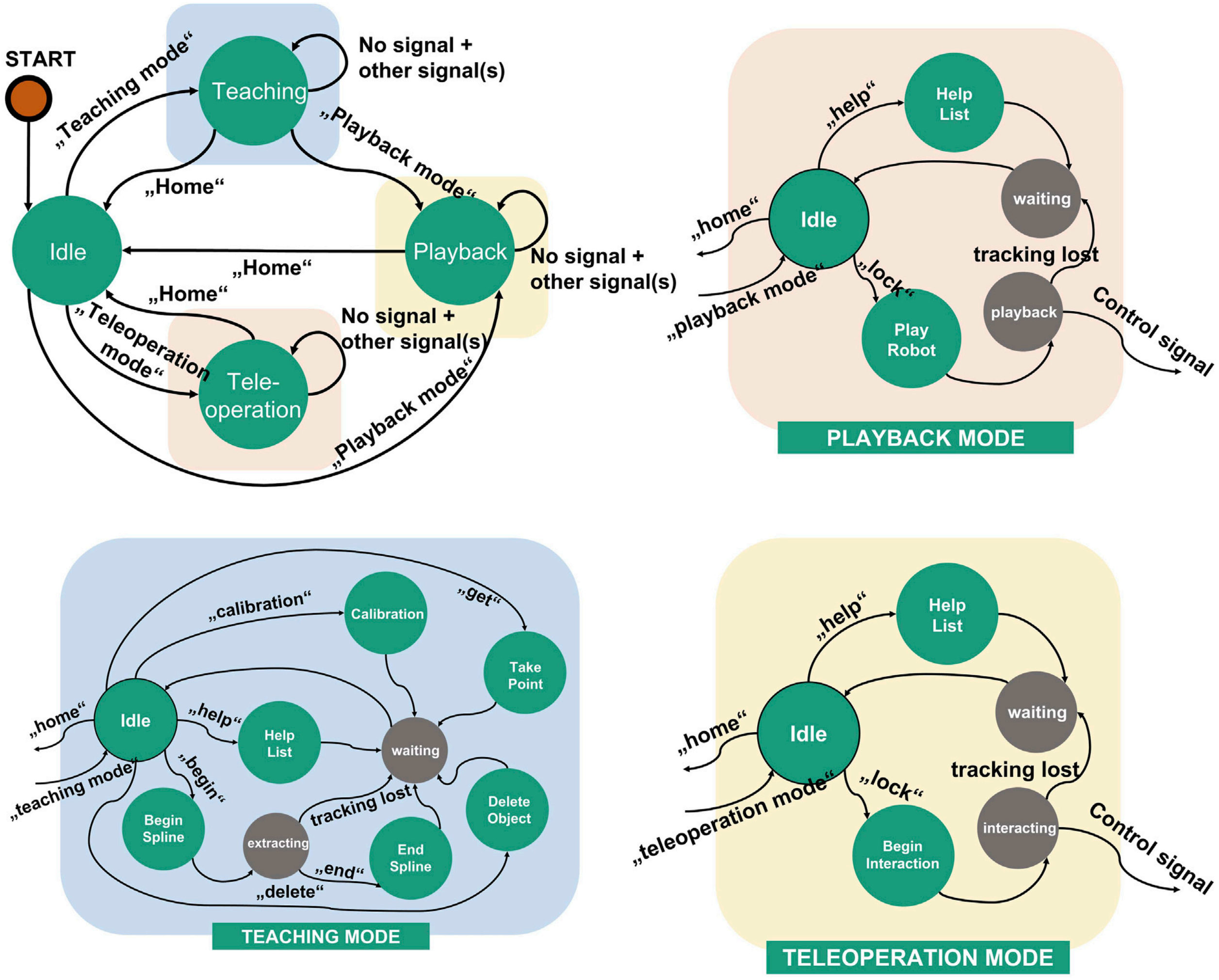


FIGURE 9
Finite state machine of the proposed system and its sub-finite state machines for each operation mode.

- **Help**: trigger the system to change the manipulation mode of the system from translation into rotation or vice versa.
- **Help**: triggers the system to show a command list for all available commands and their definitions.
- **Home**: stops the teaching mode and initialize the main menu (idle).

4.2.3 Playback mode

In the playback mode, the robot path created *via* teaching mode can be converted into robot specific language and further parameterized. After the "play" command, the robot path is automatically converted into a specific robotic programming language and deployed to the robot controller. Parameters such as robot speed, interpolation parameters and blending parameters are configurable *via* voice command.

4.3 System diagram and finite state machine (FSM)

The implemented system diagram is shown in Figure 8. To achieve system modularity, the operation modes and other functionalities are encapsulated as system modules. For intercommunication between each module Message Queuing Telemetry Transport (MQTT) protocol was used to guarantee robust information exchange (Standard (2014)).

A finite state machine allows complexity reduction in the deployment of the robotic system (Balogh and Obdržálek (2018)). Therefore, a finite state machine is used to integrate and control all modules. Figure 9 shows the finite state machine of the whole system and its sub-finite state machines. Each operation mode mentioned in 4.2 is encapsulated as system module containing a subordinate finite state machine. Each

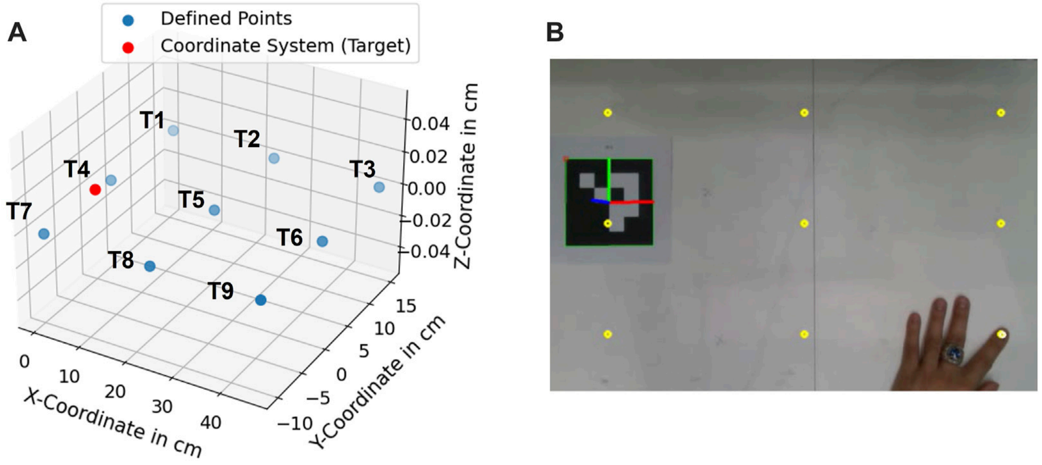


FIGURE 10
(A) defined coordinates (*T1*, . . ., *T9*), **(B)** pointing experiment at defined target points.

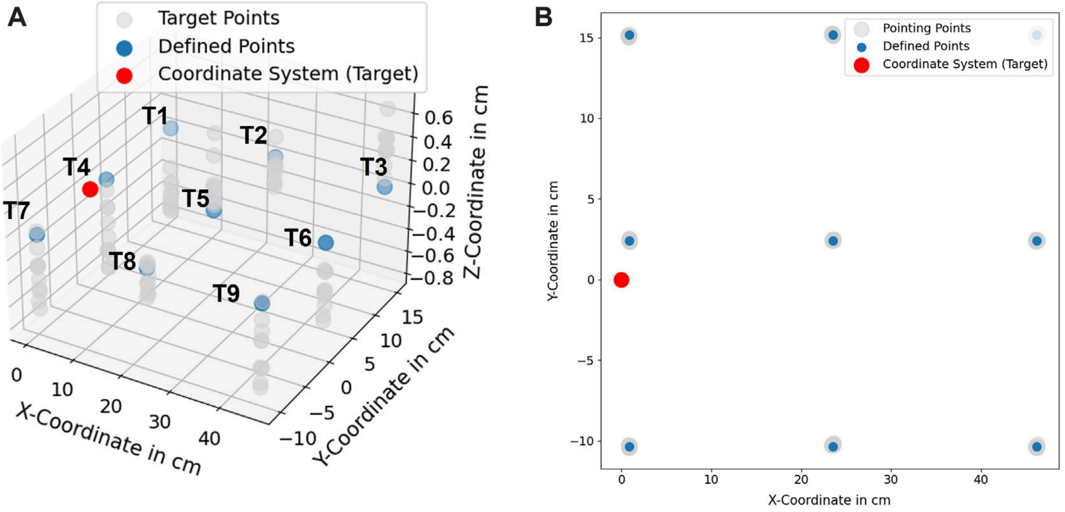


FIGURE 11
(A) Scaled position deviation at defined coordinates (T1, ..., T9), **(B)** 2D-View of scaled position deviation at defined coordinates.

module contains sub-modules that support the functionality of the system module for each operation mode, e.g. for the vision system and robot control. The teaching state server, teleoperation state server and playback state server receive a bypass information from the finite state machine when the respected operation mode is triggered. The bypass information is used as transition signal for each sub-finite state machine in each operation mode. In teleoperation mode and playback mode, a control system signal is sent to the robot immediately after it is triggered by interactions. The finite state machine shown in Figure 9 represents the implementation of the proposed system in this work. In the implementation, three operation modes are implemented by utilizing hand gestures, finger gestures and speeches as interaction modalities. Since the system is modular, each extension or customization in the system architecture will affect the finite state machine. In case of extension with additional systems and functionalities, the states and signals must be extended.

TABLE 1 Measurement uncertainty for accuracy measurement at each target point.

Point	Euclidean deviation $\overline{P_i}$ $[mm]$	Standard deviation σ_{P_i} $[mm]$
T1	6.43	0.96
T2	1.99	2.25
T3	3.63	1.77
T4	2.52	1.45
T5	2.44	1.72
T6	1.89	0.61
T7	3.49	1.57
T8	4.44	1.41
T9	3.88	2.15
Σ	3.71	2.07

5 Results

5.1 Accuracy and precision assessment

In order to assess the accuracy of the proposed hand- and finger tracking algorithm in 3.2.1.3, a pointing task was defined as in Figure 10. In this task, nine target coodinates ($T1, \ldots, T9$) were predefined and should be pointed as accurately as possible 10 times at each point. Afterwards, the average position deviation in cm $\overline{P_i}$ was calculated by using euclidean norm for position deviations for each axis (Δx, Δy, Δz) as shown in Eq. 9.

$$\overline{P_i} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{9}$$

The measurement was performed with camera height at 65 *cm*. The light intensity measured in the environment was 580 *Lux* at 1,5 *m* above the floor and the temperature was at 21°*C*. In Figure 11, the measured coordinates are compared with the defined coordinates in 3D and 2D. As a result, the spatial information of the pointed coordinates at the *z*-axis is more inaccurate in comparison to the information at the x- and *y*-axis. The inaccuracy is caused due to the noise from the depth information obtained from the camera. From the technical specification of Intel RealSense D415, the depth accuracy from the camera is 2% < 2*m* (Intel (2015)).

A recent study for the performance of Intel RealSense D415 showed that for the short distance 500–1000 *mm*, the camera delivers up to 30 *mm* deviation in depth estimation (Servi et al. (2021)). From the obtained results, it can be concluded that the accuracy of the proposed method achieves 3.71 ± 2.07 *mm*. The statistical analysis of each target point is shown in Table 1.

The resulting deviations in the system can be caused by several factors. A human can not point a target point accurately with its finger, caused by the anatomy of the human finger. This uncertainty can be varied in the range of *mm* and *cm* depending on the human hand-eye coordination skill or the dexterity of the user. A further observation was performed to assess the systematical deviations (precision) from the proposed algorithm in 3.2.1.3. A new assessment task was formulated. In this task, nine target coordinates T1. . .T9 were used. A finger was pointing to these points, and the finger was maintained to be static while the finger's position was being recorded. In Figure 14, standard deviations of the measured points at the x- and *y*-axis are shown with 95% confidence ellipsoid to give an overview of the system precision (*See* 95% confidence ellipsoid in 6 for reference). Standard deviation in the *z*-axis is also shown in Figure 15. Standard deviation in x-,y- and *z*-axis (σ_x, σ_y, σ_z) and standard deviation of Euclidean distance in 2D (σ_r) are represented in Table 2.The result showed that the tracking deviation at the x- and *y*-axis are smaller than the deviation at the *z*-axis. In each target point, the planar deviation is less than 1 *mm*. The deviation of the depth information is less than 2 *mm*. The deviations existed due to the inaccuracy in the intrinsic and extrinsic calibration of the camera system. The higher deviation in depth information indicated that the camera delivers inconsistent depth information. Despite the higher deviation in depth information, the result showed that the proposed

TABLE 2 Standard deviation of the tracking system precision.

Point	Number of Points	σ_x $[mm]$	σ_y $[mm]$	σ_r $[mm]$	σ_z $[mm]$
T1	99	0.48	0.44	0.65	1.09
T2	99	0.28	0.39	0.49	1.27
T3	99	0.41	0.45	0.61	1.71
T4	99	0.54	0.30	0.62	1.01
T5	99	0.54	0.31	0.62	1.17
T6	99	0.46	0.59	0.75	0.89
T7	98	0.36	0.42	0.55	1.25
T8	98	0.45	0.44	0.63	1.83
T9	99	0.42	0.20	0.46	1.68

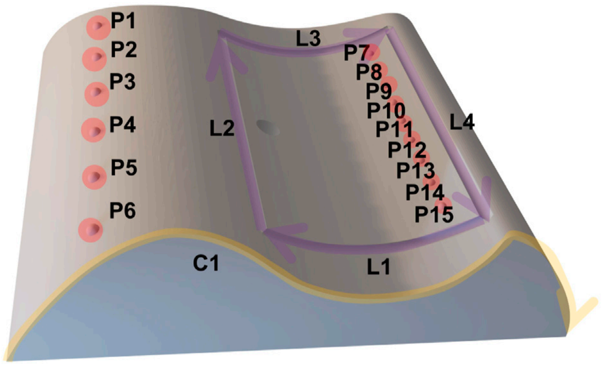


FIGURE 12
3D printed part for benchmarking assessment.

image processing algorithms mentioned in 3.2.1.3 can reduce the depth inaccuracy of the camera system. In conclusion, the assessment method shows promising results of the proposed method to be deployed for robotic programming applications with relative accuracy up to 6 *mm* and the tracking system can deliver up to 2 mm precision with the defined setup in 4.1.

5.2 Benchmarking with state-of-the-art

In order to show the practicability of the proposed method, a benchmarking is done by comparing the proposed system with the implemented methods from the state-of-the-art such as hand-guiding and programming by teach pendant in Universal Robots UR10 which are specified in the actual standards for industrial robot system [DIN EN ISO 10218-1 (2021); DIN EN ISO 10218-2 (2012); DIN ISO/TS 15066 (2017)]. This assessment is performed in a real-world teaching scenario for painting or gluing application in the real production. A workpiece as shown in Figure 12 was manufactured with specific features that would be used for the tasks in this assessment.

The features are 15 points (P1,. . .,P15), four lines with their directions (L1,. . .,L4) and a curve with its direction (C1). The tasks in this assessment consist of movement sequence based on these features. In total, four movement sequences with different complexity were executed by using the multimodal programming approach in this work. Each task will be repeated by using hand-guiding and online programming approach *via* teach pendant from Universal Robots

TABLE 3 Overview for effort reduction from the proposed method (rel. Reduction (PM)) in the benchmarking assessment, TP, teach pendant; HG, hand-guiding; PM, proposed method.

Parameter	Task 1			Task 2			Task 3			Task 4		
Method	TP	HG	PM	TP	HG	PM	TP	HG	PM	TP	HG	PM
Mean time [s]	92.73	71.08	28.69	118.35	101.63	50.28	71.19	70.38	24.34	65.47	56.28	17.71
Time ratio	3.23x	2,47x	—	2.35x	2.02x	—	2.92x	2.89x	—	3.69x	3.18x	—
Mean NoP	6	6	6	11	11	11	6	6	88	7	7	70.33
NoP ratio	1x	1x	—	1x	1x	—	0.06x	0.06x	—	0.09x	0.09x	—
Rel. Reduction (PM)	3.23x	2.47x	—	2.35x	2.02x	—	48.67x	48.17x	—	41.00x	35.33x	—

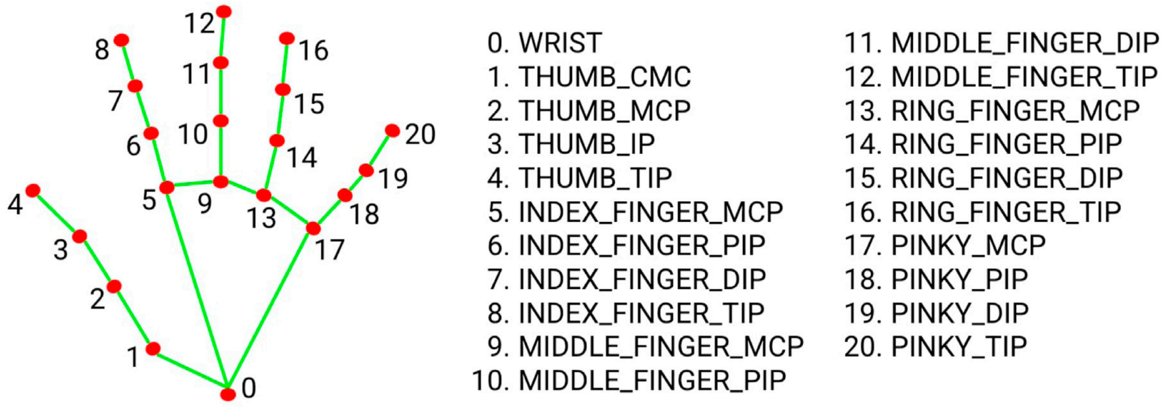


FIGURE 13
Hand model and key points of MediaPipe.

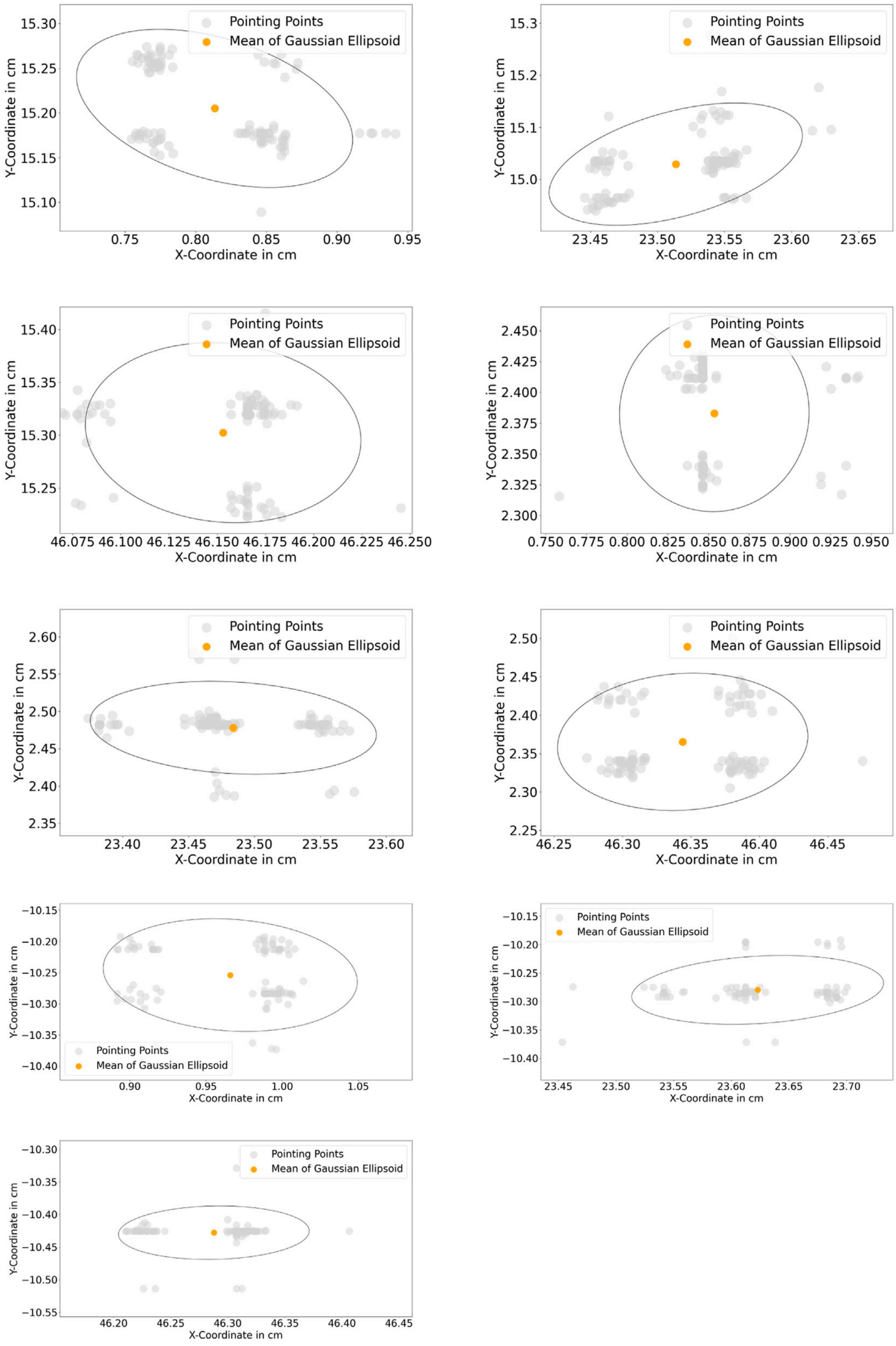


FIGURE 14
(Top left to bottom right): 95% gaussian ellipsoid for measurement of standard deviation of static points from T1. . .T9.

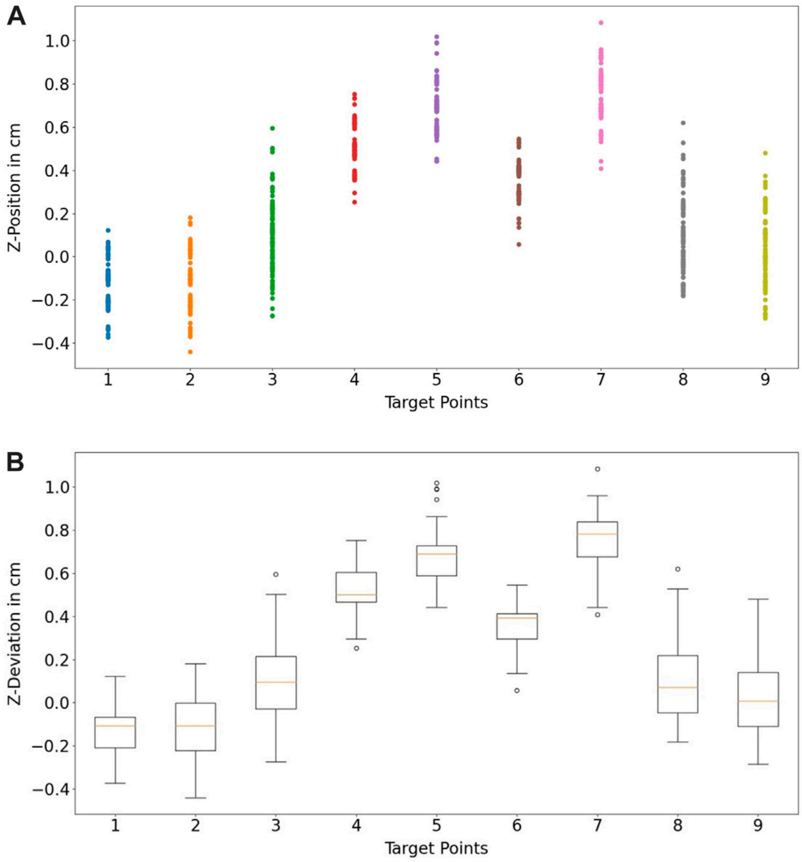


FIGURE 15
(A) depth information of the static point measurement at each target point, **(B)** standard deviation each static point in box plots.

UR10 controller. The number of points and execution time for each task are measured for the assessment. The tasks are described in following:

- Task 1: PTP linear movement ($P6 \rightarrow P5 \rightarrow P4 \rightarrow P3 \rightarrow P2 \rightarrow P1$)
- Task 2: PTP zigzag movement ($P6 \rightarrow P15 \rightarrow P5 \rightarrow P13 \rightarrow P4 \rightarrow P11 \rightarrow P3 \rightarrow P9 \rightarrow P2 \rightarrow P7 \rightarrow P1$)
- Task 3: Movement along defined features ($L1 \rightarrow L2 \rightarrow L3 \rightarrow L4$)
- Task 4: Movement along defined contour ($C1$)

The overview of the assessment result is depicted in Table 3 (detailed result in Table 7). For the assessment, the time ratio and number of point (NoP) ratio between teach pendant and hand-guiding teaching to the proposed method were calculated. The time ratio is calculated as the quotient of the mean time of the hand-guiding or teach pendant and the proposed method. For the number of point, the same normalization is performed by building quotient of number of recorded points for the programming methods from the state-of-the art and the proposed method. In the programming methods with teach pendant or hand-guiding, the user must determine how many points must be taken to extract the features of the work piece. In the proposed method, this issue does not exist because the finger's movement along the features is extracted in the teaching process. As a result, the selected features can be extracted as coordinate points in the proposed method. Therefore, the number of points as assessment criterion is necessary to give an objective benchmark in this assessment.

These ratios were used to calculate the relative reduction for the benchmarking using following equation:

$$Relative\ reduction = \frac{Time\ Ratio}{NoP\ Ratio} \tag{10}$$

For simple PTP motions in tasks 1 and 2, the proposed method showed effort reduction with 2–3× factor. In the experiments, speech commands had to be repeated several times in some cases, due to environmental noise ($> 60dB$). This led to longer teaching times. A backup solution to improve performance issues caused by environmental noise is considered by utilizing alternative

TABLE 4 Potential industrial applications for the proposed method with the estimated process tolerances and possible control strategies for development of skill based technology modules.

Applications	Tolerance	Skill based control strategy
Handing-over	⊘, ⊗	Bdiwi et al. (2013a,c)
Manipulation	⊙, ⊘, ⊗	Jokesch et al. (2014)
Painting	⊘, ⊗	Zhang et al. (2020a); Tadic et al. (2021)
Peg-in-hole	⊘	Bdiwi et al. (2015); Haugaard et al. (2020)
Polishing	⊘, ⊗	Tian et al. (2016); Kakinuma et al. (2022); Zhou et al. (2021)
Welding	⊙, ⊘	Yang et al. (2018b); Lei et al. (2021)

⊙ - fine ($x < 10\ \mu m$), ⊘ - medium ($10\ \mu m \le x \le 10\ mm$), ⊗ - coarse ($x > 10\ mm$).

TABLE 5 System requirement for the vision system.

Requirement of vision system		
Parameter	**Min. value**	**Description**
Camera RGB Resolution	1280 × 720 (HD)	Since the hand-tracking system algorithm works with RGB images as input the higher resolution offers better performance in tracking
Camera depth resolution	640 × 480	The higher depth resolution delivers higher details of in-depth information, otherwise, it costs a longer computation time to obtain this information
Camera RGB field of view	60° × 40°	The field of view (FoV) describes the possible detection area that the camera image could deliver. The higher FoV offers a wider detection area for the system
Camera depth field of view	50° × 30°	Since RGB and depth FoV are not exactly the same, a synchronization function to map both information should be performed to get more detail mapping of both information
Camera RGB/deph frame rate	30 FPS	A higher frame rate would cause less latency. In other words, a higher frame rate would improve the system response in image processing
Camera min depth information	x > 0.2 m	The actual stereo camera system has minimum depth information that can be obtained. The nearer the minimum available depth information the more accurate the calculation for the spatial information of the hand and finger tracking
Hand tracking model	All phalanx on handshould be available	The hand tracking system should deliver as many landmarks (phalanxes). Since the proposed approach with the deictic gesture of a pointing finger, the landmarks in the pointing finger should be able to be identified
Hand tracking stability	Stabile in all light conditions	The hand tracking system should work with RGB images in different lighting conditions
Hand tracking distance	0–1.5 m	The tracking system should work at a farther distance to compensate for the deficiency due to the minimum distance from the depth information of the stereo camera
Hand tracking frame rate	30 FPS	The higher the frame rate the hand tracking system could deliver, the more fluent the interaction between end-user and system could occur

input interfaces such as a keyboard or other peripheries. The results from tasks 3 and 4 showed drastic improvements in the generation of complex movement profiles, such as movement along specific features. By performing the task using the programming methods from state-of-the-art, the first hindrance was to consider how many points should be extracted to build a detailed movement profile along the desired feature. The programming effort was significantly improved when more points should be extracted. In contrast, even though less programming time can be achieved by reducing the number of points, the desired movement profile will be compensated due to adequate detailed information from the taken points. This drawback effect was shown in tasks 3 and 4 using hand-guiding and teach pendant. Hereby, less than ten points were taken to generate the movement profile. Eventually, the desired movement profile could not be fulfilled due to sufficient information on the desired feature. In comparison to the methods from state-of-the-art, the proposed method showed incisive results with 40–50× effort reduction for complex tasks such as tasks 3 and 4. In the proposed method, the desired feature can be extracted as a robot movement profile by tracking the finger movement on the corresponding feature directly. The proposed multimodal no-code programming approach showed the potential to drastically reduce the teaching time and effort for robotic programs compared to the state-of-the-art.

TABLE 6 System requirement for the voice recognition system.

Requirement of the voice recognition system		
Parameter	**Value**	**Description**
Recognition type	Offline	Since the recognition sytem is used in an industrial context, an offline voice recognition system is demandable to maintain data security
Dialogue design	Conformed based on ISO/IEC 30122	The dialogue should be designed as easily as possible as mentioned in ISO/IEC 30122
Dialogue extraction	Text-to-speech for every uttered words	The system should be able to extract single word in a sentece uttered by the end-user

TABLE 7 Benchmarking result with comparison at programming time and number of taken points (NoP), TP, Teach Pendant; HG, hand-guiding; PM, Proposed Method.

Task	Method	Time 1 [s]	Time 2 [s]	Time 3 [s]	Mean time [s]	NoP 1	NoP 2	NoP 3	Mean NoP
Task 1	TP	90.05	97.06	89.12	92.73	6	6	6	6
	HG	75.46	70.30	77.49	71.08	6	6	6	6
	PM	28.28	28.68	29.12	28.69	6	6	6	6
Task 2	TP	122.61	110.18	102.25	101.63	11	11	11	11
	HG	101.32	101.33	102.25	118.35	11	11	11	11
	PM	48.56	55.82	46.46	50.28	11	11	11	11
Task 3	TP	78.73	77.58	75.25	77.19	6	6	6	6
	HG	69.56	74.84	66.74	70.38	6	6	6	6
	PM	24,95	26,34	21.73	24.34	85	97	82	88
Task 4	TP	67.85	58.30	70.25	65.47	7	7	7	7
	HG	58.49	55.81	54.53	56.28	7	7	7	7
	PM	21.19	15.93	16.02	17.71	87	59	65	70.33

6 Discussions and conclusions

In many cases, the intuitive teaching methods from state-of-the-art are not ready to be implemented directly in an industrial environment. The proposed programming approaches from the state-of-the-art are mostly task-oriented and can be performed only to create a robot routine for a specific process. The system setups are fixed with strictly defined sensors, and there is no room for customization. Even though the proposed systems prioritize ease of use and consider intuitive interactions in the teaching process, many works are not implementable in industrial environments due to non-practicable methodologies and complex system configurations. These hurdles are antitheses to the concepts of HRC, which enables robotic systems to be agile, reconfigurable and adaptable when changes in production occur. This work proposes a novel approach to intuitive programming by utilizing multimodal interactions such as speech and gestures. The proposed programming approach introduces a generic teaching solution for HRC applications in agile production by utilizing low-cost sensors. The novel approach allows the user to (re-)configure the robot program in the scenario where major or minor changes occur in production.

Compared to state-of-the-art robotic programs, such as teach pendant and hand-guiding, the novel method proposes in this work showed that the programming effort for complex tasks can be reduced by 40–50 times. It also enables non-robotic experts to reconfigure and create robotic programs in a short time using multimodal interaction. With the approach robot paths can be taught by demonstration of finger gestures with 6 *mm* accuracy. The proposed computer vision algorithm for hand- and finger-gesture estimation has thus shown its capability to achieve a precision up to 2 *mm* in the observed environment. In comparison to alternative no-code robotic programming approaches in the state of the art, the results with the low-cost hardware in the current setup (*see* 4.1) show great potential for no-code robotic programming. The analysis of the extracted orientation in the hand- and finger-gesture estimation will be addressed in the future work by comparing a single camera setup and multi-camera setup. This comparison will give a clear overview for the singularity issues in the extraction of finger orientation. The proposed system provides a modular and expandable system setup, utilizing low-cost hardware, in contrast to many state-of-the-art reference papers. Hence, the algorithms can be applied, extended and modified to fit different applications and scenarios by using different sensor technologies, robot systems and tools for example: the speech recognition system can be substituted by other low-cost input modalities (e.g: keyboard, button), the current low-cost cameras can be upgraded with high-end industrial cameras, the current robot system can be replaced by different cobots or traditional industrial robots, and linear axes can be integrated in the system.

In a robotic-applied industrial process, process parameters and requirements should be controlled to guarantee the quality of the end product. The robotic experts should not only be proficient in creating robotic programs, but they should also integrate the process parameter in the manufacturing process to meet the aimed quality of the end product. Even though robotic programming methods from state-of-the-art have simplified robotic programming for experts, The harmonization of the process parameter is still a big topic to research in the robotic research community. Most of the introduced approaches from the state-of-the-art are focusing only in developing a task oriented solutions for a specific application (e.g., Pick-and-Place and assembly). In contrast to them, the proposed method in this work offers a new perspective for a generic solution in intuitive robot programming by addressing modularity, agility and flexibility in the system setup. As a result, integration or replacement with different systems (e.g., sensors, robots) are possible. The modularity allows the programming approach to be combined with another algorithm (skill sets) to resolve an issue for robot program with specific applications. In Table 4, robotics-based industrial applications from different works in recent years are shown with their tolerance ranges. By comparing the result from the accuracy assessment of the novel approach with the given tolerances, it can be concluded that the proposed method has enormous potential to be implemented in various applications where medium tolerances in $10\ \mu m \leq x \leq 10\ mm$ and coarse tolerances in $x > 10\ mm$ are required. On the other side, the 6 *mm* accuracy of the proposed method would not satisfy the requirement for processes with fine tolerance in $x < 10\ \mu m$. Even though the current work was focused on the proposed method of teaching the robotic path based on hand-finger-gesture and voice. The vision and speech modality used in this work allows further development of intuitive robotic skill sets for the applied industrial processes in future works. These skill sets will allow the user to parameterize their process parameters and execute the process by applying process-specific control strategies as shown in Table 4. An example of a welding application will be explained in the following to depict the potential improvement of the system's inaccuracy by developing a welding skill set. The user would draw a welding path on the welding joint using his/her finger. The user triggers the skill set by saying "welding mode on." The finite state machine may trigger the activation of the vision-based control system to follow the weld, e.g. by using the methods mentioned in Table 4. This weld tracking algorithm will be used as a reference to control and compensate for the inaccuracy from the teaching phase. Another example represents an intuitive skill set for polishing that would allow automatic generation of process paths for basic geometries based on single user-defined points or features on the work piece *via* finger tracking. Trajectories with higher complexity may be taught to the robotic system by combining finger gestures and online impedance control of the robot manipulator. Specific parameters, e.g. amount of applied force for impedance control, may be figured by the user *via* voice commands. The combination of the multimodal programming method in this paper with intuitive skill sets will accelerate the deployment and reconfiguration of robotic systems in industrial context. In the future work, the implementation of intuitive skill sets for the proposed method will be addressed and assessed in an industrial use-case.

The camera-based vision system showed great potential for implementing the LfD strategy for robotic applications compared to other technology such as VR-, AR- or XR-based motion capture, used in state-of-the-art. However, the camera system still has its characteristic limitations in certain aspects. Various vision-based algorithms have pushed the vision system's limits and can compensate for many drawbacks of camera systems. In future works, an improvement in the methodology of the vision system can be addressed by applying recent algorithms from the state-of-the-art, such as:

- Positional and rotational accuracy improvement of the system → implementation of multi-camera system (Lippiello et al. (2005); Hoang (2020)), usage of camera with different technology (Langmann et al. (2012); Lourenço and Araujo (2021))
- Translation and rotation of the component after teaching → implementation of 6D object pose algorithm (Xiang et al. (2017); Sun et al. (2022))
- Component is bigger than field of view of the camera → usage of additional axes on a workpiece fixture or camera (translating the object with respect to the camera or *vice versa*) and implementation of image stitching or photogrammetry algorithm (Li et al. (2017); Ding et al. (2019))

In conclusion, this work contributes a novel approach to multimodal robotic programming by utilizing hand-finger-gesture recognition and speech recognition which can be implemented in different industrial applications and robotic systems. The proposed method is suitable for use without or with adequate experts in robotic programming. The *bona fide* evaluation results showed the system's potential to replace actual state-of-the-art methods. The opportunities for future developments of the system depict that the system can be a game changer in industrial robotic programming. This proposed programming method will accelerate the deployment of robotic systems in industrial use-case and affect how robotic systems are programmed in the industry for serial production or even batch size 1.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

The authors confirm contribution to the paper as follows: JH: Conceptualization, Methodology, Development, Experiments, Assessments and Writing. SK: Conceptualization, Supervision and Technical Supervision. PE: Conceptualization, Supervision, Reviewing, and Editing. MB: Conceptualization, Supervision, Reviewing, and Editing. SI: Supervision and Reviewing.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ajaykumar, G., Stiber, M., and Huang, C.-M. (2021). Designing user-centric programming aids for kinesthetic teaching of collaborative robots. *Robotics Aut. Syst.* 145, 103845. doi:10.1016/j.robot.2021.103845

Akkaladevi, S. C., Pichler, A., Plasch, M., Ikeda, M., and Hofmann, M. (2019). Skill-based programming of complex robotic assembly tasks for industrial application. *Elektrotech. Inftech.* 136, 326–333. doi:10.1007/s00502-019-00741-4

Akkaladevi, S. C., Plasch, M., Chitturi, N. C., Hofmann, M., and Pichler, A. (2020). Programming by interactive demonstration for a human robot collaborative assembly. *Procedia Manuf.* 51, 148–155. doi:10.1016/j.promfg.2020.10.022

Andreff, N., Horaud, R., and Espiau, B. (1999). "On-line hand-eye calibration," in Second International Conference on 3-D Digital Imaging and Modeling (No.PR00062), (Ottawa, Canada), 430–436. doi:10.1109/IM.1999.805374

Angelidis, A., and Vosniakos, G.-C. (2014). Prediction and compensation of relative position error along industrial robot end-effector paths. *Int. Suchy' J. Precis. Eng. Manuf.* 15, 63–73. doi:10.1007/s12541-013-0306-5

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics Aut. Syst.* 57, 469–483. doi:10.1016/j.robot.2008.10.024

Balogh, R., and Obdržálek, D. (2018). "Using finite state machines in introductory robotics," in International Conference on Robotics and Education RiE 2017 (Sofia, Bulgaria: Springer), 85–91.

Bdiwi, M., Hou, S., Winkler, L., and Ihlenfeldt, S. (2021). "Empirical study for measuring the mental states of humans during the interaction with heavy-duty industrial robots," in 2021 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA) (Talinn, Estonia: IEEE), 150–155.

Bdiwi, M., Kolker, A., Suchý, J., and Winkler, A. (2013a). "Automated assistance robot system for transferring model-free objects from/to human hand using vision/force control," in Social Robotics (Cham: Springer International Publishing), 40–53.

Bdiwi, M., Kolker, A., Suchý, J., and Winkler, A. (2013b). "Segmentation of model-free objects carried by human hand: Intended for human-robot interaction applications," in 2013 16th International Conference on Advanced Robotics (ICAR) (Montevideo, Uruguay: IEEE), 1–6.

Bdiwi, M., SuchỲ, J., Jockesch, M., and Winkler, A. (2015). "Improved peg-in-hole (5-pin plug) task: Intended for charging electric vehicles by robot system automatically," in 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15) (Mahdia, TN: IEEE), 1–5.

Bdiwi, M., SuchỲ, J., and Winkler, A. (2013c). "Handing-over model-free objects to human hand with the help of vision/force robot control," in 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13) (Hammamet, TN: IEEE), 1–6.

Beck, J., Neb, A., and Barbu, K. (2021). Towards a cad-based automated robot offline-programming approach for disassembly. *Procedia CIRP* 104, 1280–1285. doi:10.1016/j.procir.2021.11.215

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). "Robot programming by demonstration," in *Springer handbook of robotics*.

Blankemeyer, S., Wiemann, R., Posniak, L., Pregizer, C., and Raatz, A. (2018). Intuitive robot programming using augmented reality. *Procedia CIRP* 76, 155–160. doi:10.1016/j.procir.2018.02.028

Bolano, G., Roennau, A., Dillmann, R., and Groz, A. (2020). "Virtual reality for offline programming of robotic applications with online teaching methods," in 2020 17th International Conference on UbiquitousRobots (UR) (Kyoto, Japan: IEEE), 625–630.

Cho, J.-M., Park, S.-Y., and Chien, S.-I. (2020). Hole-filling of realsense depth images using a color edge map. *IEEE Access* 8, 53901–53914. doi:10.1109/ACCESS.2020.2981378

Chryssolouris, G., Georgoulias, K., and Michalos, G. (2012). "Production systems flexibility: Theory and practice," in IFAC Proceedings 14th IFAC Symposium on Information Control Problems in Manufacturing, 15–21. doi:10.3182/20120523-3-RO-2023.00442

Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. *Int. J. Robotics Res.* 18, 286–298. doi:10.1177/02783649922066213

Dietz, T., Schneider, U., Barho, M., Oberer-Treitz, S., Drust, M., Hollmann, R., et al. (2012). "Programming system for efficient use of industrial robots for deburring in sme environments," in ROBOTIK 2012; 7th German Conference on Robotics.

DIN EN ISO 10218-1 (2012). DIN EN ISO 10218-1: Robotik sicherheitsanforderungen - teil 1: Industrieroboter *norm*. Geneva, CH: International Organization for Standardization.

DIN EN ISO 10218-2 (2021). *Industrieroboter - sicherheitsanforderungen - Teil 2: Robotersysteme und Integration*. Geneva, CH: Norm, International Organization for Standardization.

DIN ISO/TS 15066 (2017). *Din ISO/TS 15066: Roboter und Robotikgeräte - kollaborierende Roboter. Norm*. Geneva, CH: International Organization for Standardization.

Ding, C., Liu, H., and Li, H. (2019). Stitching of depth and color images from multiple rgb-d sensors for extended field of view. *Int. J. Adv. Robotic Syst.* 16, 172988141985166. doi:10.1177/1729881419851665

Du, G., and Zhang, P. (2014). A markerless human–robot interface using particle filter and kalman filter for dual robots. *IEEE Trans. Ind. Electron.* 62, 2257–2264. doi:10.1109/tie.2014.2362095

El Zaatari, S., Marei, M., Li, W., and Usman, Z. (2019). Cobot programming for collaborative industrial tasks: An overview. *Robotics Aut. Syst.* 116, 162–180. doi:10.1016/j.robot.2019.03.003

Elliott, S., Xu, Z., and Cakmak, M. (2017). "Learning generalizable surface cleaning actions from demonstration," in 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN (IEEE), 993.

Funes-Lora, M., Vega-Alvarado, E., Rivera-Blas, R., Calva-Yáñez, M., and Sepúlveda-Cervantes, G. (2021). Novel surface optimization for trajectory reconstruction in industrial robot tasks. *Int. J. Adv. Robotic Syst.*, 18. Publisher Copyright: © The Author(s). doi:10.1177/17298814211064767

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* 47, 2280–2292. doi:10.1016/j.patcog.2014.01.005

Gastal, E. S. L., and Oliveira, M. M. (2011). Domain transform for edge-aware image and video processing. *ACM Trans. Graph.* 30, 1–12. doi:10.1145/2010324.1964964

Hägele, M., Nilsson, K., Pires, J. N., and Bischoff, R. (2016). *Industrial robotics*. Cham: Springer International Publishing. –1422. doi:10.1007/978-3-319-32552-1_54

Haugaard, R. L., Langaa, J., Sloth, C., and Buch, A. G. (2020). *Fast robust peg-in-hole insertion with continuous visual servoing. arXiv preprint arXiv:2011.06399*.

Hoang, V. T. (2020). Hgm-4: A new multi-cameras dataset for hand gesture recognition. *Data Brief* 30, 105676. doi:10.1016/j.dib.2020.105676

Horaud, R., and Dornaika, F. (1995). Hand-eye calibration. *Int. J. Rob. Res.* 14, 195–210. doi:10.1177/027836499501400301

Hornung, L., and Wurll, C. (2022). "Human-robot collaboration: a survey on the state of the art focusing on risk assessment," in Proceedings of Robotix-Academy Conference for Industrial Robotics (RACIR) 2021, Duren, Germany: Shaker Verlag, 10–17.

Huang, W., Ren, P., Wang, J., Qi, Q., and Sun, H. (2020). Awr: Adaptive weighting regression for 3d hand pose estimation." in AAAI Conference on Artificial Intelligence (AAAI). IEEE.

Infante, M. L., and Kyrki, V. (2011). "Usability of force-based controllers in physical human-robot interaction," in 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Lausanne, Switzerland: IEEE), 355–362.

Intel (2015). *Intel RealSense D400 series product family*, 5. Intel. Rev.

Ionescu, T. B., and Schlund, S. (2021). Programming cobots by voice: A human-centered, web-based approach. *Procedia CIRP* 97, 123–129. doi:10.1016/j.procir.2020.05.213

Iturrate, I., Kramberger, A., and Sloth, C. (2021). Quick setup of force-controlled industrial gluing tasks using learning from demonstration. *Front. Robot. AI* 8, 767878. doi:10.3389/frobt.2021.767878

Jokesch, M., Bdiwi, M., and SuchỲ, J. (2014). Integration of vision/force robot control for transporting different shaped/colored objects from moving circular conveyor. In 2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings (IEEE) (Timisoara, Romania: IEEE), 78–82.

Kakinuma, Y., Ogawa, S., and Koto, K. (2022). Robot polishing control with an active end effector based on macro-micro mechanism and the extended preston's law. *CIRP Ann.* 71, 341–344. doi:10.1016/j.cirp.2022.04.074

Kalaitzakis, M., Carroll, S., Ambrosi, A., Whitehead, C., and Vitzilaios, N. (2020). "Experimental comparison of fiducial markers for pose estimation," in 2020 International Conference on Unmanned Aircraft Systems (ICUAS) (Athens, Greece: IEEE), 781–789.

Kana, S., Tee, K.-P., and Campolo, D. (2021). Human–robot co-manipulation during surface tooling: A general framework based on impedance control, haptic rendering and discrete geometry. *Robotics Computer-Integrated Manuf.* 67, 102033. doi:10.1016/j.rcim.2020.102033

Karpagavalli, S., and Chandra, E. (2016). A review on automatic speech recognition architecture and approaches. *Int. J. Signal Process. Image Process. Pattern Recognit.* 9, 393–404. doi:10.14257/ijsip.2016.9.4.34

Lambrecht, J., Kleinsorge, M., Rosenstrauch, M., and Krüger, J. (2013). Spatial programming for industrial robots through task demonstration. *Int. J. Adv. Robotic Syst.* 10, 254. doi:10.5772/55640

Langmann, B., Hartmann, K., and Loffeld, O. (2012). Depth camera technology comparison and performance evaluation. *ICPRAM* (2), 438–444.

Lee, J. (2017). *A survey of robot learning from demonstrations for human-robot collaboration*. doi:10.48550/ARXIV.1710.08789

Lei, T., Huang, Y., Wang, H., and Rong, Y. (2021). Automatic weld seam tracking of tube-to-tubesheet tig welding robot with multiple sensors. *J. Manuf. Process.* 63, 60–69. doi:10.1016/j.jmapro.2020.03.047

Li, H., Liu, H., Cao, N., Peng, Y., Xie, S., Luo, J., et al. (2017). Real-time rgb-d image stitching using multiple kinects for improved field of view. *Int. J. Adv. Robotic Syst.* 14, 172988141769556. doi:10.1177/1729881417695560

Lippiello, V., Siciliano, B., and Villani, L. (2005). "Eye-in-hand/eye-to-hand multi-camera visual servoing," in Proceedings of the 44th IEEE Conference on Decision and Control (Seville, Spain: IEEE), 5354–5359.

Liu, S., Wang, L., and Wang, X. V. (2021). Sensorless haptic control for human-robot collaborative assembly. *CIRP J. Manuf. Sci. Technol.* 32, 132–144. doi:10.1016/j.cirpj.2020.11.015

Liu, S., Wang, L., and Wang, X. V. (2020). Symbiotic human-robot collaboration: Multimodal control using function blocks. *Procedia CIRP* 93, 1188–1193. doi:10.1016/j.procir.2020.03.022

Lourenço, F., and Araujo, H. (2021). "Intel realsense sr305, d415 and l515: Experimental evaluation and comparison of depth estimation," in Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP–2021) (Online), 362 –369.

Marchand, E., Uchiyama, H., and Spindler, F. (2016). Pose estimation for augmented reality: A hands-on survey. *IEEE Trans. Vis. Comput. Graph.* 22, 2633–2651. doi:10.1109/TVCG.2015.2513408

Massa, D., Callegari, M., and Cristalli, C. (2015). Manual guidance for industrial robot programming. *Industrial Robot An Int. J.* 42, 457–465. doi:10.1108/ir-11-2014-0413

MMPose-Contributors (2020). *OpenMMLab pose estimation toolbox and benchmark*.

Neto, P., and Mendes, N. (2013). Direct off-line robot programming via a common cad package. *Robotics Aut. Syst.* 61, 896–910. doi:10.1016/j.robot.2013.02.005

Park, F., and Martin, B. (1994). Robot sensor calibration: Solving ax=xb on the Euclidean group. *IEEE Trans. Rob. Autom.* 10, 717–721. doi:10.1109/70.326576

Peng, Y.-C., Carabis, D. S., and Wen, J. T. (2018). Collaborative manipulation with multiple dual-arm robots under human guidance. *Int. J. Intell. Robot. Appl.* 2, 252–266. doi:10.1007/s41315-018-0053-y

Qiao, Y. J., Liu, Z. H., Hu, D. X., and Xu, J. W. (2013). “Camera calibration method based on opencv,” in *Materials engineering and automatic control II (trans tech publications ltd)* (Shandong, China: Trans Tech Publications Ltd), 330, 517–520. doi:10.4028/www.scientific.net/AMM.330.517

Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. (2020). Recent advances in robot learning from demonstration. *Annu. Rev. Control Robot. Auton. Syst.* 3, 297–330. doi:10.1146/annurev-control-100819-063206

Robots, U. (2015). *User Manual UR10/CB3 - original instructions (en)*, 17782. Universal Robots. Rev.

Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., and Landay, J. (2016). *Speech is 3x faster than typing for English and Mandarin text entry on mobile devices. arXiv preprint arXiv:1608.07323*.

Sanchez-Diaz, A., Zaldivar-Colado, U., Pamanes-Garcia, J. A., and Zaldivar-Colado, X. (2019). Operation of a haptic interface for offline programming of welding robots by applying a spring-damper model. *Int. J. Comput. Integr. Manuf.* 32, 1098–1116. doi:10.1080/0951192x.2019.1686177

Servi, M., Mussi, E., Profili, A., Furferi, R., Volpe, Y., Governi, L., et al. (2021). Metrological characterization and comparison of d415, d455, l515 realsense devices in the close range. *Sensors* 21, 7770. doi:10.3390/s21227770

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). “Hand keypoint detection in single images using multiview bootstrapping,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Hawaii, United States: CVF/IEEE.

Soares, I., Petry, M., and Moreira, A. P. (2021). Programming robots by demonstration using augmented reality. *Sensors* 21, 5976. doi:10.3390/s21175976

Standard, O. (2014). Mqtt version 3.1. 1. *URL* http://docs.oasis-open.org/mqtt/mqtt/v31.

Strazdas, D., Hintz, J., Khalifa, A., Abdelrahman, A. A., Hempel, T., and Al-Hamadi, A. (2022). Robot system assistant (rosa): Towards intuitive multi-modal and multi-device human-robot interaction. *Sensors* 22, 923. doi:10.3390/s22030923

Sun, J., Wang, Z., Zhang, S., He, X., Zhao, H., Zhang, G., et al. (2022). “Onepose: One-shot object pose estimation without cad models,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (New Orleans, United States: CVF/IEEE), 6825–6834.

Tadic, V., Odry, A., Burkus, E., Kecskes, I., Kiraly, Z., Klincsik, M., et al. (2021). Painting path planning for a painting robot with a realsense depth sensor. *Appl. Sci.* 11, 1467. doi:10.3390/app11041467

Tian, F., Li, Z., Lv, C., and Liu, G. (2016). Polishing pressure investigations of robot automatic polishing on curved surfaces. *Int. J. Adv. Manuf. Technol.* 87, 639–646. doi:10.1007/s00170-016-8527-2

Tirmizi, A., De Cat, B., Janssen, K., Pane, Y., Leconte, P., and Witters, M. (2019). “User-friendly programming of flexible assembly applications with collaborative robots,” in 2019 20th International Conference on Research and Education in Mechatronics (REM) (Wels, Australia: IEEE), 1–7.

Tsai, R., and Lenz, R. (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. Rob. Autom.* 5, 345–358. doi:10.1109/70.34770

Tykal, M., Montebelli, A., and Kyrki, V. (2016). “Incrementally assisted kinesthetic teaching for programming by demonstration,” in 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (Christchurch, New Zealand: IEEE), 205–212.

van Delden, S., Umrysh, M., Rosario, C., and Hess, G. (2012). Pick-and-place application development using voice and visual commands. *Industrial Robot An Int. J.* 39, 592–600. doi:10.1108/01439911211268796

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 55, 248–266. doi:10.1016/j.mechatronics.2018.02.009

Voice INTER connect GmbH (2022). *vicCONTROL industrial version 6.3.0 - User guide - phytec Voice Control Kits (phyBOARD®-Mira). voice INTER connect GmbH*, 1. Rev.

Wrede, S., Emmerich, C., Grünberg, R., Nordmann, A., Swadzba, A., and Steil, J. (2013). A user study on kinesthetic teaching of redundant robots in task and configuration space. *J. Hum. Robot. Interact.* 2, 56–81. doi:10.5898/jhri.2.1.wrede

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2017). *Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199*.

Yang, C., Luo, J., Liu, C., Li, M., and Dai, S.-L. (2018a). Haptics electromyography perception and learning enhanced intelligence for teleoperated robot. *IEEE Trans. Autom. Sci. Eng.* 16, 1512–1521. doi:10.1109/tase.2018.2874454

Yang, L., Li, E., Long, T., Fan, J., and Liang, Z. (2018b). A novel 3-d path extraction method for arc welding robot based on stereo structured light sensor. *IEEE Sens. J.* 19, 763–773. doi:10.1109/jsen.2018.2877976

Zakiev, A., Tsoy, T., Shabalina, K., Magid, E., and Saha, S. K. (2020). “Virtual experiments on aruco and apriltag systems comparison for fiducial marker rotation resistance under noisy sensory data,” in 2020 International Joint Conference on Neural Networks (IJCNN) (Glassgow, United Kingdom: IEEE), 1–6.

Zhang, B., Wu, J., Wang, L., and Yu, Z. (2020a). Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot. *Robotics Computer-Integrated Manuf.* 63, 101923. doi:10.1016/j.rcim.2019.101923

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., et al. (2020b). *Mediapipe hands: On-device real-time hand tracking*. doi:10.48550/ARXIV.2006.10214

Zhang, H.-D., Liu, S.-B., Lei, Q.-J., He, Y., Yang, Y., and Bai, Y. (2020c). Robot programming by demonstration: A novel system for robot trajectory programming based on robot operating system. *Adv. Manuf.* 8, 216–229. doi:10.1007/s40436-020-00303-4

Zhang, S., Wang, S., Jing, F., and Tan, M. (2019). A sensorless hand guiding scheme based on model identification and control for industrial robot. *IEEE Trans. Ind. Inf.* 15, 5204–5213. doi:10.1109/tii.2019.2900119

Zhou, H., Ma, S., Wang, G., Deng, Y., and Liu, Z. (2021). A hybrid control strategy for grinding and polishing robot based on adaptive impedance control. *Adv. Mech. Eng.* 13, 168781402110040. doi:10.1177/16878140211004034

Zirkelbach, C., Krause, A., and Hasselbring, W. (2019). *Modularization of research software for collaborative open source development. arXiv preprint arXiv:1907.05663*.

TYPE Perspective
PUBLISHED 25 October 2022
DOI 10.3389/frobt.2022.1027173

Technological robot—Machine tool collaboration for agile production

Markus Wabner*, Hendrik Rentzsch, Steffen Ihlenfeldt and Andreas Otto

Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany

Check for updates

OPEN ACCESS

EDITED BY
Shaoming He,
Beijing Institute of Technology, China

REVIEWED BY
Wenzeng Zhang,
Tsinghua University, China

*CORRESPONDENCE
Markus Wabner,
markus.wabner@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 24 August 2022
ACCEPTED 10 October 2022
PUBLISHED 25 October 2022

CITATION
Wabner M, Rentzsch H, Ihlenfeldt S and Otto A (2022), Technological robot—Machine tool collaboration for agile production.
Front. Robot. AI 9:1027173.
doi: 10.3389/frobt.2022.1027173

The flexibility and efficiency in parts production can be significantly increased through the technological cooperation of industrial robots and machine tools. The paper presents an approach in which a robot, in addition to the classic handling tasks, enhance machine tools by additional manufacturing technologies and thus beneficially supports workpiece machining. This can take place in various configurations, starting with pre- and final machining by the robot outside the machine, through sequential cooperative machining of the workpiece clamped in the machine, to parallel, synchronized machining of a workpiece in the machine. The approach results in a novel type of collaborative manufacturing equipment for matrix production that will improve the versatility, efficiency and profitability in production.

KEYWORDS

robot machining, collaboration, agile manufacturing, machining, matrix production

1 Introduction

The increasing customization of customer requirements, ever shorter product development and product life cycles, but also global crises confront manufacturing companies with the challenge of making their production more agile and more resilient to disruptive changes. Above all, agility and resilience in production means a high degree of flexibility and adaptability at all levels. The vision of agile manufacturing is not entirely new: “Agile manufacturing systems can be conceptually thought of as being an integrated whole of complex interacting sub-systems, organized in such a way as to endeavor towards a common set of goals” (Merchant, 1984). Even today it is a concept of economic success in production (Gunasekaran et al., 2019). Forced and enabled by the rapid progress in ICT and artificial intelligence, numerous novel solutions become possible to further enhance agility in manufacturing and production. Even if agile production is primarily a planning and control task, the functionality of the available manufacturing equipment is of crucial importance, since it must provide the necessary technological capabilities (skill level) in a highly flexible manner. Industrial robots are significantly contribute to automation in production and the enormous number of industrial robots result in low prices. This was a reason to increasingly use industrial robots for machining. Of course the functionality of standard industrial robots is limited in comparison to machine tools, especially regarding positioning and path accuracy, static and dynamic stiffness und thus process stability and dynamic path accuracy. With various ICT support and the use of NC control systems, industrial robots become more and more suitable for machining and increasingly substituting selected machining tasks.

Manufacturing cells with robot automation in particular offer extensive but still largely unused potential for improving agility and resilience in production up to novel equipment strategies for matrix production. This unused potential is seen in the flexible enhancement of machine tools by additional manufacturing technologies that are provided by combined handling and machining robots. While industrial robots are used today to automate machine tools (parts handling: loading and unloading) and remain in a waiting position during machining operations, in future the robots will also take on technological tasks instead of waiting. The aim is not only to use the unproductive phases of the robots in the future and thus increase the productivity of the entire system. It is much more promising and therefore more important to flexibly expand machine tools and machining cells with additional manufacturing technologies and thus significantly increase the overall flexibility and adaptability in matrix production.

2 Robot machining and collaboration

Industrial robots for robot machining are available from various companies. The series of HSM robots (HSM = High Speed Machining) of Stäubli is suited for precise high-speed metal machining such as deburring, polishing, drilling, thread cutting, prototyping or the reworking of weld seams of various materials (aluminum, stainless steel, composite, *etc.*). Accuracy is reached by absolute calibration, model-based error compensation, a self-developed NC control CS9 and special design of core components. Mabi Robotic is using direct encoders in the joints, which increases controllability and accuracy significantly and thus allows rough machining and finishing in milling, turning and other cutting technologies. The Mabi robots are using CNC SINUMERIK 840D sl from Siemens, and Siemens CNC control is applied increasingly to other robots for machining, e.g., Kuka, Comau or even Stäubli. The last mentioned systems are configured and delivered by various companies (e.g., Robot Machining, ibs automation, ARRTSM, Boll Automation, FerRobotics or Fill) to e.g., OEMs or die making industry. Special CAM solutions for robot machining are available (e.g., Tebis, robotized, moduleworks). Various research activities are dealing with the error compensation of industrial robots, e.g., with additional piezo actuators (Schneider, 2013) or model-based *via* control system (Sörnmo et al., 2012; Fu et al., 2020). A high precise industrial robot was developed in the Flexmatic project (Flexmatic, 2016), combining a number of approaches (very stiff design, sensor integration, various error compensations). A good overview about robot machining is also given by Ji (Ji et al., 2019).

Robot-machine cooperation is well-known from automation (handling robots). Technological collaboration, where industrial robots and machines simultaneously machine one part, are actually not known. Mitsubishi Electrics presented a NC control system which allows such a kind of robot-machine-collaboration, but also here a practical application is not known. Wieland Anlagentechnik GmbH (Wieland, 2017) has presented a simplified robot-machine-collaboration solution, where the fixture of a handling robot couples a part physically to guiding systems of a machine tool to increase accuracy and stiffness. Collaborative systems are known from robot-robot or machine-machine collaboration. For machining of weak rigid large thin-walled aerospace parts, so-called mirror milling systems are replacing traditional processing methods. A dual-robot mirror milling system consisting of a machining hybrid robot and a supporting hybrid robot is presented in (Xiao et al., 2019). The cutter and the flexible supporting head are installed at the end of the machining robot and the supporting robot. The wall thickness error is measured by ultrasonic and compensated by the machining robot for accurately controlling the machining thickness. A similar system for machine tools is described in (Zhang et al., 2019). Dual robot setup that is widely used in assembly and handling applications, is used by (Owen et al., 2004) for machining. Owen proposed a dual robot setup, with one robot handling the material and the second one bearing the tool. Due to the redundant degree-of-freedom, the authors designed an off-line programming system with an integrated algorithm to optimize the trajectories of the tool, using the pseudo-inverse method. The approach monitors torque in the robot axes while also finds the optimum configuration/poses to improve the accuracy of the final part by decreasing tool deflection and optimum absorption of machining forces. In (Lin et al., 2009) such a system is described for surface polishing.

In (Huang and Lin, 2003) a dual independent robot machining cell is described, where the programming development was carried out by using CAM software to generate cutter location data for 5-axis milling together with a post processor to translate the CL data to linear and rotational motions for the robot cell controller. The implementation of the dual robot setup was achieved by dividing the original CL data in two parts taking into account collision detection between the two robots and minimization of force generated inaccuracy of the final geometry. The author also developed an offline programming module, enabling off-line programming and simulation of the dual robot machining cell.

Optimal division and allocation of the work and performing path planning in a coordinated manner while considering the requirements and constraints of collaborative industrial robots system is addressed in (Hassan et al., 2019) for fiber placement tasks. A two-stage approach is proposed in this paper. The first stage considers multiple objectives to optimally allocate each

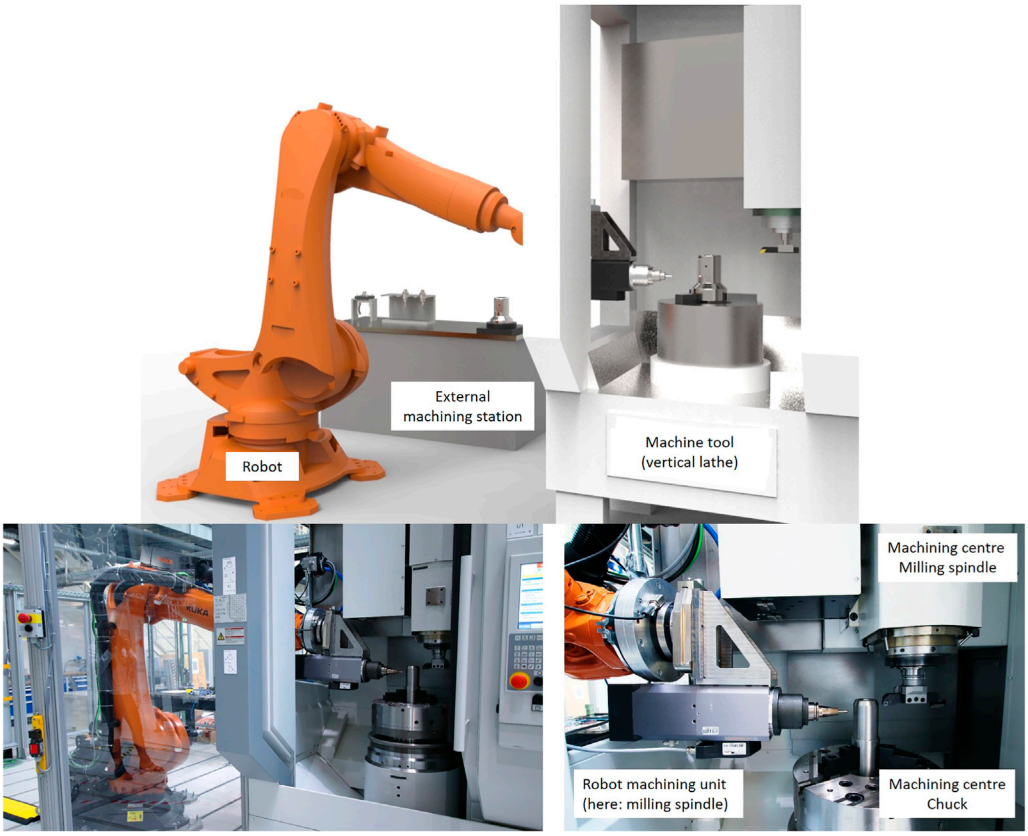

External machining station
Robot
Machine tool (vertical lathe)
Machining centre Milling spindle
Robot machining unit (here: milling spindle)
Machining centre Chuck

FIGURE 1
Principle of technological robot-machine-collaboration and authors test equipment.

industrial robots with surface areas, while the second stage aims to generate coordinated paths for the industrial robots.

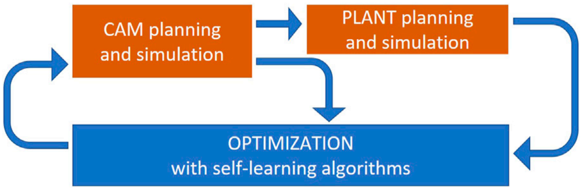

CAM planning and simulation
PLANT planning and simulation
OPTIMIZATION with self-learning algorithms

FIGURE 2
Optimization loops to find the best manufacturing scenarios.

3 Principle of the technological robot—machine tool collaboration

Actually, the technological collaboration of machines and robots has been implemented only rudimentarily in industrial environments. But the technological enhancement of the limited functionality of machine tools offers a high potential for increasing productivity and a new equipment basis for matrix production making production more flexible with limited resources. In addition, under certain circumstances the range of functions of a machine tools and thus the investment can be reduced. As a side effect, the utilization of the robots also increases. In this way, machines can be expanded with missing or similar NC robot axes, for example to add missing rotary axes to a 3-axis machine or to machine a workpiece with two tools at the same time. Another example is the enhancement of machines with non-existent technologies (e.g., enhancement of milling machines with force-controlled grinding). The principle of the technological robot-machine tool collaboration is shown in Figure 1, using the example of the test equipment used by the authors.

The technological enhancement of machine tools can take place in three different scenarios:

- Sequential pre-processing or finishing of the workpiece outside the machine tool (e.g. robot-based removal of casting flash on the raw part; robot-based additive

manufacturing; robot-based deburring or polishing on the finished part)
- Sequential machining of a workpiece within the machine tool (e.g. robot-based insertion of angled bores or execution of milling operations on a turned part clamped in a simple lathe)
- Parallel, synchronized machining of a workpiece by robot and machine tool (e.g. precision turning with machine, simultaneous robot-based deburring)

In addition to the technological scenarios mentioned as examples, the robots can of course continue to be used for conventional automation tasks such as handling or quality inspection. For this purpose, the robot must of course be equipped with the necessary technical systems (e.g. by changing systems).

4 Challenges and actual works

The realization of the robot-machine tool collaboration requires various research and development activities. It needs also novel strategies and fast CAM solutions to optimally break down a process chain to the involved systems. Such a CAM solution and the corresponding simulation tools are not available today. Furthermore, the strategy needs fast solutions for optimal (re-)configuration or (re-)allocation of the involved systems on the plant level as well as solutions for fast NC control coupling. To solve this, there is the need to describe the systems skills and to define skill levels of the systems. Together with technological and plant planning parameters, this will result in a very huge amount of thinkable combinations of machining and manufacturing scenarios. To handle the complexity and to find an optimal solution in an efficient way, AI-based optimization becomes a key technology. The main questions to be answered by optimization are.

- Which machining operation combination (or tool combination) would be the best?
- Which operation sequence would be the best?
- Which robot-machine combination should be selected?

And this under the consideration, that machine and robot are partly working together in parallel or in serial and that all machines and robots should preferable work at its maximum capacity. Thus, together with CAM simulation and PLANT simulation optimization circles (Figure 2) will be built to find the optimal parameter setup.

The inner optimization loop optimizes technological parameters, based on CAM simulation. The main optimization criterion is the machining time, which is also given to PLANT simulation. Here the various robot-machine combinations and its effects on the overall production of a plant will be simulated and optimised in the outer optimization loop. The used optimization algorithms are using artificial intelligence and are based on neural networks and genetic algorithms. The self-learning functionalities allow a significant reduction of needed optimization loops to find the optimum and its performance will increase by every new optimization task.

For the manufacturing optimization described, however, a number of other requirements must be fulfilled. One of them is the control system. It seems to make sense that each system retains its own NC control. However, the controller must be coupled together. This control coupling approach increases flexibility in production, both acutely and in the future by.

- Easy upgrade of existing machine tools and machining centres to a flexible production cell as basic equipment for matrix production
- Robots can be flexibly coupled to different machines, which increases the potential for versatility on the shop floor level
- The different systems can each be equipped with their optimal control

The control-related robot-machine coupling is implemented by coupling the two NC controllers using a Profinet (PN) connection (PN/PN coupling). The synchronization takes place both at the PLC level and at the NC level. Coupling at NC level is necessary to enable synchronized machining on a single workpiece. For this purpose, synchronization takes place at compile cycle level for synchronization in the position controller cycle.

An integrated CAM programming is the basis for the optimal splitting of the manufacturing tasks to the systems involved, for collision considerations and finally the generation of the G-codes. For this purpose, a CAM plugin for autodesk® products is being developed in order to take into account two independent but simultaneously working tool systems. The assignment of necessary machining operations is feature-based. The feature approach is necessary as an orientation for the optimizer, which should nevertheless be open to external (digitized) planning suggestions in order to also incorporate the experience of operators and planners. The optimizer's suggestion regarding the sequence, combination and parameterization of the machining tasks is then simulated in the CAM system. The result of the simulation is the digital output of key performance parameters (KPIs) for further optimization. The focus of the KPIs is on the processing time, which must be minimized. However, other criteria are also conceivable, such as energy requirements, tool wear or quality.

An essential prerequisite for the optimal distribution of tasks to the systems is a meaningful description of the technological capabilities (skill level) of the robot and machine tool systems. Ultimately, the question must be answered as to which processing task can be successfully carried out by which system? On the one hand, this requires the acquisition of system's basic parameters such as workspace, spindle parameters, general performance parameters (forces, moments, feed speeds) or accuracies, which are usually specified by the system provider. On the other hand, special technological parameters such as accuracies that can actually be achieved, knowledge from experience or dependencies (e.g. accuracy-material removal rate) must be quantified. This may require experimental analysis or special processing tests. The recording takes place *via* standardized skill level protocols (e.g. "Administration shell" VDE/DKE), which must be digitally processable. This is a basic protocol that can be manually expanded and can ultimately represent part of a digital twin of a digital process chain.

5 Summary and outlook

In the article, a new strategy for the technological cooperation of machine tools and robots was presented in order to increase flexibility and agility as well as productivity in matrix production. However, the realization of such systems represents a major challenge and requires a large number of new solutions and planning methods, which the authors are currently working on. This includes AI-based algorithms for the optimal distribution of tasks between the systems both at workpiece and shop floor level, the system's ability description (skill level), CAM-based tools for evaluating production scenarios and control solutions. The aim of the work is to implement a new type of flexible manufacturing cell that can be used both in classic parts manufacturing and as equipment for matrix production. If there are even several industrial robots and machine tools available, further optimization potential can be leveraged on the shop floor by the need-based and temporary combination of robots with corresponding machines.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Flexmatic. (2016). Flexmatic. Available at: www.flexmatik.de [Accessed March 05, 2022].

Fu, Z., Dai, J. S., Yang, K., Chen, X., and Lopez-Custodio, P. (2020). Analysis of unified error model and simulated parameters calibration for robotic machining based on Lie theory. *Robotics Computer-Integrated Manuf.* 61, 101855. doi:10.1016/j.rcim.2019.101855

Gunasekaran, A., Yusuf, Y. Y., Adeleye, E. O., Papadopoulos, T., Kovvuri, D., and Geyi, D. G. (2019). Agile manufacturing: An evolutionary review of practices. *Int. J. Prod. Res.* 57 (15-16), 5154–5174. doi:10.1080/00207543.2018.1530478

Hassan, M., Liu, D., and Xu, D. (2019). A two-stage approach to collaborative fiber placement through coordination of multiple autonomous industrial robots. *J. Intell. Robot. Syst.* 95, 915–933. doi:10.1007/s10846-018-0919-0

Huang, H.-k., and Lin, G. C. (2003). Rapid and flexible prototyping through a dual-robot workcell. *Robot. Comput. Integr. Manuf.* 19, 263–272. doi:10.1016/s0736-5845(03)00022-x

Ji, W., and Wang, L. (2019). Industrial robotic machining: A review. *Int. J. Adv. Manuf. Technol.* 103, 1239–1255. doi:10.1007/s00170-019-03403-z

Lin, F., Wang, X., and Lin, F. (2009). "Development of a dual-robot system for parametric surfaces polishing," in 2009 International Asia Conference on Informatics in Control, Automation and Robotics, Bangkok, Feb. 1 2009 to Feb. 2 2009, 161–165.

Merchant, M. E. (1984). Analysis of existing technological forecasts pertinent to utilization of artificial intelligence and pattern recognition techniques in manufacturing engineering, 16~ *CIRP Int. Sem. Manuf. Syst.*, 14:1, 16–32.

Owen, W. S., Croft, E. A., and Benhabib, B. (2004). "Real-time trajectory resolution for dual robot," in Proceedings of IEEE International Conference on Robotics & Automation ICRA, New Orleans, LA, USA, 26 April 2004 - 01 May 2004, 4332–4337.5

Schneider, U., Drust, M., Puzik, A., and Verl, A. (2013). Compensation of errors in robot machining with a parallel 3D-piezo compensation mechanism. *Procedia CIRP* 7, 305–310. doi:10.1016/j.procir.2013.05.052

Sörnmo, O., Olofsson, B., Robertsson, A., and Johansson, R. (2012). Increasing time-efficiency and accuracy of robotic machining processes using model-based adaptive force control. *IFAC Proc. Vol.* 45 (22), 543–548. doi:10.3182/20120905-3-hr-2030.00065

Wieland (2017) Industrieanzeiger.Industrie. Available at: https://industrieanzeiger.industrie.de/technik/fertigung/roboter-als-werkzeugmaschine-2/ [Accessed Marc 05, 2022].

Xiao, J., Zhao, S., Guo, H., Huang, T., and Lin, B. (2019). Research on the collaborative machining method for dual-robot mirror milling. *Int. J. Adv. Manuf. Technol.* 105, 4071–4084. doi:10.1007/s00170-018-2367-1

Zhang, S., Bi, Q., Ji, Y., and Wang, Y. (2019). Real-time thickness compensation in mirror milling based on modified Smith predictor and disturbance observer. *Int. J. Mach. Tools Manuf.* 144, 103427. doi:10.1016/j.ijmachtools.2019.103427

frontiers | Frontiers in Robotics and AI

TYPE Review
PUBLISHED 01 December 2022
DOI 10.3389/frobt.2022.1027340

OPEN ACCESS

EDITED BY
Kostas J. Kyriakopoulos,
National Technical University of Athens, Greece

REVIEWED BY
Andreas Otto,
Fraunhofer Institute for Machine Tools and Forming Technology (FHG), Germany
Helman Stern,
Ben-Gurion University of the Negev, Israel

*CORRESPONDENCE
Fouad Bahrpeyma,
bahrpeyma@ieee.org

SPECIALTY SECTION
This article was submitted to Robotic Control Systems, a section of the journal Frontiers in Robotics and AI

RECEIVED 24 August 2022
ACCEPTED 08 November 2022
PUBLISHED 01 December 2022

CITATION
Bahrpeyma F and Reichelt D (2022), A review of the applications of multi-agent reinforcement learning in smart factories.
Front. Robot. AI 9:1027340.
doi: 10.3389/frobt.2022.1027340

A review of the applications of multi-agent reinforcement learning in smart factories

Fouad Bahrpeyma* and Dirk Reichelt

Smart Production Systems, HTW Dresden, Dresden, Germany

The smart factory is at the heart of Industry 4.0 and is the new paradigm for establishing advanced manufacturing systems and realizing modern manufacturing objectives such as mass customization, automation, efficiency, and self-organization all at once. Such manufacturing systems, however, are characterized by dynamic and complex environments where a large number of decisions should be made for smart components such as production machines and the material handling system in a real-time and optimal manner. AI offers key intelligent control approaches in order to realize efficiency, agility, and automation all at once. One of the most challenging problems faced in this regard is uncertainty, meaning that due to the dynamic nature of the smart manufacturing environments, sudden seen or unseen events occur that should be handled in real-time. Due to the complexity and high-dimensionality of smart factories, it is not possible to predict all the possible events or prepare appropriate scenarios to respond. Reinforcement learning is an AI technique that provides the intelligent control processes needed to deal with such uncertainties. Due to the distributed nature of smart factories and the presence of multiple decision-making components, multi-agent reinforcement learning (MARL) should be incorporated instead of single-agent reinforcement learning (SARL), which, due to the complexities involved in the development process, has attracted less attention. In this research, we will review the literature on the applications of MARL to tasks within a smart factory and then demonstrate a mapping connecting smart factory attributes to the equivalent MARL features, based on which we suggest MARL to be one of the most effective approaches for implementing the control mechanism for smart factories.

KEYWORDS

Industry 4.0, multi-agent reinforcement learning, smart factory, smart manufacturing, smart production systems, reinforcement learning, Artificial Intelligence

1 Introduction

As the market becomes more dynamic and complex and the global demand for small-batch, short-life products increases, manufacturing systems are facing more dynamic and complex environments. A smart factory, as the heart of Industry 4.0, is envisioned as a fully automated flexible manufacturing system consisting of highly connected smart components to provide mass customization and short product life cycles in a cost-effective, agile, and self-organizing manner. To move toward the realization of the concept of "smart factory", manufacturing systems have been incrementally equipped with advanced enabling technologies such as (Industrial) Internet of things (I-IoT), cyber-physical systems, artificial intelligence (AI), cloud manufacturing systems, and big data technologies (Mittal et al., 2019). Taking these technologies into account as the backbones and building blocks of a smart factory, the key role, however, is played by the control mechanism that is responsible for making fine-grained and coarse-grained decisions to guarantee the performance of the manufacturing system. Simple examples of control mechanisms include the formation of robots, the design of paths between stations, the scheduling of operations, the management of inventories, the response to demands, and every decision that needs to be made to achieve specific goals (Jung et al., 2017). The control mechanism for smart factories, however, when compared with that of traditional manufacturing systems, is subject to various challenges. For the most part, high-level flexibility (required for smart factories) necessitates fine-grained decisions, thereby increasing the complexity of the control mechanism as more variables are introduced. This implies that the development of an optimal control mechanism will require the consideration of a larger number of contributing factors in the control equations, resulting in a longer engineering cycle and more complex equations. Traditional optimization approaches are often highly time-consuming and thus cannot be effectively incorporated to deal with the rapid and dynamic manufacturing environment in real-time. Furthermore, as manufacturing systems become larger in scale, the number of possible situations that need rapid and optimal responses will increase as well. As difficult as it is to predict all of the possible scenarios, what is even more challenging is determining the most effective reaction (real-time and efficient) to each case. Considering the complexity of this system and the limited accuracy and abilities of a human engineer, a detailed investigation of the potential situations and corresponding solutions is an extremely burdensome and expensive process. A smart factory offers a self-organizing solution to these situations through the incorporation of advanced intelligent control mechanisms, in which smart components make optimal decisions pseudo-independently in order to provide rapidity and can communicate with one another in order to act jointly toward global performance metrics such as efficiency, tardiness, and production rate.

According to the majority of the studies in the community such as Shi et al. (2020), Büchi et al. (2020), and Sjödin et al. (2018), in practice, realizing the concept of the smart factory in its full form is not possible without the use of advanced AI techniques. "Smart" in smart factories refers primarily to the application of artificial intelligence. AI provides powerful tools (for analytics and decision-making) for analyzing the vast amounts of data generated by smart components and for making optimal decisions. A variety of AI techniques have been used in the field to enhance automation, rapidity, and efficiency, such as neural networks (NNs) and their derivatives such as long short-term memory (LSTM) (Ozdemir and Koc, 2019; Nwakanma et al., 2021). These techniques were most commonly used to provide proactive decisions to pre-allocate optimal resources before the requests arrived. However, such approaches are limited when dealing with uncertainties such as machine breakdowns or the insertion of new jobs. A centralized control mechanism over multi-agent systems can also result in delays, and there is always a trade-off between global optimality and rapid response.

In this paper, *via* conducting a review study on the applications of MARL to decision-making problems within a smart factory, we suggest that one of the most effective solutions to realize self-organization in smart factories in an efficient and real-time manner can be found *via* multi-agent reinforcement learning (MARL), which offers a decentralized solution to dealing with uncertainty. MARL is an AI technique that incorporates reinforcement learning (RL) into multi-agent systems (MASs), where RL automatically finds optimal solutions to uncertainty through interactions between intelligent agents and the problem environment. More details on MARL are provided in Section 3. In order to demonstrate this, we will review the literature about MARL's application to smart factory tasks and then discuss a mapping from smart factory requirements to MARL capabilities.

Several literature reviews of MARL approaches have been conducted in recent years, including those by OroojlooyJadid and Hajinezhad (2019), Nguyen et al. (2020), and Zhang et al. (2021), which were also used as the basis for this study. However, to the extent of our knowledge, this is the first study reviewing the applications of MARL into the smart factory tasks.

The remainder of this paper is organized as follows. In Section 1, we will describe the methodology, objectives, and boundaries of this research. Section 2 presents an overview of the RL and MARL systems. In Section 3, the literature is surveyed for the application of MARL approaches to tasks within a smart factory. Section 4 presents our conceptual analysis on the match between MARL and smart factory attributes. Section 5 provides a discussion on the concerns, limitations, and potentials of the applications of MARL to smart factory tasks. Finally, Section 6 concludes this paper and states the gaps and potentials for future research.

1.1 Research objectives, boundaries, and methodology

This research seeks to showcase the capabilities that MARL brings to realizing smart factories. To this end, this paper is mainly devoted to conducting a review study on the applications

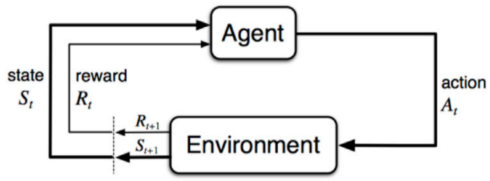


FIGURE 1
Agent in loop (Sutton and Barto (1998)).

of MARL to the tasks within a smart factory. We will then draw a mapping from the required characteristics of smart factories to the corresponding capabilities in MARL. This paper provides practitioners and researchers in the field with an overall idea of how to address control problems in smart factories *via* the use of MARL. The size limitations of this paper, however, prevent attention to every detail since the ways of formulating tasks within the smart factory into MARL problems are extremely diverse and thus are beyond the scope of this study. In view of the fact that MARL is relatively new in the field of smart factories, this paper also discusses situations where RL is applied in conjunction with multi-agent systems in the relevant field, while focusing primarily on MARL. We mainly focus on cooperative approaches. Consequently, for each work, we briefly summarize the factors that indicate how MARL is integrated with the corresponding application.

2 Overview of multi-agent reinforcement learning approaches

This section provides a brief background on MARL. In this paper, when we refer to MARL, without the loss of generality, we address the case where RL is applied to or implemented *via* a multi-agent system. Therefore, we will consider a broader area than only the scope of MARL systems in theory. In essence, RL can be viewed as the most general form of the learning problems. Contrary to supervised machine learning, the target for an RL algorithm is a feedback which is partial and almost a delayed reward (or penalty). Moreover, RL differs from unsupervised learning, since its primary objective is to maximize reward signals rather than discover hidden structures within unlabeled data. Based on Sutton and Barto (1998), the ultimate objective of an RL system is the maximization of the expected value of the cumulative sum of a reward (immediate reward) signal. RL is generally aimed at achieving a long-term goal; therefore, the reward is back-propagated and discounted by a discount factor, and the goal is to maximize the discounted cumulative future rewards at a discounted rate.

RL problems are usually formalized using Markov decision processes (MDPs). MDPs are the mathematical representations of RL problems that originate in dynamic systems. The Markov property implies that all of the relevant information for a decision is encoded in the state vector s_t. An MDP problem is described by the four-tuple M = [S; A; p; r], where S denotes the environment's state space and A denotes the agent's action space $a_t \in A$. In s_t, the agent performs a_t in order to move to s_{t+1} and is immediately rewarded with r_t. Additionally, p_{ss} represents the probability that a particular action performed at state s_t leads to the state s_{t+1} being reached. The uncertainty, however, is represented by a probability distribution function that indicates whether taking the action at the state s_t results in s_{t+1} for the agent. It emphasizes that the agent's state transitions are partially influenced by the agent's actions. Figure 1 illustrates the agent in an RL problem.

A more realistic version of MDP is the partially observable MDP (POMDP), wherein, as opposed to MDP, the agent only obtains a partial observation from the environment, leading to a higher level of uncertainty. For more details, please see Zhang et al. (2021).

The most popular RL techniques are Q-learning (QL), SARSA, and actor–critic (AC), each having variants as a means to enhance their performances in specific ways. Due to the limitations of this paper, here, we only briefly describe QL as the most popular RL technique and refer the reader to Arulkumaran et al., (2017) for other RL methods. In QL, the objective is to learn an action selection policy π_t $(a|s)$ toward maximizing the reward. The policy π_t $(a|s)$ represents the probability distribution that $a_t = a$ if $s_t = s$. The Q-function Q: $S \times A \rightarrow R$ yields the discounted cumulative reward with the discount factor γ.

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a'). \tag{1}$$

The QL algorithm is implemented to obtain the best action value function $Q^*(s,\ a)$, as shown in Eq. 2:

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[\sum_t r_t \gamma^t | s_t = s, a_t = a, \pi\right]. \tag{2}$$

Q-values are updated using Eq. 3, every time an action is taken.

$$Q_{t+1}(s,a) = Q_t(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t)\right). \tag{3}$$

With the emergence of deep neural networks (DNNs), the area of RL was able to address a wider range of applications due to generalization over past experiences for unseen situations, as opposed to traditional tabular RL, which was also vulnerable to the explosion of the state space due to the tabular formulation. DNNs are able to store values and approximate RL values for unseen situations. The well-known deep Q-network (DQN) and various types of AC methods rely upon DNNs. DNNs are one of the enabling technologies for MARL, since the extremely large state space of MARL problems cannot be contained in the tabular form.

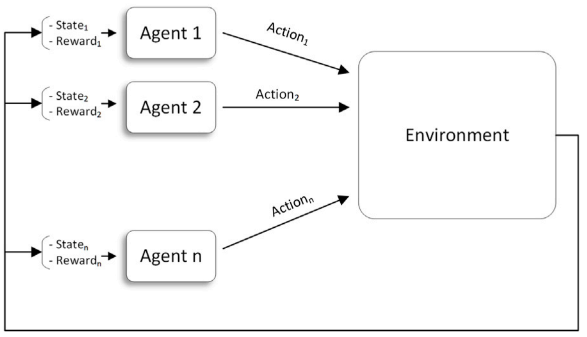


FIGURE 2
Agents in a multi-agent reinforcement learning (MARL) environment

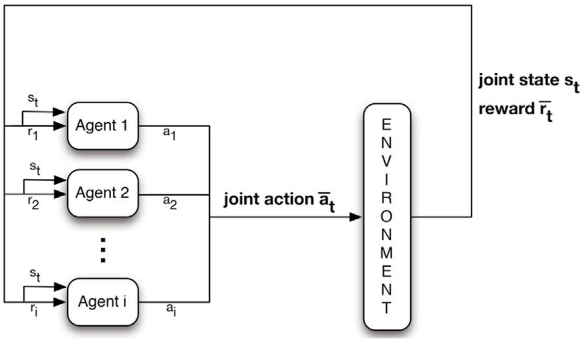


FIGURE 3
JAL (Nowé et al. (2012)).

In MARL, the actions taken by each agent change the environment and consequently change the perception of the other agents from the environment (the previous state before action *a* was taken is no longer valid, and the state has changed). This problem is known as non-stationarity.

MARL (Figure 2) incorporates multiple agents that interact with one another and with the environment. In accordance with the objectives of the application, this interaction may be cooperative, competitive, or mixed. This research focuses primarily on cooperative MARL, where the agents cooperate to maximize a common global reward while trying to maximize their own local rewards.

In the multi-agent setting, the generalization of MDP is the stochastic game. Based on Buşoniu et al. (2010), a stochastic game is described *via* a tuple $(X, U_1, \ldots, U_n, f, \rho_1, \ldots, \rho_n)$, where n denotes the number of agents; X is the environment state space; and U_i, $i = 1, \ldots, n$ represents the agents' action space, leading to a joint action set $\boldsymbol{U} = U_1 \times \cdots \times U_n$. The state transition probability function is defined as f: $X \times \boldsymbol{U} \times X \rightarrow [0, 1]$, and the rewards for the agents are ρ_i: $X \times \boldsymbol{U} \times X \rightarrow R$, $i = 1, \ldots, n$.

The state transitions in the multi-agent case result from the joint actions of the agents at step k, $\boldsymbol{u}_k = [u_{1,k}^T, \ldots, u_{n,k}^T]^T, \boldsymbol{u}_k \in \boldsymbol{U}, u_{i,k} \in U_i$.

The transpose of a vector is indicated by T. The joint policy h_i: $X \times U_i \rightarrow [0, 1]$ represents all the policies. The joint action leads to the reward $r_{i,k+1}$, and thus, the returns also depend on the joint policy.

$$R_i^h(x) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{i,k+1} | x_0 = x, \mathbf{h}\right\}. \tag{4}$$

MARL is mainly formalized *via* multi-agent (PO)MDP or decentralized (PO)MDP, which are referred to as M(PO)MDP and Dec-(PO)MDP, respectively. Depending on the application, the problem formulation may vary significantly, and thus, we refer the reader to Buşoniu et al. (2010) and Zhang et al. (2021) for more details and for the formal definition of MARL.

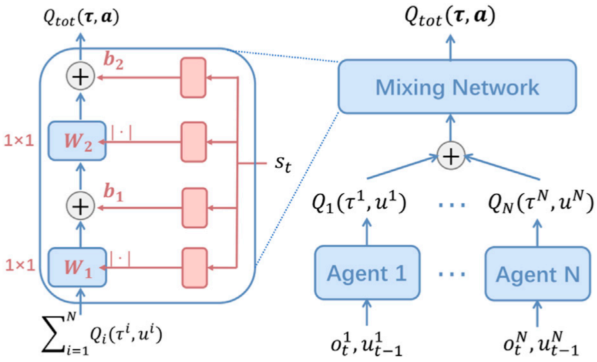


FIGURE 4
QMIX (Rashid et al. (2018)).

A variety of cooperative MARL approaches have been presented in the literature. The most famous methods are independent action learners (IALs), joint action learners (JALs), team-Q, distributed Q, distributed AC, communication-based, and network-based methods such as QMIX. Please see Zhang et al. (2021) for more details. We briefly describe some of the most popular MARL methods here.

IAL: Such as independent Q-learning (IQL), wherein agents take actions independently and only interact with each other through the environment. IAL methods incorporate RL individually and assume the other agents are part of the environment. These approaches, if not containing the impact of agents on each other, are subject to the problem of non-stationarity.

JAL (Figure 3): Agents take (and learn the value of) joint actions to avoid non-stationarity. The most concerning challenge is, however, the credit assignment problem (identifying the agents' individual share of the reward). In many cases, however, some agents become lazy in the team play because

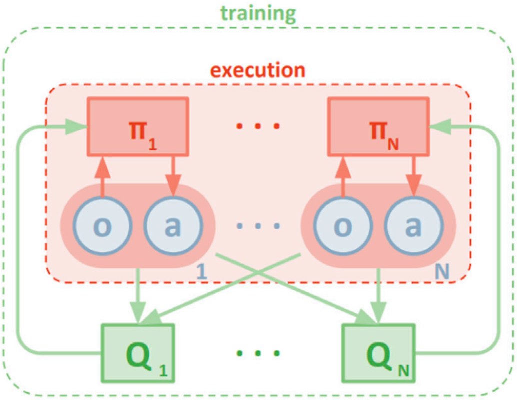


FIGURE 5
MA-DDPG (Lowe et al. (2017)).

they do not identify their contribution to the global reward. In addition, the exploration strategy is difficult to design for JAL approaches since the joint action space for JAL is much larger than that of IAL.

QMIX (Figure 4): QMIX serves as a representative value decomposition technique and works based on the principles of centralized training decentralized execution (CTDE). QMIX is primarily dependent on the mixing network to deal with the credit assignment problem. The mixing network receives the outputs of the individual agents' Q-networks $Q_a\,(T_a, u_a)$ to assign the individuals the credits and approximate the global Q-value Q_{tot} $(T,\ u,\ s;\ \theta)$. QMIX guarantees the individual-global maximum (IGM) principle, meaning that optimal local decisions jointly lead to the optimal joint decision for Q_{tot}.

$$\underset{u}{argmax}\ Q_{tot}\left(\tau, u, s\right) = \begin{pmatrix} \underset{u^1}{argmax}\ Q_1\left(\tau^1, u^1\right) \\ . \\ . \\ \underset{u^n}{argmax}\ Q_n\left(\tau^n, u^n\right) \end{pmatrix}. \qquad (5)$$

This enables the agents to take optimal local actions based on their local policies, while the joint action is also optimal for the entire system.

MADDPG (Figure 5): Multi-agent DDPG suggests the association of each agent with a separate pair of actors and critics, training the critics centrally, and only using the actors during execution. Actors are trained by local state–action data, while critics receive training through the global state–action context.

3 Applications

There are various tasks within a smart factory that can be formulated as MARL problems, among which scheduling and transportation are of the highest popularity, while the literature has also paid attention to maintenance, energy management, and human–robot collaboration. In this paper, the emphasis is primarily placed on scheduling and transportation due to the large number of existing publications; other applications, due to limited attention in the literature, are discussed selectively to complete the study.

3.1 Scheduling

In the relevant literature, the job-shop scheduling problem (JSSP) and its variants have been extensively incorporated as abstractions of the manufacturing environment and smart manufacturing systems. Due to the JSSP's NP-hard nature, only local optimal solutions can be obtained, which makes it difficult to address under multi-agent settings. Due to the complexity associated with developing such scheduling systems in dynamic environments such as smart factories, the use of self-adaptive and self-learning approaches in this regard has recently gained a great deal of attention from practitioners. MARL approaches offer varieties of advantages for developing such systems due to their effectiveness in dealing with uncertainties such as breakdowns and new job insertions.

We can categorize the related works concerning scheduling into centralized and decentralized classes.

3.1.1 Centralized approaches

Centralized approaches consider a central controller that manages collaborations between agents, and as a result, this might lead to delayed decisions and an increase in computational complexity.

Gabel and Riedmiller (2008) proposed a tabular multi-agent QL approach for addressing dynamic scheduling problems in which unexpected events may occur, such as the arrival of new tasks or the breakdown of equipment, which would require frequent re-planning. Machine agents perform scheduling decisions based on local observations. A simple joint action selection method is used with tabular QL to provide reactive scheduling policies for dynamic scheduling environments.

Wang and Yan (2016) proposed a tabular multi-agent QL approach with joint action learners for adaptive assembly scheduling in an aero-engine manufacturing system. The authors use clustering to reduce the size of state space and assign fuzzy memberships to observations to link them to the states.

Sultana et al. (2020) proposed a multi-agent A2C approach to supply and chain management for multi-inventory systems in smart factories. The main goal of this work is to minimize under-stocking and over-stocking, where both impose negative impacts on the whole system. The warehouse is responsible for supplying multiple stores, which face requests for different types of products. This case is also linked to the smart factory concept, where, due to the mass customization feature of smart factories, demands can be submitted to any sub-factories, and thus, the product parts are collected from the corresponding store (that is placed inside or near the intended sub-factory). This work considers two types of agents: the warehouse and store agents. They both contribute to the accumulative reward and are trained *via* an A2C algorithm. The state vectors for both the warehouse and store agents consist of the demand forecast to further enhance the performance. The process of training is initiated by training the store agents independently with the full availability of stocks in the warehouse and then by training the warehouse agent based on the model obtained for the store agents. At the final step, the data are used to further train all the agents together for performance improvement. However, this work is more on the side of hierarchical reinforcement learning (HRL), and concerns such as non-stationarity have not been discussed and studied.

Luo et al. (2021a) presented a hierarchical RL (HRL) approach with two hierarchies for production scheduling in order to minimize the total tardiness and the average machine utilization rate. The paper uses a high-level DDQN agent to determine the global optimization goal, and a low-level DDQN is responsible for selecting appropriate dispatching rules. However, the use of a high-level controller agent in HRL approaches increases the depth of the RL problem and thus reduces the chance of convergence of the learning process.

Luo et al. (2021b) developed a hierarchical multi-agent proximal policy optimization (HMAPPO) approach as a means of dealing with the dynamic multi-objective flexible job-shop scheduling problem, where some operations are subject to the no-wait constraint. The method incorporates three types of agents, including the objective agent (as the controller), the job agent, and the machine agent (as the local actuators). The object agent periodically specifies temporary optimization objectives, the job agent chooses the job selection rule, and the machine agent chooses the machine assignment rules for the corresponding temporary objectives. With HRL, this method conducts learning at different levels of abstraction, whereby the high-level controller (the objective agent) learns policies over high-level objectives at a slow pace, while the lower-level actuators (job and machine agents) learn policies over low-level actions that meet the real-time constraints and goals. This means that some jobs should be continuously processed without interruption. However, with HRL, the architecture is not fully decentralized, as there should be a high-level controller at the top. It is also important to note that HRL approaches cannot guarantee the optimality of the overall aggregate policy of multiple agents.

3.1.2 Decentralized approaches

As opposed to centralized approaches, decentralized methods are not characterized by a central control mechanism to manage agents toward their tasks. Decentralized approaches allow for agile decisions and a reduction in the overall computational complexities resulting from the elimination of the need for a central controller.

Qu et al. (2016) developed a two-agent Markov game approach based on QL to realize real-time cooperation between machines (scheduling) and the workforce (human resource management agents). This work aims to obtain an appropriate performance both for the scheduling agent and the human resource management agent for handling multi-process operations associated with different products in the dynamically changing environment of manufacturing systems.

Bouazza et al. (2017) presented a distributed QL approach for production scheduling that considers products as intelligent agents (which perform independently without a centralized control) and that intends to highlight the significant impact of considering the contribution of setup time (which is mainly neglected) in decision-making on the overall performance. Intelligent product agents can decompose decision-making into the choice of the machine selection rule and the selection of the dispatching rule. However, although agents each has impacts on the environment, this work fails to indicate the common impact of the decisions made by the agents toward the environment and toward each other.

Wang et al. (2017) developed a tabular multi-agent QL approach for dynamic flow-shop scheduling in manufacturing systems. Machine agents learn to make independent dispatching decisions simultaneously, regarding the expectations from other agents to deal with the changing environment. The proposed method showed superiority over first-in-first-out (FIFO), earliest due date (EDD), and shortest processing time (SPT) with respect to the mean flow time, mean lateness, and percentage of late jobs.

In Hong and Lee (2018), the authors used an asynchronous advantage actor–critic (A3C) to schedule some robotic arms for cluster cleaning in the semiconductor factory. The learning occurs in a two-stage process. In the first stage, the learning is carried out for the robot agent's action selection policy π_{arm} $(a_R| s_t = s; \theta_{arm})$, and in the second stage, the policy $\pi_{i,j}$ $(a_{i,j}|s_t = s, a_{arm} = a_R; \theta_{i,j})$ is learned for deciding the start time of the cleaning operation in the j^{th} chamber at the i^{th} processing step. In order to perform MARL without dealing with non-stationarity, they used A3C, which enables learning in multiple parallel environments in an asynchronous manner.

In Waschneck et al. (2018a) (and also the extended experiments in Waschneck et al. (2018b)), the authors presented a multi-agent DQN approach for production scheduling in the semiconductor industry in an attempt to address re-entrant production flows and sequence-dependent setups. This work assigns agents to production stages and assigns jobs to a single machine in each stage. In an attempt to enhance stability, while all agents are trained independently, they use the DQNs of the other agents for the remainder of the work centers. While the environment is controlled under the influence of all the DQN agents associated with the agents, in order to deal with non-stationarity, only one agent trains its DQN actively at a time.

The active agent considers the actions of the other agents during its training process. Due to the fact that all the agents strive toward maximizing a single global reward, the entire process can be described as cooperative reinforcement learning. The training process, in this work, comprises two phases. During the first phase, only one DQN agent is trained (separately repeated for each agent), while the rest are controlled using heuristics. The second phase consists of all work centers being controlled by DQN agents, while agents are trained in turn for a limited amount of time. The experiments indicate that the proposed method outperforms some dispatching heuristics. This work, however, appears to be incapable of handling changes in production requirements and the number of machines since the network must be retrained each time such changes are made.

Wang et al. (2018) proposed the concept of shared cognition *via* the use of a tabular multiple-agent QL to deal with disturbances in manufacturing cells. When disturbances occur, the corresponding agent distributes the information, and the agents share their cognition to raise the occurrence of a disturbance (a solution for that). Agents each incorporate a tabular QL model to learn the appropriate dynamic scheduling strategy without causing conflicts. The use of shared cognition provides the manufacturing cells with the ability to communicate and to distribute disturbed jobs to the appropriate cells in order to avoid conflicting with the quality of the service.

Motivated by the approach presented by Waschneck et al. (2018a), another multi-agent DQN-based approach was presented in Park et al. (2019) for production scheduling in semiconductor manufacturing systems. Regarding Waschneck et al. (2018a), the main objective in this work was to resolve issues with training the DQN agents in dealing with the scheduling problem under variable production requirements, such as a variable number of machines and a variable initial setup status. This work incorporates a shared DQN (presented in Foerster et al. (2016)) to improve the scheduling performance when dealing with variable production settings, especially the number of machines and the initial setup status. This approach, similar to Waschneck et al. (2018a), has two main phases. In the training phase, the production scheduling problem is practiced in an episodic manner *via* simulation. A double QL setting is incorporated, where a target DQN is used and updated periodically to resolve the stability issues encountered in the traditional DQN approach. Additionally, experience replay is considered to improve sample efficiency. In the second phase, the trained DQNs make appropriate scheduling decisions, even in unseen cases where some production parameters such as the number of machines and the initial setup times were not faced in the training phase.

Qu et al. (2019) developed a multi-agent AC approach for production scheduling that incorporates experts to guide the exploration of agents in order to improve the convergence of the distributed dynamic scheduling process in manufacturing systems. By following expert advice rather than randomly exploring the environment, the agents will be able to make more informed decisions at the start, leading to an increase in the speed of convergence. Agents select experts who perform better in the scheduling environment, observe their actions, and learn a scheduling policy from their demonstrations.

In Kim et al. (2020), the authors presented a multi-agent DQN-based approach that can learn from the dynamic environment and make better decisions regarding the allocation of jobs and the prioritization of tasks for mass customization. The DQN-based agents corresponding to different manufacturing components evaluate job priorities and schedule them *via* negotiation while continuously learning to improve their decision-making performance.

The framework consists of three layers (enterprise, cloud, and machine layers). There is one enterprise agent (EA) associated with the enterprise layer and two agents associated with the cloud layer, namely, the database agent (DA) and the simulation agent (SA). The machine layer accommodates six types of intelligent agents, including job agents (JAs), negotiation agents (NAs), job weight learning agents (WLAs), job dropout learning agents (DRLAs), job dismiss learning agents (DMLAs), and execution agents (EXAs). These agents interact and cooperate to realize the following five types of functionalities: information sharing (*via* EA, DA, and SA), job index calculation (*via* JA), negotiation (*via* NA), learning (*via* WLA, SRLA, and DMLA), and execution (*via* EXA). This work is primarily centered on the use of RL for negotiation learning to realize communications between the agents in order to save the setup process. In this process, if the sum of the setup and performance times of incorporating more machines causes a longer job completion time than those of doing the job using fewer machines, negotiations will lead the scheduling process to use fewer machines. Figure 6 illustrates the architecture of the smart factory.

Liu et al. (2020) presented a parallel training mechanism under multi-agent settings by using a deep deterministic policy gradient (DDPG) approach and asynchronous updates in an effort to address the potential sources of uncertainty, such as machine breakdowns and unexpected incoming orders. In this work, machines are agents, and the state space includes a processing time matrix, a binary matrix of the jobs assigned to agents, and a binary matrix of completed jobs. These matrices are then used as inputs to a convolutional neural network (CNN), which is similar to the use of CNNs for RGB image processing tasks. The action space is limited to a choice among a set of dispatching rules, such as SPT and FIFO. Additionally, the reward is defined as a function of the process time of the selected job, the remaining process time of the job, and the comparison of the smallest makespan. Figure 7 represents the MARL approach presented in this work.

This work incorporates a global network that updates parameters based on the aggregate gradient of the exploring

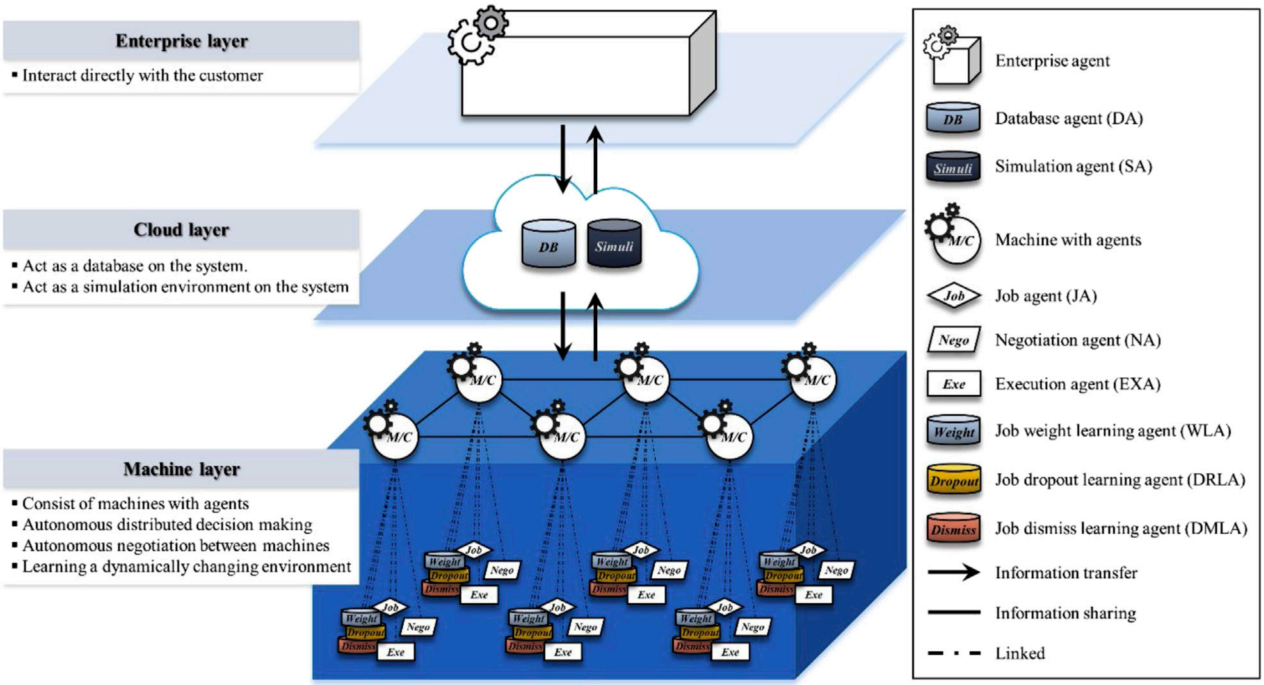

Enterprise layer
▪ Interact directly with the customer
Cloud layer
▪ Act as a database on the system.
▪ Act as a simulation environment on the system
Machine layer
▪ Consist of machines with agents
▪ Autonomous distributed decision making
▪ Autonomous negotiation between machines
▪ Learning a dynamically changing environment
DB
Simuli
M/C
Job
Nego
Exe
Weight
Dropout
Dismiss
Enterprise agent
Database agent (DA)
Simulation agent (SA)
Machine with agents
Job agent (JA)
Negotiation agent (NA)
Execution agent (EXA)
Job weight learning agent (WLA)
Job dropout learning agent (DRLA)
Job dismiss learning agent (DMLA)
Information transfer
Information sharing
Linked

FIGURE 6
Smart factory presented in Kim et al. (2020).

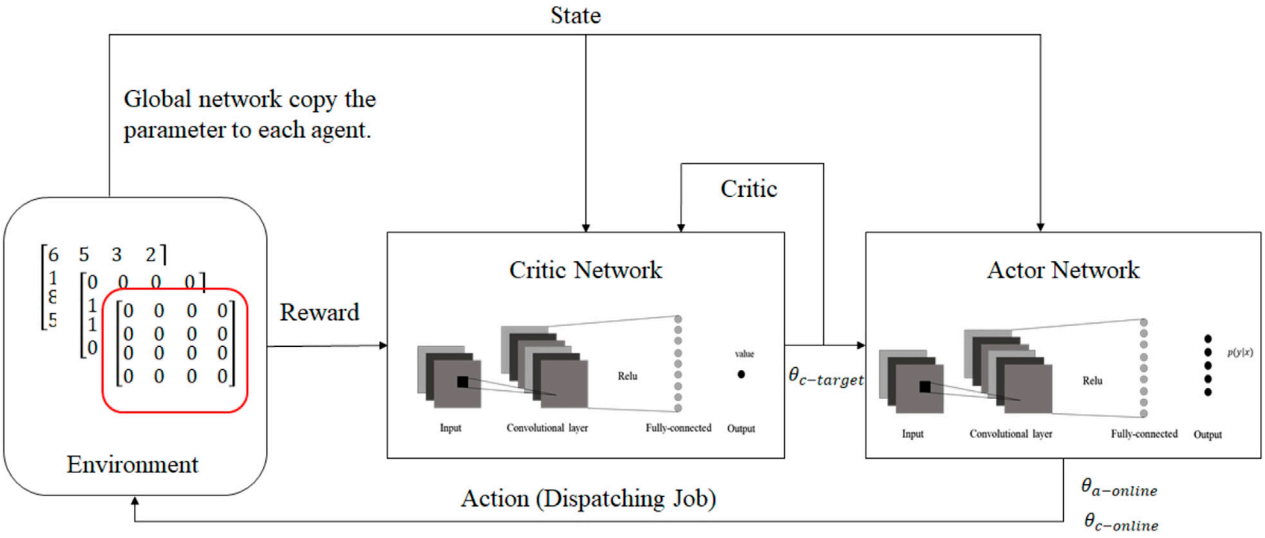


FIGURE 7
MARL method presented in Liu et al. (2020).

agents, and the exploring agents copy weights asynchronously from the global network. To deal with non-stationarity, agents are deployed in separate and parallel environments, each exploring a different part of the problem space, so that they cannot affect each other. Therefore, cooperation between agents under the MARL settings is established through the global network.

Wang (2020) developed a multi-agent weighted QL approach for adaptive job-shop scheduling. The work incorporates several agents, including machine, job, state, and buffer agents. These agents interact with each other to optimize earliness and tardiness in a dynamic manufacturing environment. This work uses the tabular QL form and works based on clustered system states and the degree of difference between the actual state and the cluster's representative in order to avoid explosion of the state space. Actions are taken based on a search algorithm to obtain the maximum Q at the corresponding state (the cluster's representative state). However, a negotiation protocol is used between the agents upon the receipt of a new job. A number of shortcomings of this approach can be identified, such as the need for a long initial exploration phase and the need to select appropriate parameters (for example, the number of clusters).

Baer et al. (2020) presented a multi-agent DQN that uses independent action learners through parameter sharing and an experience replay memory. Agents learn in turn, while the rest are fixed. Agents each incorporate a DQN with an input size equal to the state vector. Each agent consists of one DQN (in the parameter sharing approach) with an input layer for the length of the state vector. All scenarios have a fixed job specification and share an equal local optimization objective.

Dittrich and Fohlmeister (2020) proposed a DQN-based approach for order scheduling in manufacturing systems for minimizing the mean cycle time that implements MARL *via* the use of both local and global rewards (for achieving cooperation between agents). The machine agents only collect data, while the order agents use DQNs to make scheduling decisions. The agents communicate to exchange information and report statuses. This work incorporates the principles of CTDE. Local immediate rewards are given to order agents during the processing of orders, and global rewards redefine Q-values for a global DQN when the orders are complete and new orders are to be initiated. In other words, local DQNs are replaced by the global DQN when a new order is initiated.

In Denkena et al. (2021), the authors improved the work presented in Dittrich and Fohlmeister (2020) by incorporating the agent's field of view and formulating the problem as a Dec-POMDP.

Zhou et al. (2021) proposed a smart factory comprising varieties of components and developed a multi-agent actor–critic approach for decentralized job scheduling. The machines each correspond with an actor–critic agent (enhanced also with target policy and target critic networks), which can all observe all the states of the other agents (*via* communication) in order to deal with the dynamic environment and also deal with non-stationarity. The architecture of the MARL approach for this work is illustrated in Figure 8. The schedulers or the actors in this paper are called the scheduling policy network, and they each schedule their next operation based on the states of the other machine. Thus, they make fully informed decisions.

Pol et al. (2021) addressed the challenge of achieving collaboration between multiple agents in MARL for scheduling purposes in manufacturing systems when dealing with objectives such as makespan minimization. The authors developed a handcrafted empirical logic to quickly estimate a reference makespan that works based on the sum of all operation times per job. They proposed a dense local reward augmented by global reward factors and a sparse global reward to realize cooperation between agents. Their work is based on DQN and simply includes other agents' information in the state space of each agent to let them make more informed decisions when dealing with the dynamically changing environment of multiple agents. Communication is realized implicitly due to the fixed topology of the manufacturing system, and the DQN is shared among the agents to simplify the training process. To deal with the credit assignment problem, training is performed in two phases, that is, first with local rewards and then retaining with the local rewards augmented by a global reward factor. In addition, they proposed the use of sparse rewards given to each agent at the end of episodes, in place of the global reward, in an effort to simplify the process of learning to cooperate (by combining eligibility traces in place of replay memories).

Gankin et al. (2021) presented a multi-agent DQN approach for minimizing production costs in modular manufacturing systems. The environment comprises a grid of 5 × 5 production modules, and AGV units are used to carry products between the modules. AGV units are modeled as agents that use a shared DQN (with an experience buffer for training) to decide where to route jobs, considering the source module. Agents identify themselves based on their location, which is used in the state vector. Action filtering is considered *via* setting too low Q-values to invalidate actions. This work considers scheduling and routing simultaneously.

Wang et al. (2022b) incorporated the QMIX algorithm to develop a MARL approach for scheduling in resource preemption environments. QMIX works on CTDE, which provides agents with local and global objectives. QMIX has a mechanism that encourages the agents to achieve higher global rewards rather than only focusing on local rewards.

Johnson et al. (2022) proposed a multi-agent DQN approach for scheduling the assembly jobs that arrive dynamically in a robotic assembly cell. Their work is based on CTDE, with local independent agents and limited communications between agents to reduce communication costs. Agents each correspond to a robot that distributes jobs over a conveyor belt with a limited window size to workbenches. This work uses a shared DQN enriched by a target network to avoid the overestimation problem encountered with the standard multi-agent DQN architecture. The shared DQN is fed by properly encoded observation vectors to distinguish between the agents and equipped with a filter layer to filter out invalid actions in order to reduce the convergence time.

Gerpott et al. (2022) presented a distributed advantage actor–critic (A2C) method for production scheduling in two-

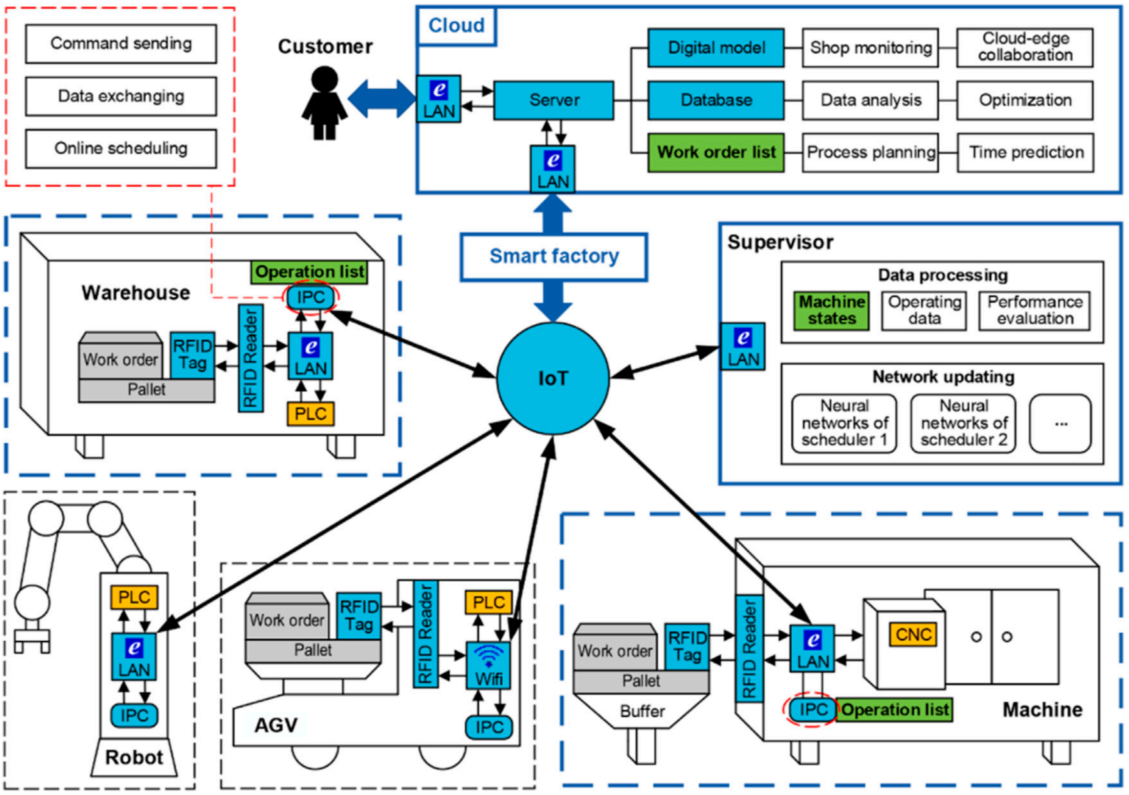


FIGURE 8
Smart factory presented in Zhou et al. (2021).

stage hybrid flow-shop (THFS) manufacturing systems as an attempt to minimize the total tardiness and the makespan. This work uses completely identical scheduling agents that explore different parts of the problem space and share their gradients with the critic. This study uses global parameters that are shared by several agents who explore the environment concurrently. As a synchronous and deterministic method, A2C waits for each agent to complete the corresponding portion of experiments and then performs a global update by taking the average of all gradients received from the actors. A coordinator is used that manages the collection of local gradients and passes them to the global network.Table 1 summarizes the salient attributes of the reviewed applications of MARL to scheduling tasks within smart factories. In the table, the terms "Comm", "Env", "Col", and "Dec" are the shortened forms for "communication quality", "environment", "collaboration method", and "decentralized versus centralized management", respectively. Collaboration quality represents the way in which collaboration is established between the agent, which can be through environment (Env), communication (Comm), parameter sharing (Param), state sharing (SS), a high-level controller (HLC), and asynchronous updates (ASUs).

3.2 Transportation and monitoring (moving agents)

Among the characteristics of smart factories is the large amount of work performed by autonomous moving devices serving a variety of functions, including material handling, monitoring, providing support, and delivering products. Moving robots such as drones, autonomous guided vehicles (AGVs), and overhead hoist transporters (OHTs) are able to move or transport material from one location to another, while the multi-agent setting imposes issues such as uncertainty. MARL has also been incorporated to address moving robots for various functions in smart factories.

3.2.1 Multi-agent pathfinding

Pham et al. (2018) presented a multi-agent QL approach to UAV coordination for the optimal sensing coverage problem. In this work, UAVs are used as mobile sensors to provide visual coverage over a field. Thus, the aim is to coordinate UAVs in such a way that coverage is maximized and overlaps are minimized. The UAVs must then cooperate in order to accomplish the stated objective. A game-theoretical correlated equilibrium mechanism

TABLE 1 Comparison between the MARL approaches incorporated for scheduling in smart factories.

Reference	RL method	Comm	Agent	Env	Metrics	Col	Dec/ Cen
Gabel and Riedmiller (2008)	QL	None	Scheduling agents	MMDP	Makespan	Env	Cen
Wang and Yan (2016)	QL	Explicit	Manufacturing cells	MDP	Run-through time and costs	Env	Cen
Sultana et al. (2020)	A2C	None	Warehouse and store	MDP	Replenishment cost	HLC	Cen
Luo et al. (2021a)	THDQN	None	Work-centers	MDP	Tardiness and utilization	HLC	Cen
Luo et al. (2021b)	HRL	Implicit	Objective, job, and machine agents	MDP	Tardiness, utilization rate, and workload variance	HLC	Cen
Qu et al. (2016)	QL	None	Machines	MDP	Total production costs	Env	Dec
Bouazza et al. (2017)	IQL	Limited	Products	MDP	Waiting time	Enc	Dec
Wang et al. (2017)	Tabular MAQL	None	Machines	MDP	(Mean) lateness, flow time, and late job rate	Env	Dec
Hong and Lee (2018)	A3C	None	Chamber and robots	MDP	Productivity	AsU	Dec
Waschneck et al. (2018a)	DQN	None	Work-centers	MDP	Run-through time and cycle times	Param	Dec
Waschneck et al. (2018b)	DQN	None	Work-centers	MDP	Capacity utilization	Env	Dec
Wang et al. (2018)	SMGWQL	Explicit	Manufacturing cell	MDP	Cost	Comm	Dec
Park et al. (2019)	DQN	Explicit	EA, DA, SA, NA, JA, WLA, DRLA, DMLA, and EXA	MDP	Makespan	Comm	Dec
Qu et al. (2019)	DQN	None	Scheduling agents	Semi-MDP	Work in progress and revenue	Param	Dec
Kim et al. (2020)	DQN	Explicit	Machines	MDP	Productivity and delay	Comm	Dec
Liu et al. (2020)	DDPG	None	Scheduling agents	MMDP	Makespan	AsU	Dec
Wang (2020)	WQ	Explicit	Job, state, machines, and buffers	MDP	Tardiness and run-time	Comm	Dec
Baer et al. (2020)	DQN	None	Independent schedulers	MDP	Makespan	Env	Dec
Dittrich and Fohlmeister (2020)	DQN	Explicit	Orders and machines	MDP	Mean cycle times	Param	Dec
Denkena et al. (2021)	DQN	Limited	Orders and machines	Dec-POMDP	Mean tardiness	Param	Dec
Zhou et al. (2021)	MAAC	Explicit	Machines	MDP	Makespan	Comm	Dec
Pol et al. (2021)	DQN	Implicit	Products	DEC-POMDP	Makespan	SS	Dec
Gankin et al. (2021)	MA-DQN	None	Workstations	MDP	Production rate	Param	Dec
Wang X. et al. (2022)	QMix	None	Jobs	DEC-POMDP	Makespan	QMix	Dec
Johnson et al. (2022)	DQN	Limited	Assembly cells	MDP	Makespan	Param	Dec
Gerpott et al. (2022)	A2C		Parallel scheduling agents	MDP	Tardiness and makespan	Param	Dec

and a function approximation are used to address the challenges of joint-action selection and the high dimensions of the problem. The problem is simulated in a 3D environment of identical cubic cells and formulated as a general locational optimization problem. The joint-action selection problem requires that the agents reach a consensus, and thus the correlated equilibrium (which can be solved using linear programming) was used to evaluate the agreement regarding the selection of the joint-action set. This paper uses fixed sparse representation (FSR) and the radial basis function (RBF) as an attempt to map the original Q to a parameter vector θ by using state- and action-dependent basis functions ϕ.

In order to address the multi-agent path finding (MAPF) problem, Sartoretti et al. (2019) incorporated A3C to introduce a MARL approach called PRIMAL (pathfinding *via* reinforcement and imitation multi-agent learning). PRIMAL integrates RL with imitation learning (IL) to enable agents to learn the homogeneous path planning policy needed for online coordination in a POMDP environment. IL serves as a centralized planner and learns the impact of actions on the agents and the team as a whole in order to train the agents for coordination (behavioral cloning) in order to eliminate the need for explicit communication. PRIMAL was implemented in a partially observable discrete grid world with a limited field of view (FOV), meaning that the agents have local observations. Agents are modeled *via* a CNN of seven convolutional layers, followed by an LSTM, to approximate the individual agent's policy. Collision results in a penalty, and a large positive reward is obtained upon the achievement of the goal. A3C trains the agents *via* local states, which might result in selfishness (locally optimized decisions), and thus, randomized environments were used to avoid selfishness. PRIMAL does not allow agents to follow the others. Thus, this leads to a reduced collision chance. More details on A3C can be found in Mnih et al. (2016).

Qie et al. (2019) proposed a multi-agent DDPG (MADDPG) approach for the multi-UAV task assignment and path planning problem. While the two tasks are optimization problems that are commonly addressed separately in dynamic environments, a large number of recalculations are required to be performed in real-time. In spite of the fact that all of the UAVs are identical, the formation of the UAVs over the distributed locations of the targets should be optimized to reach the total flight distance, taking into consideration the presence of risky areas in the field and the likelihood of collisions. During the training phase, all agents have access to the observations and actions of other agents through a distributed actor–critic architecture (DDPG uses actor–critic at its core). Actors see local observations, whereas critics have access to the entire observation space (each agent has an actor and a critic). During the execution phase, only actors are active in the field, which means that the execution process is decentralized.

Zhiyao and Sartoretti (2020) proposed $PRIMAL_c$, which extended the PRIMAL's search space from two dimensions to three dimensions. $PRIMAL_c$ suggested the use of the capacity of agent modeling to enhance the performance of path planning *via* the prediction of the actions of other agents in a decoupled manner. However, learning others' behaviors introduces the problem of prediction inaccuracies and would appear to be ineffective in practice.

Zhang et al. (2020b) proposed a decentralized multi-agent actor–critic-based framework that leverages the multi-step-ahead tree search (MATS) strategy to address the AGV pathfinding problem. To address scalability for a large number of agents while maintaining the response time within a predetermined range, experiments were conducted in a real-world warehouse. Different from PRIMAL (which avoids letting agents follow the others), this work allowed agents to follow the others to improve the job completion rate while avoiding collisions *via* the incorporation of MATS and post processing the actions. MATS assists in finding the possible actions from other agents so that possible collisions could be predicted and avoided by reducing the probability of taking actions that result in collisions. This work was able to outperform PRIMAL when applied to a real-world warehouse case.

Malus et al. (2020) incorporated the twin-delayed deep deterministic (TD3) policy gradient algorithm for autonomous mobile robots (AMRs) scheduling to address the complexities encountered due to rapid changes in the production environment and the tight relationships between dispatching and routing (planning and execution) problems. AMRs are distinguished from AGVs by their navigational capabilities. AMRs are equipped with sensory devices that detect the surrounding static and dynamic objects, allowing them to navigate and localize autonomously. Controlling a group of mobile robots is called a fleet management system (FMS), which performs tasks such as transportation, order dispatching, routing, and the scheduling of job executions. The problem with AMRs is that they are often tightly coupled, resulting in a system of immense computational complexity. Consequently, centralized approaches to managing fleets of AMRs often fail to provide real-time routing and dispatching at the same time. TD3 is an extension of the DDPG algorithm (both work on the basis of the actor–critic algorithm) used for continuous action-space problems wherein neural networks are incorporated to concurrently approximate two Q-networks and a policy network. In this work, with TD3, agents see local and partial observations, and each has a dedicated policy for mapping local states to actions. An agent's action involves a bidding value between 0 and 1, where the highest value between the two currently assigned orders is taken (one for execution and one for the next processing step for real-time constraints). The reward design considers the cooperative nature of the agents, and all the agents receive the same reward/penalty. The order, once received by the agent, is added to the agent's queue of orders, and the corresponding AMR vehicle performs it autonomously in a first-in-first-out manner. In the event that an order is completed within the time constraint, all agents receive a positive constant reward. Otherwise, they will be subjected to a penalty that increases quadratically with respect to tardiness.

Damani et al. (2021) later proposed PRIMAL2 to address lifelong multi-agent pathfinding (LMAPF) for applications such as smart warehousing in smart factories. LMAPF is a variant of MAPF, in which agents are assigned a new goal as soon as their current objective is reached in a dense and structured environment such as industrial warehouses. PRIMAL2 suggests the use of convention learning to enable the agents to learn a generalizable policy. Identifying certain conventions and forcing agents to learn them can enhance performance. They also incorporate environment randomization (sampling from a variety of environments during training) to enable the agents to learn to deal with different environments. Nevertheless, they assume that the tasks are sparsely distributed across random locations, thereby eliminating local congestion. PRIMAL2, like previous versions of PRIMAL, has a long training time.

Shen et al. (2021) combined the multi-agent asynchronous advantage actor–critic (MA-A3C) with an additional attention mechanism for the multi-AGV pick and place problem in smart warehousing systems. This attention mechanism allows attention to be focused on the beneficial information that arises from the interaction between AGV units in order to increase learning efficiency and minimize the corresponding complexity. They also used CTDE to address the dynamic Markov environment and non-stationarity. By incorporating the attention mechanism, this work is able to select AGV units dynamically during the training and thus improve the collaboration between AGV agents. The experiments were conducted on the Amazon Kiva system, consisting of a picking table, shelf, and AGV. Five possible choices are available in this system: up, down, right, left, and stay. The experiments reported that this method outperformed MAAC, MADDPG, MADDPG + SAC, and COMA + SAC.

Choi et al. (2022) incorporated QMIX for the cooperative control of AGVs in smart factories' warehouses. An agent can choose an action in a grid-like environment (move forward, backward, to the left, to the right, and to stop) and receive an individual reward for its action. The reward is positive if the Manhattan distance to the target is reduced as a result of the action taken. CTDE was incorporated to address both the scalability issue of centralized learning and the non-stationarity of a fully decentralized learning process at the same time. The method is able to evaluate individual agents' contributions due to the receipt of both the individual reward Q_a and the global reward Q_{tot} from the environment in QMIX. This method was able to outperform IQL.

Yun et al. (2022) incorporated an actor–critic method called CommNet for the deployment of CCTV-equipped multi-UAVs with a focus on autonomous network recovery to ensure reliable industry surveillance. As mobile CCTV UAVs can continuously move over a wide area, they provide a robust solution for surveillance in dynamic manufacturing environments. In order to enhance the surveillance performance, the study aims to improve the energy consumption of surveillance drones. In this case, surveillance drones are deployed in heavily populated areas. This work involves a single UAV serving as the communication leader, some UAVs serving as agents, and some targets serving as surveillance targets. Communication between UAV agents is handled by the leader UAV agent. The agents observe the local surroundings, take joint actions, and are rewarded individually and jointly for their cooperative efforts.

3.2.2 Pathfinding + scheduling

Mukhutdinov et al. (2019) developed a multi-agent IQL approach for material handling in smart factories, inspired by the packet routing problem in computer networks. In this work, routing hubs are considered intelligent agents, which are equivalent to routers in computer networks. The formulation of this routing system is considered a graph, where nodes are router agents and edges are the paths between the agents. Each agent only observes its neighbor agents, and actions are defined as choosing between the outgoing edges. Each router agent has a DQN component for the approximating function Q_v (S_v, u), which is the estimation of the minimal cost of the path from the routing agent v to the destination of the current node d *via* a neighbor u. A reward for action is the negated cost of the edge over which the packet has been sent: $r = -Cost\ (e, S_e)$. By modeling each router using a DQN, each router is able to account for heterogeneous data about its environment, which allows for the optimization of more complicated cost functions, such as the simultaneous optimization of bag delivery time and energy consumption in a baggage handling system.

Zhang et al. (2020a) proposed a centralized multi-agent DQN approach for the open-pit mining operational planning (OPMOP) problem (an NP-hard problem that seeks to balance the tradeoffs between mine productivity and operational costs), which works based on learning the memories from heterogeneous agents. Open-pit mine dispatch decisions coordinate the route planning of trucks to shovels and dumps for the loading and delivery of ore. The queuing of trucks as a result of a high truck arrival rate and the starvation of shovels as a result of a low truck arrival rate can both negatively impact productivity. Instead of being restricted to fixed routes, trucks can be dispatched to any shovel/dump in a dynamic allocation system. An appropriate dispatch policy should minimize both shovel starvation and truck queuing. This work presents an experience sharing DQN in order to provide a shared learning process for heterogeneous agents and also to deal with unplanned truck failures and the introduction of new trucks (without retraining). A post-processing step and a memory tailoring process were used to enable the DQNs to be trained by the samples obtained from trucks of diverse properties (heterogeneity).

Li et al. (2021) proposed MADDPG-IPF (information potential field) as a means of enhancing the adaptability of AGV coordination to different scenarios in smart factories for material handling. Typically, raw materials used in manufacturing workshops are stored at various locations throughout a warehouse. Thus, AGVs have to visit a number of locations in order to coordinate transportation tasks. To reach different targets, all AGVs must avoid collisions and self-organize as quickly as possible. The authors address the problem of reward sparsity *via* the incorporation of the information potential field (IPF) in the reward-shaping strategy, which brings stepwise rewards and implicitly leads AGVs toward material handling targets.

3.2.3 Mobile operator

Karapantelakis and Fersman (2021) developed a multi-agent deep IQL-based approach to provide connectivity coverage services *via* distributed mobile network operators. To respond to the dynamic demands on mobile networks, mobile operators collaborate as individual agents. Agents have full observability of their environment and train their deep recurrent Q-network (DRQN) independently with respect to a common joint reward function. In order to mitigate the non-stationarity imposed by IQL, a cyclic replay memory (replacing old memories with recent ones) and a global target network are used.

3.2.4 Overhead hoist transporters

Ahn and Park (2019) incorporated a graph neural network (GNN)-based factorized policy gradient (GNN-FPG) method based on the factorized actor-factorized critic (fAfC) method for the cooperative rebalancing of overhead hoist transporters (OHTs) as an attempt to enhance the productivity (reduce the lead, delivery, and retrieving times) of the material handling process in the semiconductor fabrication (Fab) system. OHTs are used to transport semiconductor wafers between machines and are considered to be an essential component of an automated material handling system (AMHS). Generally speaking, OHT refers to an automated transport system that travels on an overhead track *via* a belt-driven hoisting mechanism that facilitates direct access to the load port of the stocker. This work proposes a MARL algorithm for dispatching, routing, and rebalancing these OHTs. The problem of OHT rebalancing is quite similar to the problem of empty vehicle redistribution (EVR) in a traffic system. In this work, the Fab is discretized into a number of zones, and decentralized rebalancing strategies are developed for the idle OHTs of each zone (idle OHTs are assigned to new zones) in order to minimize the lead (retrieval) time and congestion. This work uses a collective decentralized partially observable Markov decision process (CDec-POMDP) for which the objective is to obtain a decentralized policy with respect to local observations in order to achieve the system-level goal. The proposed cooperative rebalancing strategy accepts the distributions of idle OHTs, working OHTs, and the loads (delivering tasks) over discretized zones in the Fab as an input and outputs decentralized rebalancing strategies for each zone.

Ahn and Park (2021) proposed a factorized actor–critic (FAC) method for establishing cooperative zone-based rebalancing (CZR) for OHTs in the semiconductor industry. The objective of this work is to reduce the average retrieval time and the OHT utilization ratio by incorporating graph neural networks. Using joint state and joint action information, a central model learns the interactions between the agents and the corresponding future accumulated shared return. Using only local observations and communication information, the learned policy is executed independently by the agent. The rebalancing problem is formulated as a partially observable Markov game (POMG), in which the Nash equilibrium policy of the game is to be determined. The agents in a stochastic game strive to obtain the policy that maximizes the expected accumulated reward. Decentralized optimal control was reformulated as a general stochastic game by the authors.

3.2.5 Pick and place

Zong et al. (2022) proposed a multi-agent A2C approach for the cooperative pickup and delivery problem (as a variant of the vehicle routing problem) for warehousing in smart factories. The work also provides paired delivery, which implies that a vehicle might take more than one product part to deliver to more than one destination. In order to deal with the structural dependencies imposed between deliveries, this work incorporates a paired context embedding architecture based on the transformer model (Vaswani et al., 2017). A2C (with a joint critic and individual actors) and communication were used to build a centralized architecture. While different agents generate their individual policies (actors), they share the paired context embedding and context encoding within the centralized architecture of A2C.

Lau and Sengupta (2022) developed a shared experience actor–critic (SEAC) approach for the lifelong MAPF problem. The work was formulated as a partially observable Markov decision process, with the MARL's aim being to determine the optimal joint policy of the agents. By sharing experiences, agents can learn from one another's experiences without receiving the same rewards. In SEAC, the trajectories collected from other agents are incorporated for off-policy training, while importance sampling with a behavioral policy was used to correct the off-policy data.

Table 2 compares the literature in relation to the applications of MARL to transportation within smart factories in terms of its salient characteristics. In the table, the terms "Comm", "Env", "Col", and "Dec" are the shortened forms for "communication quality", "environment", "collaboration method", and "decentralized versus centralized management", respectively. Collaboration quality represents the way in which collaboration is established between the agent, which can be through environment (Env), communication (Comm), parameter sharing (Param), state sharing (SS), a high-level controller (HLC), and asynchronous updates (ASUs).

3.3 Maintenance

Considering various reasons such as machine breakdowns and deadlocks, smart factories also need automated and efficient maintenance strategies. MARL has been applied to maintenance problems in the relevant literature to provide a more automated solution to the uncertainty faced in the maintenance process. Wang et al. (2016) presented a tabular multi-agent QL approach for making maintenance decisions in a two-machine system. The presented method aims to make the agents learn the control-limit maintenance policy for each machine associated with the observed state represented by the yield level and buffer level. Due to the non-synchronicity of the state transitions between both machines, an asynchronous updating rule is also incorporated in the learning process. Zinn et al. (2021) presented a MARL system based on DQN and actor–critic to learn the distributed fault-tolerant control policies for automated production systems during fault recovery to increase availability. Liu et al. (2022) proposed a multi-agent DQN approach to make maintenance scheduling decisions for personnel and also production control during maintenance. This work

TABLE 2 Comparison between the applications of MARL to transportation tasks in smart factories.

Reference	RL method	Comm	Agent	Env	Metrics	Col	Dec/ Cen
Multi-agent pathfinding							
Pham et al. (2018)	MAQL	Implicit-limited	UAVs	Markov game	Coverage	Param	Cen
Sartoretti et al. (2019)	A3C	Implicit	Moving robots	POMDP	Travel distance and success rate	AsU	Dec
Qie et al. (2019)	MADDPG	None		MDP	Travel costs and collision	Param	Dec
Zhiyao and Sartoretti (2020)	A2C	Explicit	Moving robots	POMDP	Travel distance and success rate	AsU	Dec
Zhang Y. et al. (2020)	AC	None	AGV units	MDP	Collision rate and job completion rate	Param	Dec
Malus et al. (2020)	TD3	None	AMR agents	POMDP	Completion time	Param	Dec
Damani et al. (2021)	A3C	Implicit	Moving robots	POMDP	Travel distance and success rate	Param	Dec
Shen et al. (2021)	MAA3C	None	AGV units	MDP	Collision rate and travel distance	Comm	AsU
Choi et al. (2022)	QMix	Explicit	AGV units	Dec-POMDP	Path length and success rate	QMix	Dec
Yun et al. (2022)	AC	Explicit	UAVs	MMDP	Coverage	Comm	Dec
Multi-agent pathfinding	+scheduling						
Mukhutdinov et al. (2019)	IQL	None	Routing units	POMDP	Delivery time and energy	Env	Dec
Zhang C. et al. (2020)	EM-DQN	None	Trucks	POMDP	Production rate, cycle period, and matching factor	Param	Dec
Li et al. (2021)	MADDPG-IPF	Explicit	AGV units	POMDP	Task response time	Param	Dec
Mobile operator							
Karapantelakis and Fersman (2021)	IQL and DRQN	Explicit	Operator agents	Dec-POMPD	Service fulfillment	Env	Dec
OHTs							
Ahn and Park (2019)	fAfC	None	OHT zones	CDec-POMPD	Lead time and congestion	Param	Dec
Ahn and Park (2021)	Factorized AC	Explicit	Zone agents	POMG	Retrieval time and utilization ratio	Param	Dec
Pick and place							
Zong et al. (2022)	A2C (central critic)	Share info	AGV units	MDP	Travel distance	Param	Dec
Lau and Sengupta (2022)	SEAC	None	Vehicles	POMDP	Flow time, makespan, and delivery rate	Param	Dec

incorporates a CNN-LSTM-based architecture for the DQN, while the impacts of agents are neglected. Su et al. (2022) presented a MARL approach using value decomposition actor–critic (VDAC) to enable physical machines (in a serial production line that requires multiple levels of machine decisions) to learn local maintenance policies in a distributed and cooperative manner. The proposed solution is formulated as a DEC-POMDP problem, and CTDE was used to provide the solution. Action masking is also incorporated to filter invalid actions. In VDAC, distributed actors make decisions for designated machines, while a central critic estimates the global state value.

3.4 Energy

The dependence of smart components in smart factories on electrical energy, together with the need for cost-effective, reliable, and efficient energy supplies, has led to the use of smart grids that are adaptive and can distribute energy in an on-demand manner. MARL has also been applied to smart grids. Smart grids and smart manufacturing systems share common properties/objectives, such as communication, integration, and automation, which define the commonalities of their applications. Samadi et al. (2020) presented a tabular MA-IQL approach for decentralized energy management in smart grids and proposed a system including high-level energy management agents, low-level heterogeneous resource agents, and consumer agents. The agents adapt themselves to maximize their profits without communication in the grid environment. Wang et al. (2022c) presented an MA-DDQN approach, named P-MADDQN, for resilience-driven routing and the scheduling of mobile energy storage systems (MESSs). This work formulates the problem of POMG and MESS agents interacting with the environment and making independent decisions for simultaneous routing and scheduling based on local information. Charbonnier et al. (2022) presented a tabular MA-QL approach for energy coordination management.

Agents are proactive consumers, and local observations are modeled in a Dec-POMDP environment; they make individual decisions to find a trade-off between local, grid, and social objectives. Bollinger and Evins (2016) presented a MADQN and another MARL approach known as the continuous actor–critic learning automaton (CACLA) for optimizing technology deployment in distributed multi-energy systems. The work is based on technology agents, building agents, a grid agent, and a market agent. Alqahtani et al. (2022) presented a multi-agent AC approach for the energy scheduling and routing of a large fleet of electric vehicles (EVs) in a smart grid to address the power delivery problem. It is possible to use a fleet of EV batteries as a source of sustainable energy since they are capable of storing solar energy and discharging it to the power grids later on, which can result in lower energy costs. It is important to note that the use of a fleet of electric vehicles for power generation is only effective if they are dispersed appropriately across the area of need. The MARL approach is incorporated to address the joint problem of vehicle routing (VR) and energy dispatching (ED), with a special focus on enhancing the scalability and addressing the complexities involved. The problem is formulated as a DEC-MDP problem, while the vehicle's position, the vehicle's state of charge, solar irradiance, and power load are used as state variables. In addition, the action set includes mobility actions (up, down, left, right, or stay still) in the grid and energy dispatch decisions (charging, discharging, and idle). This paper uses actor–critic, where the actors are local and the critic is shared among the agents.

3.5 Human–robot collaboration

Humans play an important role in manufacturing systems, and their impact on the manufacturing environment is significant when designing smart factories. As a result, due to the unpredictable and dynamic behavior of human workers, their role should not be considered as a stationary part of the environment, while this fact has mainly been neglected in the relevant literature. HRC is a broad field, and the application of RL to HRC should be formulated in multi-agent settings. In this study, due to the limitations, we only bring some representative examples and leave the comprehensive study for future work.

Yu et al. (2021) presented a MA-DQN approach for scheduling human and robot collaborative tasks to optimize the completion time in an assembly chess board simulation of the manufacturing environment. The agent learns the optimal scheduling policy without the need for human intervention or expert knowledge, using a Markov game model. Wang et al. (2022a) incorporated a multi-agent extension of generative adversarial imitation learning (GAIL) to generate a diverse array of human behaviors from an example set. The behavior is then used in a MARL approach to account for the human during the human–robot handover and for the multi-step collaborative manipulation tasks. An approach presented by Zhang et al. (2022) generates the appropriate action sequence for humans and robots in collaborative assembly tasks using a MADDPG approach. A real-time display of the agent–human's behavior is shown to the operator. In this scenario, the operator would be able to carry out the assembly task in accordance with the planned assembly behavior under the globally optimal strategy for the expected performance.

3.6 Other applications

Chen et al. (2021) presented a multi-agent DDPG approach for the coordinated welding of multiple robots with a continuous action space and local observations. Lan et al. (2021) studied the application of MARL to multi-robot pick and place problems in Dec-POMDP environments and suggested the use of the variants of MADQN and DRQN in combination with CTDE. Ji and Jin (2022) proposed a MARL system based on independent DQN agents in an attempt to capture self-organizing knowledge for developing multi-robot self-assembly systems. They obtained superior results for decentralized teamwork rather than a centralized approach.

3.7 Discussion and potentials

In this section, we review the applications that used MARL to address control problems in smart factories, with a main focus on scheduling and transportation and a brief review of maintenance, energy, and human–robot collaboration. Upon close examination, we can observe patterns for how MARL, which was a significantly difficult approach to implement, was able to be used for different tasks in smart factories. In this section, we identify a number of concerns and analyze them based on a review of the works provided earlier. The main concerns are the choice of the MARL method, scalability, problem formulation, scalability, convergence, a cooperation strategy, and information exchange. For the most part, MARL problems are significantly prone to non-stationarity, and different approaches showed certain ways to deal with this problem. Joint actions are one of the main approaches used to deal with non-stationarity, but this has drawbacks such as the difficulty of designing the exploration strategy due to the large joint action space. The other approach used was a profound support of agents, with all the necessary information about the states and actions of the other agents. However, the corresponding information sharing costs reduce the scalability and efficiency of this approach. A close look at the most comprehensive approaches that incorporate a larger number of agents, such as Kim et al. (2020) and Zhou et al. (2021), demonstrates the fact that scalable approaches tend to

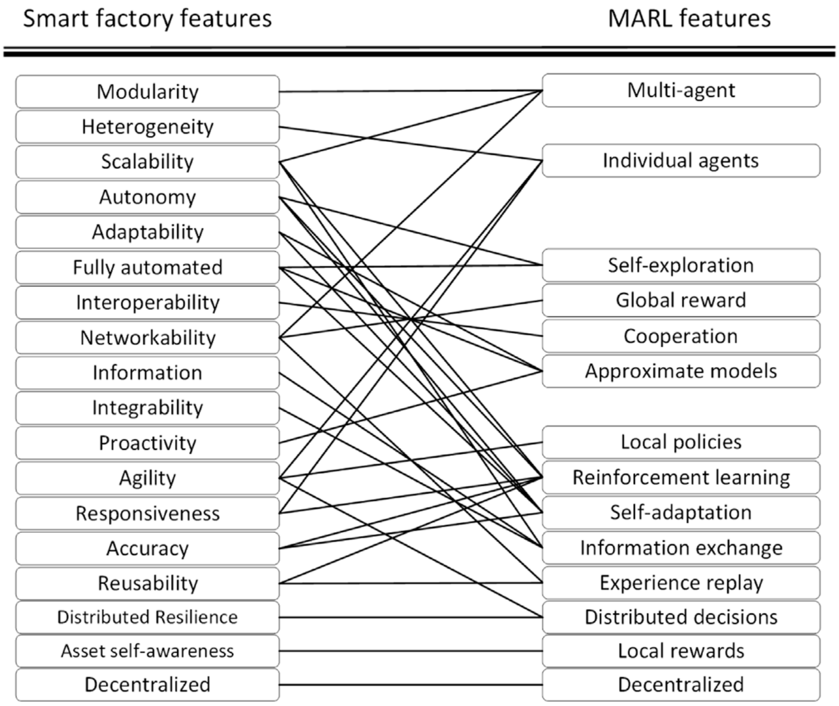


FIGURE 9
Matching between smart factory features and MARL features.

use more sparse communication strategies where agents are selectively chosen to exchange information to save time and cost. A more advanced approach, however, is the use of negotiation-based strategies incorporated in Wang (2020) and Kim et al. (2020), which can even be enhanced by taking advantage of negotiation learning. Furthermore, even those applications that used independent learners incorporated some techniques to deal with non-stationarity, such as asynchronous updates, CTDE, exploring separate parts of the environments in parallel with the aggregate gradient, and behavior forecasting. However, as the dynamicity of the environment grows, adequate information should be exchanged between agents for making informed decisions; otherwise, impacts from other agents on the environment cannot always be avoided or predicted. Therefore, we conclude that establishing spare communications is the recommended solution for both efficiency and rapidity. Collaboration is another important concern in MARL applications, which is mainly realized *via* the definition of a global reward. However, the credit assignment problem is challenging in this regard, since decomposing the reward to determine the share of each agent is not always possible. Without consideration of the credit assignment problem, it is possible that some of the agents become lazy and have a negative impact on the global performance of the system. Nonetheless, approaches such as QMIX suggest a dedicated approach to considering both global and local rewards in the process of learning. However, in real-time environments, QMIX might not show a good performance since an agent's action might become dependent on the actions taken by other agents. The choice of the MARL method is also another concern, as it has an impact on the convergence and also on other performance factors. DQN, as the most popular approach, has certain limitations, such as the dimensionality and continuity of the state–action space. Altogether, AC-based methods such as MADDPG have other limitations such as slow convergence due to a larger number of parameters included in the learning process. As the last point to discuss in this section, the choice of state variables and the corresponding concept behind the definition of an agent have a significant impact on the performance of MARL systems. Considering the fact that the most successful applications of MARL incorporate supervised learning techniques such as NNs (and DNNs) for function approximation, the complexity of the problem grows significantly in the event of the choice of ineffective state variables. A correlation between these variables, randomness, anomalies, duplicates, low content, the irrelevance of a variable, and the number and usage of the variables, all influence the performance of the MARL system. As an example, the one hot encoding technique is one of the main approaches that should

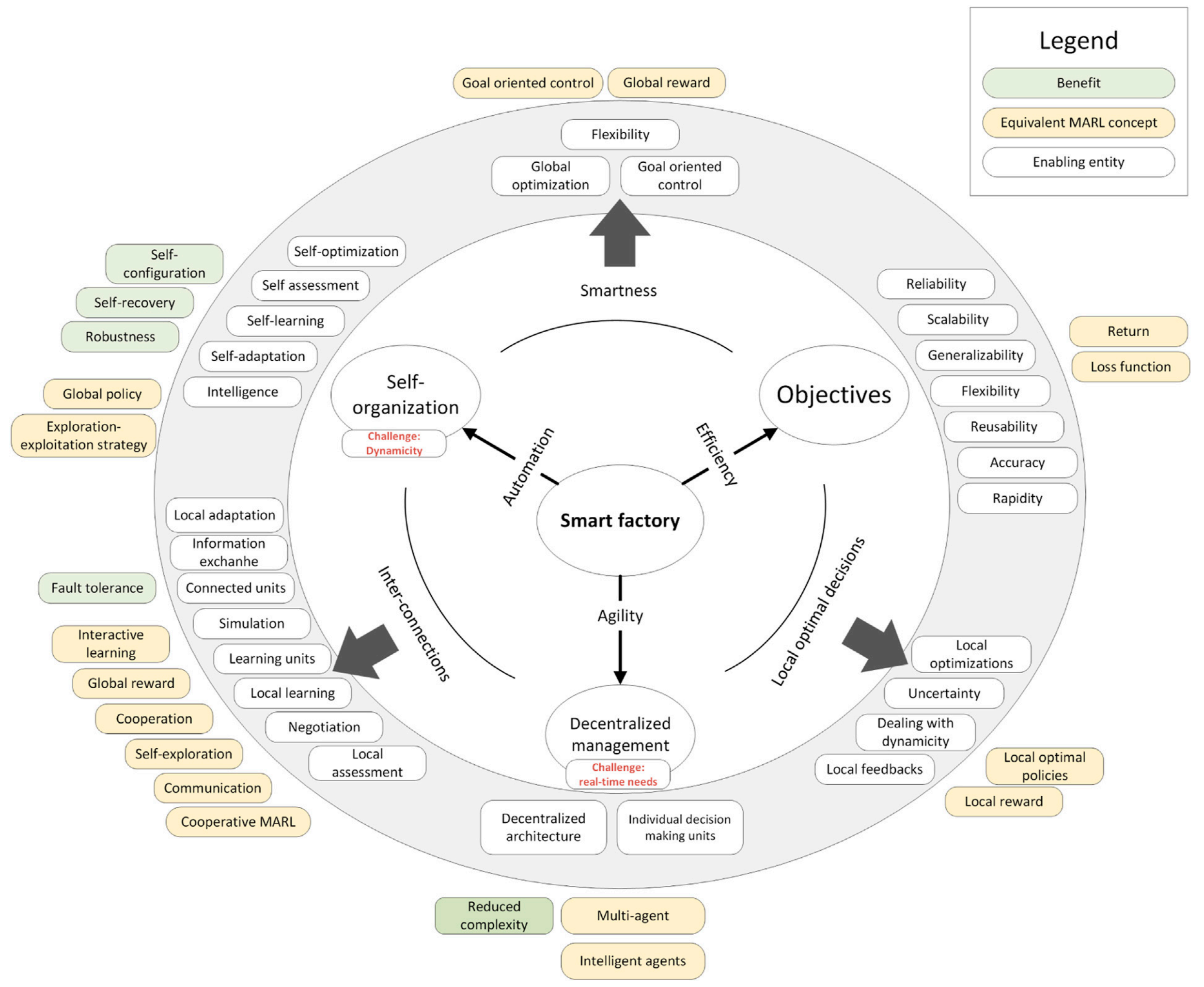


FIGURE 10
Mapping from smart factory features and MARL features.

be incorporated when dealing with categorical state variables to be used as an input to a DNN.

4 Mapping from smart factory features to multi-agent reinforcement learning features

In the previous section, we reviewed the literature on the applications of MARL to tasks within a smart factory and noticed that there are various ways to convert these tasks into a MARL problem. As an early conclusion, considering the presence of uncertainty from various sources in the dynamic environment of smart factories, MARL has natural potential for dealing with uncertainty in smart factories in an automated manner. Many of the works reviewed in the previous section offer their solutions through POMDP problem environments that explicitly consider partial observability in their problem formulation, meaning that uncertainty is naturally considered with such applications. Furthermore, it is promising that MARL has a natural counterpart for almost all of the features required for establishing optimal management in smart factories, with all being provided at the same time and focusing on self-organization. Mittal et al. (2019) identified 22 main characteristics for smart factories, including digital presence, modularity, heterogeneity, scalability, context awareness, autonomy, adaptability, robustness, flexibility, fully automated, asset self-awareness, interoperability, networkability, information, appropriateness, integrability, sustainability, compositionality, composability, proactivity, reliability, agility, responsiveness, accuracy, reusability, decentralized, and distributed resilience. Regarding these features, focusing on the features that are explicitly important in the control mechanism for smart factories, we provide a match between the features required for a smart factory and the equivalent enabling features offered by MARL approaches, as shown in Figure 9.

Above all these characteristics, borrowed from Mittal et al. (2019), there are three leading factors in smart factories that are directly provided through the control mechanism, including automation, agility, and efficiency. Based on the review that we provided earlier in Section 3, we propose a mapping from the requirements of a smart factory initiated by these leading factors to the equivalent MARL features, shown in Figure 10.

As shown in Figure 10, the smart factory is mainly used to provide automation, efficiency, and agility. Agility is realized *via* decentralized decision-making by individual units, since the centralized form imposes delay as a result of the need for multi-objective optimization on a large number of factors and massive communications (regardless of the possibility of information loss) between the components. A decentralized architecture and individual decision-making units will both reduce the complexity of a manufacturing system. The equivalent concept with regard to these features of a smart factory is the multi-agent setting and the intelligent agents in a MARL framework. Automation at the highest level is provided *via* self-organization, where intelligence, self-centered assessment, optimization, learning, and adaptation are incorporated to deal with the dynamicity encountered in the manufacturing environment. These features, when established, can provide the manufacturing environment with the abilities of self-configuration, self-recovery, and robustness. The equivalent concepts in MARL for establishing these features are the global policy that controls the harmony of the entire system by defining the high-level long-term and short-term control directions for the low-level agents and the exploration–exploitation strategy that determines how and how often the system tries to explore novel behavior. Efficiency, as the main factor around which the entire solution is built, is highly dependent on the objectives defined by the application. Efficiency in practice is a multi-objective optimization concept, meaning that multiple factors contribute to the efficiency of the solution. The important characteristics that have a direct relationship with efficiency in smart factories are scalability, generalizability, flexibility, accuracy, and rapidity, while reliability and reusability can be considered as constraints when developing a control mechanism for a high-level or low-level task within a smart factory. Additionally, the equivalent concepts in a MARL framework, with regard to the mentioned objectives, are the return, which is the discounted cumulative future reward, and the loss function that appears in the training of deep neural networks used for complex MARL problems.

As shown in Figure 10, we have also identified the features that should be considered when each two of the three main factors are concerned. In order to not step beyond the scope of the study and also due to space limitations, we postpone the detailed analysis of these features to a future study.

5 Conclusion

In this study, we reviewed a wide range of applications that incorporated MARL into the tasks within a smart factory from a technical and analytical perspective. Specifically, MARL applications were studied for tasks including scheduling, transportation, maintenance, energy management, and human–robot collaboration, while the main focus was devoted to the first two categories. For the scheduling and transportation applications, we provided a comparative analysis representing how different MARL characteristics are chosen to implement the corresponding MARL solution. We also demonstrated how different aspects of smart factories match the objectives and capabilities of MARL and suggested a mapping from smart factory features to the equivalent concepts in MARL, indicating how MARL provides an appropriate solution to provide almost all the required features in the smart factory at once. Our investigations in this paper suggest that MARL is one of the most appropriate AI techniques for implementing tasks in smart factories, for the most part due to its natural ability to deal with uncertainty in multi-agent and decentralized systems, in a self-organized manner.

Author contributions

FB was the main author, and DR provided support and feedback as the director of the Smart Production Systems group, at HTW Dresden, Saxony, Germany.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ahn, K., and Park, J. (2019). "Idle vehicle rebalancing in semiconductor fabrication using factorized graph neural network reinforcement learning," in 2019 IEEE 58th Conference on Decision and Control (CDC) (Nice, France: IEEE), 132–138.

Ahn, K., and Park, J. (2021). Cooperative zone-based rebalancing of idle overhead hoist transportations using multi-agent reinforcement learning with graph representation learning. *IISE Trans.* 53, 1140–1156.

Alqahtani, M., Scott, M. J., and Hu, M. (2022). Dynamic energy scheduling and routing of a large fleet of electric vehicles using multi-agent reinforcement learning. *Comput. Ind. Eng.* 169, 108180. doi:10.1016/j.cie.2022.108180

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* 34, 26–38. doi:10.1109/msp.2017.2743240

Baer, S., Turner, D., Mohanty, P., Samsonov, V., Bakakeu, R., and Meisen, T. (2020). "Multi agent deep q-network approach for online job shop scheduling in flexible manufacturing," in International conference on manufacturing system and multiple machines, Tokyo, Japan.5

Bollinger, L. A., and Evins, R. (2016). "Multi-agent reinforcement learning for optimizing technology deployment in distributed multi-energy systems," in 23rd International Workshop of the European Group for Intelligent Computing in Engineering, Krakow, Poland.

Bouazza, W., Sallez, Y., and Beldjilali, B. (2017). A distributed approach solving partially flexible job-shop scheduling problem with a q-learning effect. *IFAC-PapersOnLine* 50, 15890–15895. doi:10.1016/j.ifacol.2017.08.2354

Büchi, G., Cugno, M., and Castagnoli, R. (2020). Smart factory performance and industry 4.0. *Technol. Forecast. Soc. Change* 150, 119790. doi:10.1016/j.techfore.2019.119790

Buşoniu, L., Babuška, R., and Schutter, B. D. (2010). "Multi-agent reinforcement learning: An overview," in *Innovations in multi-agent systems and applications-1*, 183–221.

Charbonnier, F., Morstyn, T., and McCulloch, M. D. (2022). Scalable multi-agent reinforcement learning for distributed control of residential energy flexibility. *Appl. Energy* 314, 118825. doi:10.1016/j.apenergy.2022.118825

Chen, W., Hua, L., Xu, L., Zhang, B., Li, M., Ma, T., et al. (2021). "Maddpg algorithm for coordinated welding of multiple robots," in 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 1–5.

Choi, H.-B., Kim, J.-B., Ji, C.-H., Ihsan, U., Han, Y.-H., Oh, S.-W., et al. (2022). "Marl-based optimal route control in multi-agv warehouses," in 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (Jeju Island, Korea: IEEE), 333–338.

Damani, M., Luo, Z., Wenzel, E., and Sartoretti, G. (2021). PRIMAL$_2$: Pathfinding via reinforcement and imitation multi-agent learning - lifelong. *IEEE Robot. Autom. Lett.* 6, 2666–2673. doi:10.1109/lra.2021.3062803

Denkena, B., Dittrich, M.-A., Fohlmeister, S., Kemp, D., and Palmer, G. (2021). "Scalable cooperative multi-agent-reinforcement-learning for order-controlled on schedule manufacturing in flexible manufacturing systems," in *Simulation in Produktion und Logistik 2021*, 15.-17, 305.*Erlangen*

Dittrich, M.-A., and Fohlmeister, S. (2020). Cooperative multi-agent system for production control using reinforcement learning. *CIRP Ann.* 69, 389–392. doi:10.1016/j.cirp.2020.04.005

Foerster, J., Assael, I. A., De Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *Adv. Neural Inf. Process. Syst.* 29. doi:10.48550/arXiv.1605.06676

Gabel, T., and Riedmiller, M. (2008). Adaptive reactive job-shop scheduling with reinforcement learning agents. *Int. J. Inf. Technol. Intelligent Comput.* 24, 14–18.

Gankin, D., Mayer, S., Zinn, J., Vogel-Heuser, B., and Endisch, C. (2021). "Modular production control with multi-agent deep q-learning," in 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (Västerås, Sweden: IEEE), 1–8.

Gerpott, F. T., Lang, S., Reggelin, T., Zadek, H., Chaopaisarn, P., and Ramingwong, S. (2022). Integration of the a2c algorithm for production scheduling in a two-stage hybrid flow shop environment. *Procedia Comput. Sci.* 200, 585–594. doi:10.1016/j.procs.2022.01.256

Hong, C., and Lee, T.-E. (2018). "Multi-agent reinforcement learning approach for scheduling cluster tools with condition based chamber cleaning operations," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (Orlando, Florida: IEEE), 885–890.

Ji, H., and Jin, Y. (2022). "Designing self-assembly systems with deep multiagent reinforcement learning," in *Design computing and Cognition'20* (Springer), 667–679.

Johnson, D., Chen, G., and Lu, Y. (2022). Multi-agent reinforcement learning for real-time dynamic production scheduling in a robot assembly cell. *IEEE Robot. Autom. Lett.* 7, 7684–7691. doi:10.1109/lra.2022.3184795

Jung, K., Choi, S., Kulvatunyou, B., Cho, H., and Morris, K. C. (2017). A reference activity model for smart factory design and improvement. *Prod. Plan. control* 28, 108–122. doi:10.1080/09537287.2016.1237686

Karapantelakis, A., and Fersman, E. (2021). Mobile operator collaboration using cooperative multi-agent deep reinforcement learning. *IEEE Internet Things J.*

Kim, Y. G., Lee, S., Son, J., Bae, H., and Do Chung, B. (2020). Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. *J. Manuf. Syst.* 57, 440–450. Elsevier doi:10.1016/j.jmsy.2020.11.004

Lan, X., Qiao, Y., and Lee, B. (2021). "Towards pick and place multi robot coordination using multi-agent deep reinforcement learning," in 2021 7th International Conference on Automation, Robotics and Applications (ICARA) (IEEE), 85–89.

Lau, T. T.-K., and Sengupta, B. (2022). *The multi-agent pickup and delivery problem: Mapf, marl and its warehouse applications. arXiv preprint arXiv: 2203.07092.*

Li, M., Guo, B., Zhang, J., Liu, J., Liu, S., Yu, Z., et al. (2021). "Decentralized multi-agv task allocation based on multi-agent reinforcement learning with information potential field rewards," in 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) (Denver, Colorado: IEEE), 482–489.

Liu, C.-L., Chang, C.-C., and Tseng, C.-J. (2020). Actor-critic deep reinforcement learning for solving job shop scheduling problems. *Ieee Access* 8, 71752–71762. doi:10.1109/access.2020.2987820

Liu, C., Zhu, H., Tang, D., Nie, Q., Zhou, T., Wang, L., et al. (2022). Probing an intelligent predictive maintenance approach with deep learning and augmented reality for machine tools in iot-enabled manufacturing. *Rob. Comput.-Integr. Manuf.* 77, 102357. doi:10.1016/j.rcim.2022.102357

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. (2017). "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 30.

Luo, S., Zhang, L., and Fan, Y. (2021a). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Comput. Industrial Eng.* 159, 107489. doi:10.1016/j.cie.2021.107489

Luo, S., Zhang, L., and Fan, Y. (2021b). Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning. *IEEE Trans. Autom. Sci. Eng.* 19, 3020–3038. doi:10.1109/tase.2021.3104716

Malus, A., Kozjek, D., and Vrabic, R. (2020). Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Ann.* 69, 397–400. doi:10.1016/j.cirp.2020.04.001

Mittal, S., Khan, M. A., Romero, D., and Wuest, T. (2019). Smart manufacturing: Characteristics, technologies and enabling factors. *Proc. Institution Mech. Eng. Part B J. Eng. Manuf.* 233, 1342–1361. doi:10.1177/0954405417736547

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). "Asynchronous methods for deep reinforcement learning," in International conference on machine learning (New York, NY: PMLR), 1928–1937.

Mukhutdinov, D., Filchenkov, A., Shalyto, A., and Vyatkin, V. (2019). Multi-agent deep learning for simultaneous optimization for time and energy in distributed routing system. *Future Gener. Comput. Syst.* 94, 587–600. doi:10.1016/j.future.2018.12.037

Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Trans. Cybern.* 50, 3826–3839. doi:10.1109/tcyb.2020.2977374

Nowé, A., Vrancx, P., and Hauwere, Y.-M. D. (2012). "Game theory and multi-agent reinforcement learning," in *Reinforcement learning* (Springer), 441–470.

Nwakanma, C. I., Islam, F. B., Maharani, M. P., Lee, J.-M., and Kim, D.-S. (2021). Detection and classification of human activity for emergency response in smart factory shop floor. *Appl. Sci.* 11, 3662. doi:10.3390/app11083662

OroojlooyJadid, A., and Hajinezhad, D. (2019). *A review of cooperative multi-agent deep reinforcement learning. arXiv preprint arXiv:1908.03963.*

Ozdemir, R., and Koc, M. (2019). "A quality control application on a smart factory prototype using deep learning methods," in 2019 IEEE 14th international conference on computer sciences and information technologies (CSIT) (Lviv, Ukraine: IEEE) 1, 46–49.

Park, I.-B., Huh, J., Kim, J., and Park, J. (2019). A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. *IEEE Trans. Autom. Sci. Eng.* 17, 1–12. doi:10.1109/tase.2019.2956762

Pham, H. X., La, H. M., Feil-Seifer, D., and Nefian, A. (2018). *Cooperative and distributed reinforcement learning of drones for field coverage. arXiv preprint arXiv: 1803.07250*.

Pol, S., Baer, S., Turner, D., Samsonov, V., and Meisen, T. (2021). "Global reward design for cooperative agents to achieve flexible production control under real-time constraints," in Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021), Setúbal, Portugal, 515–526.

Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., and Wang, L. (2019). Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. *IEEE access* 7, 146264–146272. doi:10.1109/access.2019.2943253

Qu, S., Wang, J., Govil, S., and Leckie, J. O. (2016). Optimized adaptive scheduling of a manufacturing process system with multi-skill workforce and multiple machine types: An ontology-based, multi-agent reinforcement learning approach. *Procedia Cirp* 57, 55–60. doi:10.1016/j.procir.2016.11.011

Qu, S., Wang, J., and Jasperneite, J. (2019). "Dynamic scheduling in modern processing systems using expert-guided distributed reinforcement learning," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (Zaragoza, Spain: IEEE), 459–466.

Samadi, E., Badri, A., and Ebrahimpour, R. (2020). Decentralized multi-agent based energy management of microgrid using reinforcement learning. *Int. J. Electr. Power & Energy Syst.* 122, 106211. doi:10.1016/j.ijepes.2020.106211

Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T. S., Koenig, S., et al. (2019). Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robot. Autom. Lett.* 4, 2378–2385. doi:10.1109/lra.2019.2903261

Shen, G., Ma, R., Tang, Z., and Chang, L. (2021). "A deep reinforcement learning algorithm for warehousing multi-agv path planning," in 2021 International Conference on Networking, Communications and Information Technology (NetCIT) (Manchester, United Kingdom: IEEE), 421–429.

Shi, Z., Xie, Y., Xue, W., Chen, Y., Fu, L., and Xu, X. (2020). Smart factory in industry 4.0. *Syst. Res. Behav. Sci.* 37, 607–617. doi:10.1002/sres.2704

Sjödin, D. R., Parida, V., Leksell, M., and Petrovic, A. (2018). Smart factory implementation and process innovation: A preliminary maturity model for leveraging digitalization in manufacturing moving to smart factories presents specific challenges that can be addressed through a structured approach focused on people, processes, and technologies. *Research-Technol. Manag.* 61, 22–31. doi:10. 1080/08956308.2018.1471277

Su, J., Huang, J., Adams, S., Chang, Q., and Beling, P. A. (2022). Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems. *Expert Syst. Appl.* 192, 116323. doi:10.1016/j.eswa.2021.116323

Sultana, N. N., Meisheri, H., Baniwal, V., Nath, S., Ravindran, B., and Khadilkar, H. (2020). *Reinforcement learning for multi-product multi-node inventory management in supply chains. arXiv preprint arXiv:2006.04037*.

Sutton, R. S., and Barto, A. G. (1998). *Introduction to reinforcement learning*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Advances in neural information processing systems*, 30.

Wang, H.-X., and Yan, H.-S. (2016). An interoperable adaptive scheduling strategy for knowledgeable manufacturing based on smgwq-learning. *J. Intell. Manuf.* 27, 1085–1095. doi:10.1007/s10845-014-0936-1

Wang, X., Wang, H., and Qi, C. (2016). Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. *J. Intell. Manuf.* 27, 325–333. doi:10.1007/s10845-013-0864-5

Wang, J., Qu, S., Wang, J., Leckie, J. O., and Xu, R. (2017). "Real-time decision support with reinforcement learning for dynamic flowshop scheduling," in Smart SysTech 2017; European Conference on Smart Objects, Systems and Technologies (Munich, Germany: VDE), 1–9.

Wang, M., Chen, X., Zhou, J., Jiang, T., and Cai, W. (2018). "Shared cognition based integration dynamic scheduling method," in 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (Shaanxi, China: IEEE), 1438–1442.

Wang, C., Pérez-D'Arpino, C., Xu, D., Fei-Fei, L., Liu, K., and Savarese, S. (2022a). "Co-gail: Learning diverse strategies for human-robot collaboration," in Conference on Robot Learning (London, United Kingdom: PMLR), 1279–1290.

Wang, X., Zhang, L., Lin, T., Zhao, C., Wang, K., and Chen, Z. (2022b). Solving job scheduling problems in a resource preemption environment with multi-agent reinforcement learning. *Rob. Comput. Integr. Manuf.* 77, 102324. doi:10.1016/j. rcim.2022.102324

Wang, Y., Qiu, D., and Strbac, G. (2022c). Multi-agent deep reinforcement learning for resilience-driven routing and scheduling of mobile energy storage systems. *Appl. Energy* 310, 118575. doi:10.1016/j.apenergy.2022.118575

Wang, Y.-F. (2020). Adaptive job shop scheduling strategy based on weighted q-learning algorithm. *J. Intell. Manuf.* 31, 417–432. doi:10.1007/s10845-018-1454-3

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., et al. (2018a). "Deep reinforcement learning for semiconductor production scheduling," in 2018 29th annual SEMI advanced semiconductor manufacturing conference (ASMC) (IEEE), 301–306.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., et al. (2018b). Optimization of global production scheduling with deep reinforcement learning. *Procedia Cirp* 72, 1264–1269. doi:10.1016/j.procir.2018. 03.212

Yu, T., Huang, J., and Chang, Q. (2021). Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning. *J. Manuf. Syst.* 60, 487–499. doi:10.1016/j.jmsy.2021.07.015

Yun, W. J., Park, S., Kim, J., Shin, M., Jung, S., Mohaisen, A., et al. (2022). Cooperative multi-agent deep reinforcement learning for reliable surveillance via autonomous multi-uav control. *IEEE Trans. Ind. Inf.* 18, 7086–7096. doi:10.1109/tii. 2022.3143175

Zhang, C., Odonkor, P., Zheng, S., Khorasgani, H., Serita, S., Gupta, C., et al. (2020a). "Dynamic dispatching for large-scale heterogeneous fleet via multi-agent deep reinforcement learning," in 2020 IEEE International Conference on Big Data (Big Data) (Atlanta, Georgia: IEEE), 1436–1441.

Zhang, Y., Qian, Y., Yao, Y., Hu, H., and Xu, Y. (2020b). "Learning to cooperate: Application of deep reinforcement learning for online agv path finding," in Proceedings of the 19th International Conference on autonomous agents and multiagent systems, Auckland, New Zealand, 2077–2079.

Zhang, K., Yang, Z., and Başar, T. (2021). "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of reinforcement learning and control*, 321–384. doi:10.1007/978-3-030-60990-0_12

Zhang, R., Lv, Q., Li, J., Bao, J., Liu, T., and Liu, S. (2022). A reinforcement learning method for human-robot collaboration in assembly tasks. *Rob. Comput.-Integr. Manuf.* 73, 102227. doi:10.1016/j.rcim.2021.102227

Zhao, J., Zhang, Y., Hu, X., Wang, W., Zhou, W., Hao, J., et al. (2022). *Revisiting qmix: Discriminative credit assignment by gradient entropy regularization. arXiv preprint arXiv:2202.04427*.

Zhiyao, L., and Sartoretti, G. (2020). Deep reinforcement learning based multiagent pathfinding. Tech. rep., Technical Report

Zhou, T., Tang, D., Zhu, H., and Zhang, Z. (2021). Multi-agent reinforcement learning for online scheduling in smart factories. *Robotics Computer-Integrated Manuf.* 72, 102202. doi:10.1016/j.rcim.2021.102202

Zinn, J., Vogel-Heuser, B., and Gruber, M. (2021). Fault-tolerant control of programmable logic controller-based production systems with deep reinforcement learning. *J. Mech. Des.* 143. doi:10.1115/1.4050624

Zong, Z., Zheng, M., Li, Y., and Jin, D. (2022). Mapdp: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. *Proc. AAAI Conf. Artif. Intell.* 36, 9980–9988. doi:10.1609/aaai.v36i9.21236

TYPE Brief Research Report
PUBLISHED 14 December 2022
DOI 10.3389/frobt.2022.1021755

OPEN ACCESS

EDITED BY
Shaoming He,
Beijing Institute of Technology, China

REVIEWED BY
Rocco Furferi,
University of Florence, Italy
Bernd Peukert,
Royal Institute of Technology, Sweden

*CORRESPONDENCE
Thorben Schnellhardt,
✉ thorben.schnellhardt@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems, a section of the journal Frontiers in Robotics and AI

RECEIVED 17 August 2022
ACCEPTED 02 December 2022
PUBLISHED 14 December 2022

CITATION
Schnellhardt T, Hemschik R, Weiß A, Schoesau R, Hellmich A and Ihlenfeldt S (2022), Scalable production of large components by industrial robots and machine tools through segmentation. *Front. Robot. AI* 9:1021755.
doi: 10.3389/frobt.2022.1021755

Scalable production of large components by industrial robots and machine tools through segmentation

Thorben Schnellhardt[1]*, Rico Hemschik[2], Arno Weiß[1], Rene Schoesau[1], Arvid Hellmich[1] and Steffen Ihlenfeldt[1]

[1]Fraunhofer Institute for Machine Tools and Forming Technology IWU, Dresden, Germany,
[2]Fraunhofer Institute for Material and Beam Technology IWS, Dresden, Germany

The production of large components currently requires cost-intensive special machine tools with large workspaces. The corresponding process chains are usually sequential and hard to scale. Furthermore, large components are usually manufactured in small batches; consequently, the planning effort has a significant share in the manufacturing costs. This paper presents a novel approach for manufacturing large components by industrial robots and machine tools through segmented manufacturing. This leads to a decoupling of component size and necessary workspace and enables a new type of flexible and scalable manufacturing system. The presented solution is based on the automatic segmentation of the CAD model of the component into segments, which are provided with predefined connection elements. The proposed segmentation strategy divides the part into segments whose structural design is adapted to the capabilities (workspace, axis configuration, etc.) of the field components available on the shopfloor. The capabilities are provided by specific information models containing a self-description. The process planning step of each segment is automated by utilizing the similarity of the segments and the self-description of the corresponding field component. The result is a transformation of a batch size one production into an automated quasi-serial production of the segments. To generate the final component geometry, the individual segments are mounted and joined by robot-guided Direct Energy Deposition. The final surface finish is achieved by post-processing using a mobile machine tool coupled to the component. The entire approach is demonstrated along the process chain for manufacturing a forming tool.

KEYWORDS

segmented manufacturing, large component, CNC machining, process planning, mobile machine tool, laser metal deposition

1 Introduction

The demand for large and accurately machined parts is rising. Currently, those parts are usually machined on large machine tools. However, large machine tools suffer from some significant shortcomings (Uriarte et al., 2013), i.e. high level of investment, low sustainability due to high energy and material consumption, low productivity due to long machining cycle times and several technical limitations that arise from the large dimension/workspace size (e.g. thermal issues, reduced stiffness). To address these shortcomings various approaches exist in the research field of machine tools. E.g. optimized machine structures to improve the eco-efficiency (Zulaika and Campa, 2009), size-scalable machine tool frames based on polyhedral building blocks that enable reconfigurability (Uhlmann and Peukert, 2019) or mobile machine tool solutions (Neugebauer et al., 2012) that improve the utilization of resources. However, most approaches have not yet found their way into industry or target only a subset of the aforementioned issues. When compared to conventional machine tools, industrial robots have a large workspace that can be expanded further. DeVlieg (2011) and Saund and DeVlieg (2013), for instance, extend a robot with an additional linear axis for the machining of large-scale aluminum aircraft components. Möller et al. (2017) and Susemihl et al. (2017), on the other hand, developed a mobile robot system on an autonomous mobile platform, capable of machining composite (CFRP) aircraft components. An advantage of such mobile robotics solutions is the possibility of scaling and parallelization, which can significantly reduce process times.

Although robots are currently used in the machining of large components, they are significantly less rigid compared to conventional machine tools (static Cartesian stiffness is up to 50 times lower), which reduces machining accuracy, hence they cannot be utilized for all applications (Verl et al., 2019).

To overcome this problem, numerous approaches can be found in the literature, including the design optimization of milling robots (Denkena et al., 2017), various concepts for structural optimization (Tao et al., 2019) or placement optimization (Xue et al., 2022). However, robotic milling in large-scale component manufacturing is currently limited to softer materials such as aluminum, plastic, and composite (Kim et al., 2019). This excludes components with high hardness and accuracy requirements such as forming tools.

To address this problem, we propose a novel and sustainable approach to large component manufacturing that enables robots and regular sized machine tools to manufacture large components (with high hardness and accuracy requirements) within a highly scalable manufacturing system. Thus reducing the need for expensive and resource intensive large machine tools. The approach is based on segmented manufacturing and subsequent joining of the large component. Key to the approach is the segmentation of the component into segments that are tailored to the capabilities of the available shop floor entities, i.e., machine tools and robots equipped with different end effectors and tools.

So far, approaches towards segmented manufacturing are sparse in the literature. Ley et al. (2018) present an approach for hybrid-optimized manufacturing of large components by segmenting the component into subcomponents. However, their focus is primarily on the combination of additive and conventional manufacturing and less on automation and productivity. Further examples of segmented manufacturing can be found in the area of toolmaking. These include mold inserts in forming tools (Cao et al., 2019) or multi-part injection molds (Stoyan and Chen, 2010). However, their focus is primarily on the functionalization of the component rather than the actual manufacturing process.

The main goal of this paper is to present the basic principles of the proposed approach. For this purpose, a general overview of the underlying process chain is given in Section 2.1. Section 2.2 presents the conceptual architecture as an essential enabler for the modularity and scalability of the approach. Section 2.3 describes the segmentation procedure. Joining and finishing of the segments are discussed in Section 2.4 and Section 2.5. Finally, the basic manufacturing approach is demonstrated and validated on a downscaled forming tool.

2 Methods

2.1 General

This section provides an overview of the proposed approach. Figure 1 depicts the basic process steps of the approach in terms of the involved modules. Modules (see Section 2.2) have a specific task in the manufacturing system and represent the necessary software and hardware entities (assets). Modules are modeled by lightweight information models, which they expose *via* uniform communication interfaces. These information models describe the capabilities, states, and services of the assets. The considered manufacturing system consists of a set of loosely coupled machine tools and robots, equipped with different end effectors and tools.

Starting point of the manufacturing process is the CAD model and a Finite Element Analysis (FEA) of the large component. The *SegmentationModule* applies a predefined, application specific segmentation strategy and splits the CAD model into a combination of smaller, less complex segments. The segments are split in a way, that optimizes a given target quantity, e.g., time, cost or utilization rate. Therefore, the *SegmentationModule* provides an interface to an *OptimizationModule*, which determines a combination of segments that suits the capabilities of the existing assets and maps them accordingly. The CAD models of the individual segments are extended with connection elements (e.g. screw

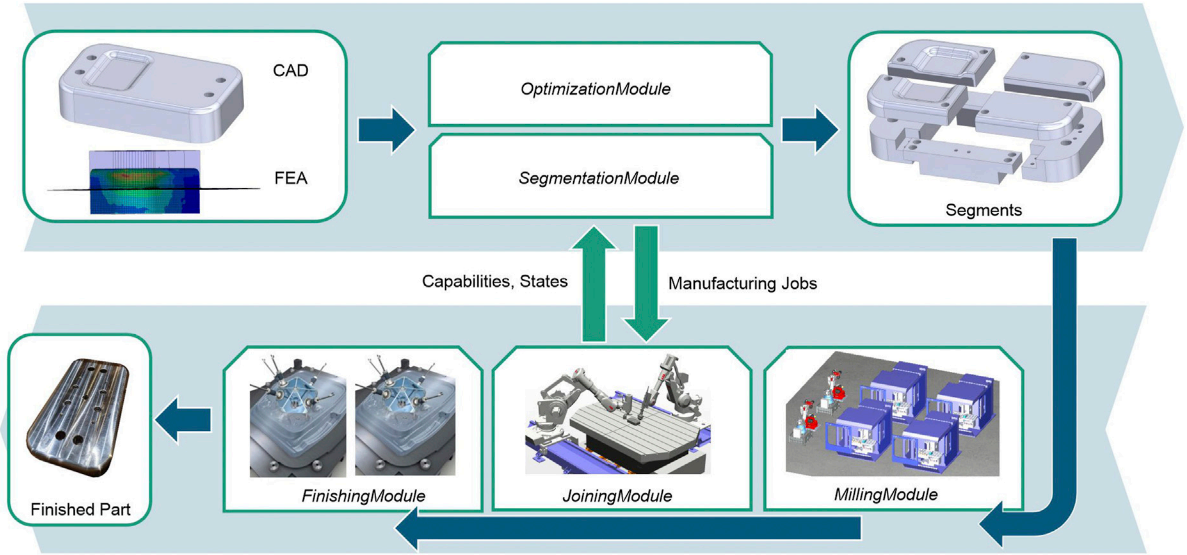


FIGURE 1
Overview of the approach and the underlying process chain.

connections) for assembly. The connection elements are part of the segmentation strategy. Afterwards, NC programs are created for the individual segments using a feature-based approach with predefined machining operations. The NC programs are passed to the corresponding *MillingModules* and are manufactured. *MillingModules* represent assets such as conventional CNC machines or milling robot cells.

The manufactured segments are assembled and thermally joined by the *JoiningModule* to bridge the segment gaps induced by segmentation. Thus, a robot-guided Direct Energy Deposition (DPD) process is used at this point. The aim is to minimize component distortion in order to reduce the amount of post-processing required during the final stage of the process. After the joining, the component is measured *in situ* by the robot to determine the finishing effort. The measurement data is finally used for the finishing process, which is performed by mobile machine tools (*FinishingModules*) that are temporarily coupled to the component.

2.2 Architecture

Executing the previously described process is primarily a technological task with high requirements to process quality. However, the targets motivating this novel approach (scalability, productivity) can only be met with comprehensive automated planning and execution. A particular challenge arises from the inherent heterogeneity of production assets involved. They differ with regard to their scope (i.e., segmentation, production, joining), with regard to their origin (vendors, PLCs) and with regard to their age. All these factors influence the interfaces available to upper-level process coordination in the *OptimizationModule*. An architecture suitable for the task must handle this complexity, thus automate the material and information flow between production assets.

To meet these requirements, we followed the integration guide of the SWAP-IT architecture (Lünsch et al., 2022). The SWAP-IT architecture is composed of autonomous *Modules*, each of which specifies its characteristic *Services* to peers. A *Service* signifies the *Module's* potential to trigger an executable process, similar to a *Skill* in literature (Köcher et al., 2020). The *Module Services* are implemented by its member *Agents*. Each *Agent* encapsulates an asset (logical or physical) and adapts the proprietary communication interface to a harmonized representation in the network. All *Agents* register with a central entity that provides transparency on available *Agents* and their *Services* in the network. This highly decentralized architecture enables flexible routing and reallocation of resources according to the requirements of the production order and current availability. The communication backbone for interaction between and within modules is OPC UA as it supports the reuse of common semantic information models that are tightly coupled to communication mechanisms for machines and computers alike. These information models are used to represent the *Module*'s specific *Capabilities* and *States*.

As all steps mentioned in Section 2.1 must be represented on the *Module*-level, *Modules* for segmentation, milling, joining and finishing were designed and implemented on *Agents*. The

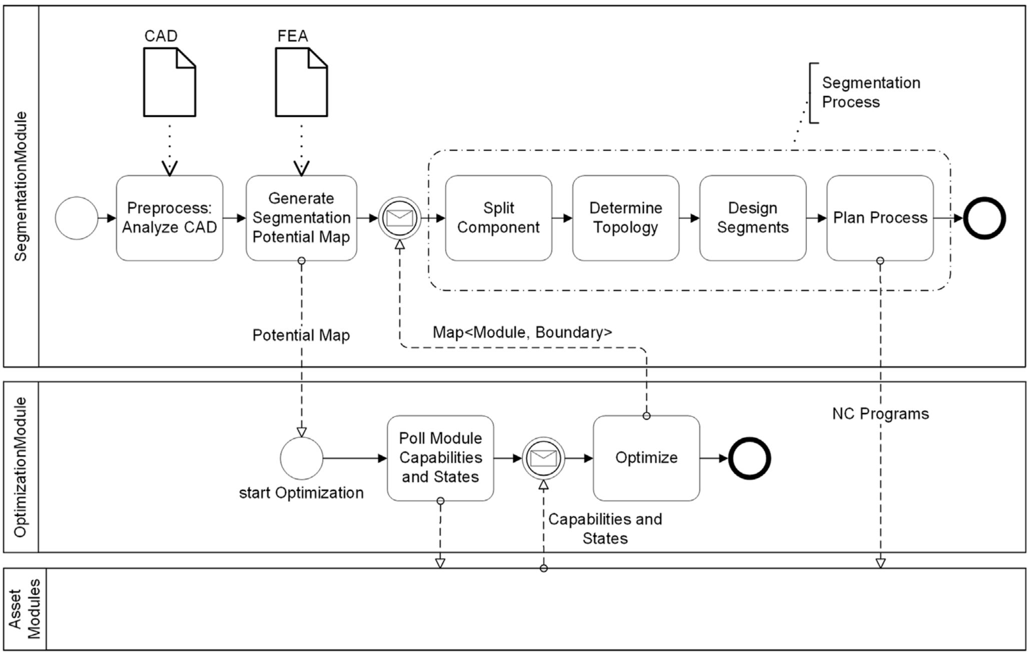


FIGURE 2
Workflow of the segmentation procedure.

following chapters provide an in-depth description of their functionality.

2.3 Segmentation

The proposed approach increases the number of parts to be manufactured compared to conventional large component production. Accordingly, the design and planning effort also increases, which in turn has a negative impact on production costs and duration and thus eventually also has a negative effect on overall productivity. Moreover, manual segmentation is prone to errors, as the complexity and dependencies within the assembly scale with the number of segments. Therefore, it is crucial for the applicability of the approach to automate the segmentation.

In our case, the *SegmentationModule* handles this task, i.e., the automatic execution of the CAD-CAM chain along the segmentation from the CAD model of the large component, through the design of the individual segments, to the process planning and the NC programs. Thereby, the characteristic properties of the segmentation (similarity of the segments, lower complexity compared to the overall component) are utilized. In addition, the module provides an interface to the *OptimizationModule*, which determines suitable segment combinations. Figure 2 summarizes the process steps involved in the segmentation procedure, which are described in detail in the following.

2.3.1 Preprocess – analyze CAD

In the beginning, an initial part analysis is performed to derive the geometric constraints for the segmentation. For this purpose, an automatic design feature analysis of the CAD model is carried out. The goal of this analysis is to identify those areas of the part that must not be segmented, e.g., connection holes. In the case of a native CAD format, the vendor specific design features can usually be utilized for this task, e.g. we use the design features that SolidWorks provides *via* their API. Alternatively, a surface analysis of the B-Rep model must be conducted using one of the various feature recognition methods (e.g. (Han et al., 2000), (Zhang et al., 2017), (Zhang et al., 2018)). In addition, the part is searched for areas that require 5-axis machining. For this purpose, the surfaces of the CAD model are grouped into standard 2.5D feature surfaces and freeform surfaces according to their B-Rep model.

2.3.2 Generate segmentation potential map

For the optimization phase, the constraints of the segmentation are merged and brought into a uniform format in form of a segmentation potential map. Therefore, a mapping is used which assigns numerical values to the regions of the large component, which describe the segmentation potential of the respective region. To generate these values, the absolute values of the FEA stress data are utilized. The FEA denotes the expected stress during the usage of the final product. To integrate the geometric constraints (from the preprocess stage), the positional data of the geometric constraints are amplified with high values and overlaid with the absolute stress values. This results in a mapping where low absolute values describe a high segmentation potential and high values describe a poor segmentation potential.

2.3.3 Optimization

Input of the *OptimizationModule* is the potential map generated in the previous step. Furthermore, the *Module* uses the *Capabilities* and *States* (workspace, machine costs, axis configuration, availability, etc.) of the *MillingModules.* The goal of the optimizer is to segment the part and map the segments to the available assets in such a way that an optimal production time, cost or utilization rate is achieved. At the same time, the constraints, encoded within the potential map, have to be satisfied. I.e. the boundaries of the segments should only intersect low valued regions of the potential map. The result of the *OptimizationModule* is a segmentation recommendation in form of an assignment of segment boundaries (Axis-Aligned Bounding Boxes) and *MillingModules.*

2.3.4 Split component

In this step, the actual CAD model is finally split into the individual raw segment pieces according to the segmentation recommendation of the *OptimizationModule.* Furthermore, the classification of the segments into specific types is performed in this step, if required by the applied segmentation strategy.

2.3.5 Determine topology

Next, segment topology is determined, i.e., the spatial relationship of the individual segment parts and their geometric entities (faces and edges) to each other. This is necessary for the extension of the segments with the connection elements for the assembly. Furthermore, the topology information is used to apply the tolerances to the segments. Based on the assumption that the intersections between the segments are planar surfaces, a collision check using Oriented Bounding Boxes in 3D/2D (body to body and face to face) is applied for the topology determination.

2.3.6 Design segments

According to the selected segmentation strategy and the face/body relationships (topology), the connection features and tolerances are constructed automatically onto the segments. To simplify CAM planning, only subtractive design features are used during this stage.

2.3.7 Plan process

Based on the data collected during the previous process steps (feature recognition, topology, tolerances) feature-based process planning is performed. Therefore, a matching between predefined machining operations, tools and machining features (based on ISO 14649-10, 2004) is carried out. The basis for this is the similarity of the segments and their reduced complexity compared to the overall component. Segments consist of the planar cut surfaces caused by segmentation, a fraction of the surface of the original component and a number of predefined connection elements. This simplifies the accessibility and clamping of the segments significantly and thus eases the automation potential of the processes. For the generation of the machine-specific NC code, the *MillingModule's* information models and the corresponding postprocessor are utilized. Finally, the execution of the *MillingModule's Services* are triggered on the field device using the generated NC-Programs.

2.4 Joining

After the individual segments have been manufactured, they are joined by using industrial robots. Here, a DED process is used that applies a powdery filler material to the component's surfaces with the help of a laser and thus fuses the segments metallurgically. For this purpose, a double robot system is used with two processing heads on which OTS-2 laser optics are mounted. Two COAX Powerline powder nozzles are utilized to inject the powder material into the process zone, each of which are integrated into a COAXshield system (Kolsch et al., 2020) for better shielding of the process. This makes it possible to carry out the welding process within a local shielding gas without having to flood the entire installation space of the cell, which is crucial when processing large components.

The robot cell used in this research offers a usable workspace of 3 m length, 3 m width and 1 m height. Both robot systems are mounted onto linear axes and are able to operate synchronously. In addition, the cell offers a rotary table. The robot's traversing strategy for the welding process must be automated for the most part to ensure productivity. As the positions of the weld seams are already known from the preceding segment design process of the *SegmentationModule*, the necessary welding trajectories of the robot can be derived directly.

For the alignment of the segments onto the machine table, it is necessary to induce a workflow that avoids clamping errors, since they have a major influence on the final geometry and accuracy of the component. Therefore, two approaches were established. On the one hand, when subdividing the large component, it is important to ensure that the segments produced can be mounted and clamped easily and form-fittingly. On the other hand, it is important to pre-measure the entire geometry of the clamped large component in order to enable precise alignment of the coordinate system. For both measures, it is advisable to provide mounting holes whose arrangement corresponds to the standardized clamping grooves on machine tables. With the aid of these holes, the segments can be positioned quickly and precisely, and a subsequent 3D scan or measurement with a probing device, allows the workpiece coordinate system to be set up precisely.

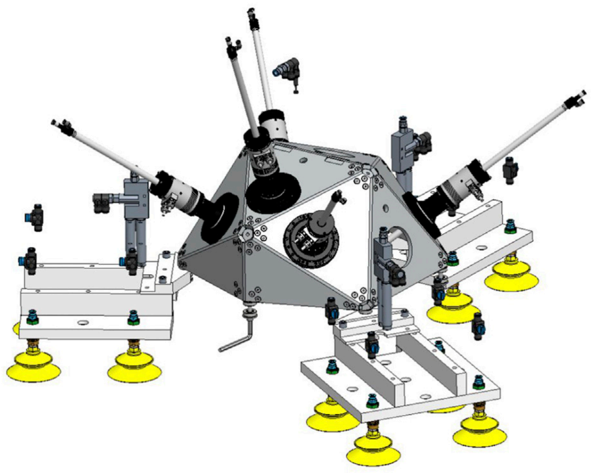


FIGURE 3
Mobile machine tool with parallel kinematic design equipped with a pneumatic coupling solution.

After the joining process, a 3D measurement of the large component is conducted, in order to measure both the distortion and the geometries of the applied weld seams. The measurement is carried out automatically by the manufacturing robot. For this purpose, a MICRO-EPSILON scanControl 2900–100/BL line scanner is mounted onto the robot. The robot automatically scans the entire component surface with the help of a measuring routine and prepares the data as a STL file. Since the applied line scanner only has a relatively narrow measuring range of approx. 8 cm, the component is scanned several times with overlapping strips. The individual scans are automatically combined by using feature recognition, in order to map the entire surface of the component. For the actual measurement process, it is important to move as few robot axes as possible in order to avoid error entry due to tolerances. Therefore, only the linear axis on which the robot arm is mounted on or the rotary table that holds the component is moved for the measurement.

After a nominal-actual comparison with the existing CAD data of the component, it is possible to detect any distortion and the weld seam geometry for the entire component. Therefore, it is possible to reduce the subsequent finish machining to areas that deviate from the nominal geometry and thus increase the efficiency of the entire production process.

2.5 Finishing

The finish machining of the areas deviating from the nominal geometry is carried out with a small mobile machine tool with parallel kinematic design (Georgi et al., 2018) which is temporarily coupled to the workpiece. The main advantage of this mobile machine tool is the possibility of flexible and highly dynamic local machining of large components. Due to the direct positioning of the small machine on the workpiece, the dependency of machine size to workpiece dimension is resolved and downsizing of the production equipment is possible, which in turn increases the transportability and manageability of the production systems as well as the efficiency of the overall system. Due to the necessity that all movement axes have to be on the tool side, the parallel kinematic machine concept proves to be advantageous due to the low moving masses and enables movement with five degrees of freedom.

Solutions for temporarily coupling the mobile machine to the large component are subject to certain design constraints. For example, magnetic clamping systems reach their limits due to their principle when processing aluminium or CFRP components. Thus, a pneumatic coupling solution (Figure 3) consisting of twelve bellows suction cups with upstream rubber joints was developed to compensate for unevenness and displacement due to concave or convex workpiece contours.

The workflow for finishing the areas deviating from the nominal geometry is based on the data from the scan at the end of the joining process. The miniaturized machine tool is positioned over the areas in question. The coordinate systems of the machine and the workpiece are precisely aligned with each other, and the deviation areas (e.g. weld seams, segment deviations) are post-processed and finished.

3 Results

The validation of the proposed approach has been implemented and applied on the manufacturing process of a downscaled forming tool. Thereby, the die punch of a rectangular deep drawing tool (347.36 mm × 197.36 mm × 76 mm) served as the test object. The goal of the validation was to investigate the basic applicability of the approach for highly loaded (large) components. Figure 4 shows the individual steps of the validation.

To determine the process-related stresses, a forming simulation was performed using LS-Dyna (Figure 4A). In order to calculate the stress distribution of the die punch, the simulation was carried out with an elastically modeled die. Therefore the elastic tool method of (Haufe et al., 2008) was applied.

The selected segmentation strategy involves the division of the component into shell and core segments (Figure 4B). Core segments are screwed together to form a framework that provides the basic component stability. Shell segments contain the functional surfaces of the component. These segments are attached to the framework of core segments using screw connections. To control the tolerance chain, the shell segments are provided with a clearance fit. The minimal gap between the segments is later closed through the joining

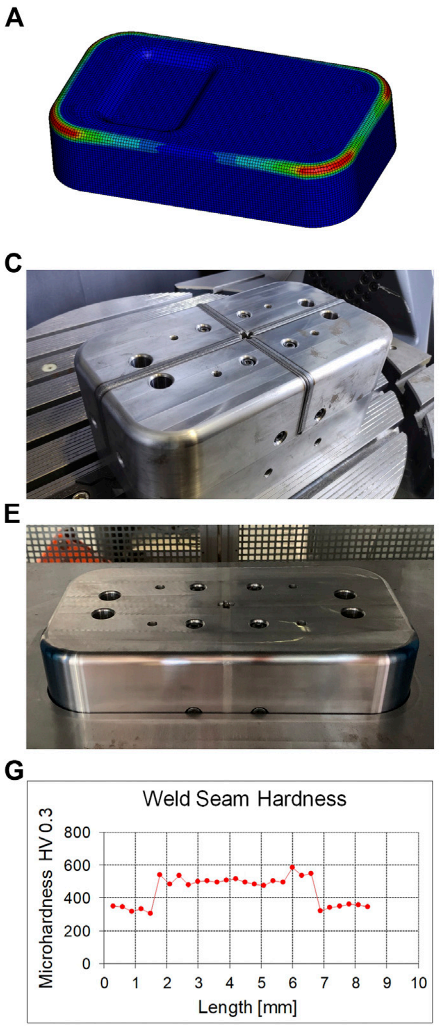


FIGURE 4
Segmented manufacturing of a deep drawing stamp: **(A)** FEA of the deep drawing process, **(B)** segmented part that consist out of (inner) core- and (outer) shell-segments assembled *via* screw connections, **(C)** assembled and welded stamp, **(D)** distortion of the stamp after the welding process, **(E)** finished stamp, **(F)** marks from the weld seams on the sheet after deep drawing, **(G)** hardness measurement of the seam and the heat affected zone.

process. The segmentation strategy was implemented in the CAD/CAM part of the *SegmentationModule* by using the SolidWorks API.

Toolox 33 prehardened steel was selected as the workpiece material for the punch, in order to avoid a subsequent hardening process in the assembled and joined condition. In a preliminary material characterization study, 1.4404/316L (CrNiMo) and 1.4057/431 (CrNiFe) were analyzed to find a suitable filler material for the DPD process. Due to lower hardening around the heat input zone and the weld seam, 1.4404/316L (CrNiMo) was chosen as filler material for joining the punch. The chosen weld depth was 5.5 mm. The welding was carried out with the robot cell described in Section 3.3. The result is depicted in Figure 4C.

To determine the component distortion, the component was measured optically by a laser line scanner (see Section 2.3) before and after the welding process. This resulted in a distortion of +0.3 mm at the outer edges of the component and a distortion of -0.3 mm in the center of the component (Figure 4D). In addition, a hardness test was carried out in the areas of the heat input zone and the seam. This resulted in a hardening of approximately 200 HV as shown in Figure 4G.

Based on the measurement data, post-processing was carried out to generate the final contour and surface quality (Figure 4E). Therefore, the mobile machine tool was manually aligned on the workpiece in five different clamping setups in order to machine the entire surface. To investigate the functional capability and performance of the segmented die, it was tested in a deep-drawing test series. The sheet material used was DC01 with a thickness of 1 mm. The segmented die withstood the process load and delivered comparable production results to a monoblock die of the same type. As can be seen in Figure 4F, however, the forming process applied light marks to those areas that were in contact with the post-processed weld seams. We attributed these marks to the hardening of the seams that results from the welding process.

4 Discussion

A new approach to manufacturing large components by regular-sized robots and machine tools was presented. The approach is based on dividing the large component into segments tailored to the capabilities of the available assets. The segments are joined and post-processed with a mobile machine tool to obtain the final contour. The approach implements a scalable architecture that enables massive parallelization in large-scale component manufacturing.

The associated process chain was validated on a minimal example (forming tool) yielding comparable results to a conventional part. We conclude that the approach is in principle suitable and can increase sustainability and productivity in large component manufacturing due to its scalability. In addition, the approach opens up the possibility for distributed manufacturing at different locations. Furthermore, it mitigates availability problems and reduces the need for cost-intensive special purpose large machine tools in exchange for flexible industrial robots.

A downside is the increased number of necessary manufacturing processes in comparison to conventional process chains. This induces additional complexity. Another shortcoming is the need for an application-specific segmentation strategy, which must be investigated and developed. A general transferability is usually not given and must be examined in each individual case. In the forming tool domain, the approach is currently only suitable for components with limited surface finish requirements, due to the emerging marks on the sheet.

Future work will focus on the implementation of the entire method for a more complex use case where the advantages of the method regarding productivity and scalability can be utilized. In doing so, the potential of the method in its entirety will be investigated. Furthermore, technological fine-tuning is necessary, mainly in the area of the joining and finishing process. Tasks here are the homogenization of the hardening process, the reduction of distortion to reduce the necessary finishing effort and the (semi-)automation of the positioning and alignment of the mobile machine tool.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AH contributed to conception and design of the research project. TS, AH, AW, RH, and RS contributed to the conception of the proposed method. TS, RH, and RS performed the proof of concept and the experiments. TS wrote the first draft of the manuscript. TS, RH, AW, and RS wrote sections of the manuscript. TS, RH, AW, and RS contributed to manuscript revision, read, and approved the submitted version. SI supervised the project.

Funding

This work has been funded by the Fraunhofer Lighthouse project SWAP.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Cao, J., Brinksmeier, E., Fu, M., Gao, R. X., Liang, B., Merklein, M., et al. (2019). Manufacturing of advanced smart tooling for metal forming. *CIRP Ann.* 68, 605–628. doi:10.1016/j.cirp.2019.05.001

Denkena, B., Bergmann, B., and Lepper, T. (2017). Design and optimization of a machining robot. *Procedia Manuf.* 14, 89–96. doi:10.1016/j.promfg.2017.11.010

DeVlieg, R. (2011). High-accuracy robotic drilling/milling of 737 inboard flaps. *SAE Int. J. Aerosp.* 4, 1373–1379. doi:10.4271/2011-01-2733

Georgi, O., Rentzsch, H., and Blau, P. (2018). "Miniaturized parallel kinematic machine tool for the machining of small workpieces," in Proceedings of the 18th International Conference of the European Society for Precision Engineering and

Nanotechnology, June 4th-8th June 2018Venice, IT. D. Billington (Bedford, UK: euspen).

Han, J., Pratt, M., and Regli, W. C. (2000). Manufacturing feature recognition from solid models: A status report. *IEEE Trans. Robot. Autom.* 16, 782–796. doi:10.1109/70.897789

Haufe, A., Roll, K., and Bogon, P. (2008). "Sheet metal forming simulation with elastic tools in ls-dyna," in Numisheet 2008, Interlaken, Switzerland, September 1 - 5, 2008.

ISO 14649-10 (2004). Industrial automation systems and integration — physical device control — data model for computerized numerical controllers — Part 10: General process data. Available at: https://www.iso.org/standard/40895.html.

Kim, S. H., Nam, E., Ha, T., Hwang, S.-H., Lee, J. H., Park, S.-H., et al. (2019). Robotic machining: A review of recent progress. *Int. J. Precis. Eng. Manuf.* 20, 1629–1642. doi:10.1007/s12541-019-00187-w

Köcher, A., Hildebrandt, C., Vieira da Silva, L. M., and Fay, A. (2020). "A formal capability and Skill model for use in plug and produce scenarios," in 2020 IEEE 25th International Conference on Emerging Technologies and Factory Automation (ETFA): Technical University of Vienna, Vienna, Austria, 08-011 September 2020. proceedings (Piscataway, NJ: IEEE), 1663–1670.

Kolsch, N., Seidel, A., Finaske, T., Brueckner, F., Gumpinger, J., Bavdaz, M., et al. (2020). Novel local shielding approach for the laser welding based additive manufacturing of large structural space components from titanium. *J. Laser Appl.* 32, 022075. doi:10.2351/7.0000114

Ley, M., Buschhorn, N., Stephan, N., Teutsch, R., Deschner, C., and Bleckmann, M. (2018). "Hybrid-optimized manufacturing of load-bearing components by combining of conventional and additive manufacturing processes," in *Commercial vehicle technology 2018: Proceedings of the 5th commercial vehicle Technology symposium (CVT 2018)*. K. Berns, K. Dressler, P. Fleischmann, D. Görges, R. Kalmar, B. Sauer, et al. (Wiesbaden: Springer Vieweg), 214–232.

Lünsch, D., Detzner, P., Ebner, A., and Kerner, S. (2022). "Swapit: A lightweight, modular industry 4.0 architecture," in 2022 IEEE 18th International Conference on Automation Science and Engineering: CASE 2022, Mexico City, Mexico, 20-24 Aug. 2022. X. Li, [in press].

Möller, C., Schmidt, H. C., Koch, P., Böhlmann, C., Kothe, S.-M., Wollnack, J., et al. (2017). Machining of large scaled CFRP-Parts with mobile CNC-based robotic system in aerospace industry. *Procedia Manuf.* 14, 17–29. doi:10.1016/j.promfg.2017.11.003

Neugebauer, R., Priber, U., Rentzsch, H., Ihlenfeldt, S., and Hoffmann, D. (2012). "Mobile systems for machining large work pieces," in *Enabling manufacturing competitiveness and economic sustainability*. Editor H. A. ElMaraghy (Berlin, Heidelberg: Springer Berlin Heidelberg), 135–140.

Saund, B., and DeVlieg, R. (2013). High accuracy articulated robots with CNC control systems. *SAE Int. J. Aerosp.* 6, 780–784. doi:10.4271/2013-01-2292

Stoyan, S., and Chen, Y. (2010). "Multi-piece mold design based on linear mixed-integer program toward guaranteed optimality," in 2010 International Conference on Manufacturing Automation, Hong Kong, China, 3-15 December 2010. 112–119.

Susemihl, H., Brillinger, C., Stürmer, S. P., Hansen, S., Boehlmann, C., Kothe, S., et al. (2017). Referencing strategies for high accuracy machining of large aircraft components with mobile robotic systems. SAE Technical Paper Series. (Warrendale, PA, United States: SAE International400 Commonwealth Drive).

Tao, B., Zhao, X., and Ding, H. (2019). Mobile-robotic machining for large complex components: A review study. *Sci. China Technol. Sci.* 62, 1388–1400. doi:10.1007/s11431-019-9510-1

Uhlmann, E., and Peukert, B. (2019). Reconfiguring machine tool behavior via smart building block systems. *Procedia Manuf.* 28, 127–134. doi:10.1016/j.promfg.2018.12.021

Uriarte, L., Zatarain, M., Axinte, D., Yagüe-Fabra, J., Ihlenfeldt, S., Eguia, J., et al. (2013). Machine tools for large parts. *CIRP Ann.* 62, 731–750. doi:10.1016/j.cirp.2013.05.009

Verl, A., Valente, A., Melkote, S., Brecher, C., Ozturk, E., and Tunc, L. T. (2019). Robots in machining. *CIRP Ann.* 68, 799–822. doi:10.1016/j.cirp.2019.05.009

Xue, Y., Sun, Z., Liu, S., Gao, D., and Xu, Z. (2022). Stiffness-oriented placement optimization of machining robots for large component flexible manufacturing system. *Machines* 10, 389. doi:10.3390/machines10050389

Zhang, Y., Luo, X., Zhang, B., and Zhang, S. (2017). Semantic approach to the automatic recognition of machining features. *Int. J. Adv. Manuf. Technol.* 89, 417–437. doi:10.1007/s00170-016-9056-8

Zhang, Z., Jaiswal, P., and Rai, R. (2018). FeatureNet: Machining feature recognition based on 3D convolution neural network. *Computer-Aided Des.* 101, 12–22. doi:10.1016/j.cad.2018.03.006

Zulaika, J., and Campa, F. J. (2009). "New concepts for structural components," in *Machine tools for high performance machining*. Editors L. N. López de Lacalle and A. Lamikiz (London: Springer London), 47–73.

frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 12 January 2023
DOI 10.3389/frobt.2022.1030668

A data-driven approach for motion planning of industrial robots controlled by high-level motion commands

Shuxiao Hou*, Mohamad Bdiwi, Aquib Rashid, Sebastian Krusche and Steffen Ihlenfeldt

Fraunhofer Institute for Machine Tools and Forming Technology (Fraunhofer IWU), Chemnitz, Germany

Check for updates

OPEN ACCESS

EDITED BY
Jose Luis Sanchez-Lopez,
University of Luxembourg, Luxembourg

REVIEWED BY
Omid Elhaki,
Islamic Azad University of Najafabad, Iran
Hongchao Ji,
North China University of Science and Technology, China

*CORRESPONDENCE
Shuxiao Hou,
shuxiao.hou@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 29 August 2022
ACCEPTED 27 December 2022
PUBLISHED 12 January 2023

CITATION
Hou S, Bdiwi M, Rashid A, Krusche S and Ihlenfeldt S (2023), A data-driven approach for motion planning of industrial robots controlled by high-level motion commands.
Front. Robot. AI 9:1030668.
doi: 10.3389/frobt.2022.1030668

Most motion planners generate trajectories as low-level control inputs, such as joint torque or interpolation of joint angles, which cannot be deployed directly in most industrial robot control systems. Some industrial robot systems provide interfaces to execute planned trajectories by an additional control loop with low-level control inputs. However, there is a geometric and temporal deviation between the executed and the planned motions due to the inaccurate estimation of the inaccessible robot dynamic behavior and controller parameters in the planning phase. This deviation can lead to collisions or dangerous situations, especially in heavy-duty industrial robot applications where high-speed and long-distance motions are widely used. When deploying the planned robot motion, the actual robot motion needs to be iteratively checked and adjusted to avoid collisions caused by the deviation between the planned and the executed motions. This process takes a lot of time and engineering effort. Therefore, the state-of-the-art methods no longer meet the needs of today's agile manufacturing for robotic systems that should rapidly plan and deploy new robot motions for different tasks. We present a data-driven motion planning approach using a neural network structure to simultaneously learn high-level motion commands and robot dynamics from acquired realistic collision-free trajectories. The trained neural network can generate trajectory in the form of high-level commands, such as Point-to-Point and Linear motion commands, which can be executed directly by the robot control system. The result carried out in various experimental scenarios has shown that the geometric and temporal deviation between the executed and the planned motions by the proposed approach has been significantly reduced, even if without access to the "black box" parameters of the robot. Furthermore, the proposed approach can generate new collision-free trajectories up to 10 times faster than benchmark motion planners.

KEYWORDS

robot motion planning, data driven robot learning, neural network, industrial robot, robot simulation

1 Introduction

Motion Planning is one of the fundamental problems in robotics fields. For decades numerous methods have been proposed for this task by leveraging two common techniques: Optimization-based and heuristic search-based techniques. The trajectories generated by both motion planning paradigms usually include a large number of *via* points (Figure 1A) and require post-processing to deploy to industrial robots.

The trajectories of the industrial robot are typically programmed in the native language of the robot manufacturer. These programming languages pre-define a set of high-level motion commands. The typically high-level motion commands are Point-to-Point, Linear and Cycle motion. The robot control system provided by the robot manufacturer has its own interpolation algorithm and control loop to execute the programmed motion. These control parameters are finely tuned by the robot manufacturer according to the dynamic behavior of each robot and they are usually inaccessible for the user.

There are two ways to plan and deploy robot motion on most control systems of industrial robots.

1.1 Planning and deploying robot motion with high-level motion commands

For some robot systems, the user can only use the pre-defined high-level motion commands and adapt their parameters to program the desired robot motions, such as programming the start and goal configuration of Point-to-Point motion. In this case, most methods use random shortcuts to reduce the amounts of *via* points. For example (Hauser and Ng-Thow-Hing, 2010), uses various interpolation algorithms, such as parabola and linear interpolation, to directly connect two *via* points on the trajectory. If the direct connection is collision-free, the redundant *via* points can be eliminated (Figure 1C). Since some parameters of the robot are inaccessible, such as dynamic behavior and control parameters, these interpolation algorithms usually use estimated values to interpolate the robot's motion. Then the post-processed trajectory should be converted to pre-defined high-level motion commands and imported into robot control systems (Figure 2B) in the offline phase. In the online phase, the robot control system provided by the robot manufacturer executes the motion commands. The robot control system uses the interpolation algorithm and control parameters implemented and fine-tuned by the robot manufacturer, which differ from the estimated value used in the offline phase. It may result in a geometric and temporal deviation between the executed and the planned motions. The geometric deviation may cause a collision between the robot and static environments. For example, Figure 1 shows the trajectory planned by the interpolation algorithm used by the shortcut method during offline post-processing, and Figure 2 shows the actual robot motion executed by a real robot control system with the interpolation algorithm implemented by robot manufacturers.

1.2 Planning and deploying robot motion with low-level motion commands

Some robot control systems with an additional communication interface allow an additional control loop to command the robots with low-level control inputs in real time, such as position, the velocity of robot joints (yellow arrows in Figure 2A). Most state-of-the-art planners interpolate the motion between the *via* points to low-level control inputs in the offline phase (Figure 1B) and use an additional controller to execute the interpolated trajectories (Figure 2A). (Elhaki and Shojaei, 2022; Rahali et al., 2022; Tan et al., 2023) use various control algorithms in the online phase to minimize the deviation between the executed and the planned motion. However, in heavy-duty industrial robot applications that widely use high-speed and long-distance motions, the deviation becomes significant. For example, in the motion planning framework MoveIt (Chitta et al., 2012), the user should define the maximum jerk and acceleration of joints to interpolate the planned motion in the offline phase. In the online phase, the additional controller tracks the planned motion in real time. If the actual maximum acceleration of the joint during the execution can not reach the values defined by the user, the executed robot motion is slower than planned. This temporal deviation may lead to a collision between the robot and dynamic obstacles such as other robots. For example, in some multi-robot system, the planner schedule multiple robots to pass through a shared area at different timesteps. A robot may collide with others when it enters the shared area earlier or later than planned.

1.3 Contributions

The robot motions planned in the described two ways above should be verified in deploying phase to check whether the geometric and temporal deviation between planning and executing of robot motion results in a collision. When the deviation leads to a collision, the actual robot motion must be adjusted and verified again. This process usually iterates manually many times, thus increasing the effort of deploying robot motion.

Most state-of-the-art collision-free motion planning methods focus on improving the performance of motion planning algorithms in the offline phase, such as computation time and success rate of collision avoidance. Today's agile manufacturing systems require not only automatic robot motion planning but also rapid deployment of robot motions. Therefore, more research is still needed to bridge the gap between offline planning and the rapid deployment of robot motions for reliable online execution. Therefore, we proposed a data-driven motion planning approach that considers deploying and deploying the planned motion already in the offline planning phase. The proposed approach overcomes the problems mentioned above:

(1) The proposed approach uses a neural network structure to simultaneously learn high-level commands and robot dynamics from acquired realistic collision-free trajectories. In the offline planning phase, the trained neural network structure can generate collision-free trajectory as high-level motion commands, such as long-distance, high-speed Point-to-Point and Linear motion. These motion commands can be converted as manufacture-specifical robot language and directly imported into any robot control system (Figure 2C). Because the robot control system can execute these motion commands, the proposed approach does not need an additional control loop to control robot motions in real time and constructs a simpler control architecture. Furthermore, the robot manufacturers tuned the control algorithm of their robot control systems by fully accessing the robot parameters. Therefore, the proposed approach achieves a more stable control structure than the methods described in Figure 2A.

(2) The neural network learns realistic robot dynamics and motion interpolation from actual robot motion execution and uses them to accurately calculate the actual robot motions executed by the robot control system. For example, at each search step in the offline planning phase, the proposed approach use learned robot

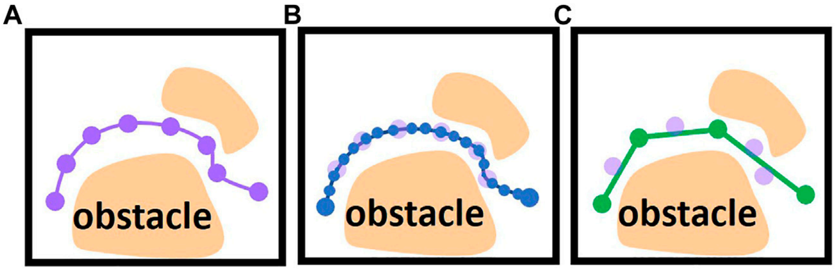


FIGURE 1
(A) Generated trajectory with a large number of *via* points (violet dots). **(B)** An interpolated trajectory interpolated as low-level control inputs (blue dots). **(C)** post-processed trajectory using shortcuts (green dots are reduced *via* points after shortcuts).

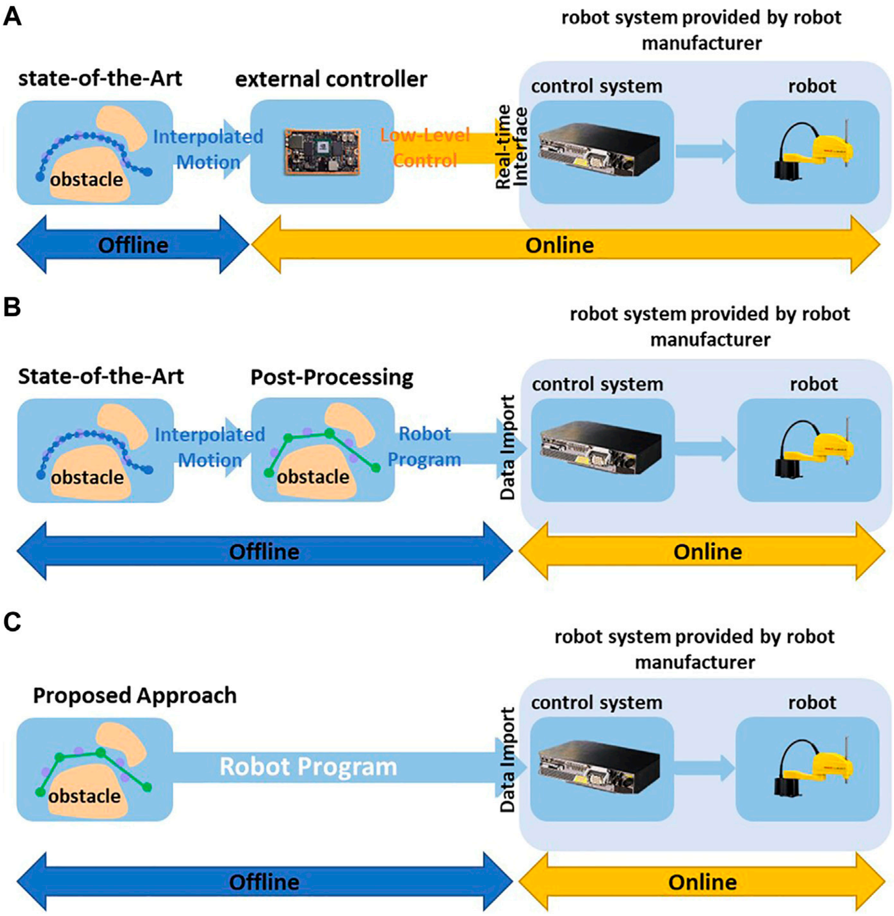


FIGURE 2
(A) Deploying trajectories using post-processing and external control loop. **(B)** Deploying trajectories using post-processing and robot's own control system. **(C)** Direct Deploying trajectories based on High-Level Motion Commands without post-processing and using robot's own control system.

dynamic behavior to interpolate the robot motion and check whether the robot collides with other obstacles. In the proposed approach, the planned robot motion deviates from the realistic robot motion slightly. Therefore, it can be guaranteed that as long as the robot motion planned offline is collision-free, the robot will also not collide with obstacles as executed by the robot control system. This feature addresses the problem described in Section 1.2. The robot motion planned by the proposed approach does not need to be verified iteratively physically during the deployment phase, thus reducing the manual effort and the time-consuming of the engineering process.

The proposed approach is evaluated on two different industrial applications. The results indicate that the proposed approach can generate high-level motion commands directly deployed to real robot systems with a reduced temporal and spatial deviation between executed and planned motions.

2 Related works

2.1 Motion planning methods

2.1.1 Optimization-based motion planning methods

Optimization-based motion planning originated in the field of optimal control and has been used for decades in robotics. The trajectories are usually discretized into *via* points, equally spaced in time. The control inputs at every *via* point are considered as optimization variables, such as angle, velocity and acceleration of robot joints. The collision and the kinematics limits of robot joints are modeled as constraint items. The length, smoothness and execution time of trajectory are described as cost functions that should be minimized. Ratliff et al. (2009), Schulman et al. (2013), Zucker et al. (2013), and Schulman et al. (2014) use different approaches to optimize the modeled motion planning problems with various constraints and objectives. The trajectories are usually finely discretized into a large number of *via* points to find valid solutions in complex and high-dimensional solution spaces.

2.1.2 Sampling-based heuristic search methods

In the past several decades, the sampling-based heuristic search method has been widely adopted in the field of motion planning in high-dimensional configuration space with great success. Rapidly-exploring Random Trees (RRT) (LaValle, 1998), optimal Rapidly-exploring Random Trees (RRT*) (Karaman and Frazzoli, 2011), Fast Marching Tree (FMT) (Janson et al., 2015) and their extensions (Kuffner and LaValle, 2000; Karaman and Frazzoli, 2011; Bdiwi et al., 2018; Otto et al., 2021) explore the configuration space incrementally by connecting feasible samples to a search tree. As the complexity of environments and the DOFs (degrees of freedom) of robot increase, samples are often infeasible. Therefore, the number of samples needs to be raised to achieve probabilistic completeness.

Multiple informed methods explore regions with a higher probability of generating feasible paths to improve the searching efficiency in the configuration space of robots. Data-driven techniques such as supervised learning, imitation learning and deep reinforcement learning techniques are quickly becoming useful tools to improve the efficiency of informed searching in high-dimensional configuration space.

2.1.2.1 Learning sampling strategy

Cheng et al. (2020) learns to predict the optimal sampling distribution over low-cost, valid samples. Based on the learned optimal sampling distribution, the classical searching algorithms are used in the planning phase to guide the search progress towards the region with more optimal, feasible paths. Similarly (Gaebert and Thomas, 2022), uses a CVAE Network to learn a sampling strategy that draws samples based on the environment perception to improve sampling efficiency. In the planning phase, the learned adaptive sampling strategy is used with an adaptive probability λ and a uniform sampling with $1 - \lambda$. The combination of these two strategies guarantees asymptotic optimality. Instead of implicit learning of sampling distribution (Molina et al., 2020; Shah and Srivastava, 2022), learn to predict critical regions that have a high density of feasible motion plans in the given environments.

2.2.2.2 End-to-end learning low-level control policy

In addition to the learning of sampling strategy (Bhardwaj et al., 2017; Huh and Lee, 2018; Jurgenson and Tamar, 2019; Qureshi et al., 2019; Qureshi et al., 2020; Jinwook et al., 2022), learn to directly generate end-to-end low-level control policy to guide the search progress efficiently towards goal regions. These methods learn search strategies from previous planning problems and apply them to new ones. Qureshi et al. (2019) and Qureshi et al. (2020) designs two neural networks. The first one is embedding the points cloud of the environment into a hidden vector. The second network takes the environment embedding, current state, start and goal state as inputs to generate a sample for the next search step. In (Huh and Lee, 2018), a reinforcement learning approach is proposed. The control actions and corresponding state-action values in a given state can be learned in the learning phase. The trajectory expands towards the goal in the planning phase based on the state-action value of possible control action at each search step. Bhardwaj et al. (2017) defines the search process as a Markov decision process and uses dynamic programming to estimate the cost-to-go value of each possible sample. In (Jurgenson and Tamar, 2019), a modified Deep Deterministic Policy Gradient (DDPG) algorithm is proposed to learn control policy through a trial-and-error fashion, which generates data with a more reasonable distribution, including collision-free expert data and data that escapes the obstacle. Jinwook et al. (2022), new trains a Higher Order Function network to represent the cost-to-go function over the configuration space. In the planning phase, the trained network generates a smooth and continuous cost-to-go function directly from workspace information. The gradient of the cost-to-go function yields continuous collision-free trajectories.

The aforementioned learning-based methods generate low-level control inputs, such as position, the velocity of robot joints. These low-level control inputs should be post-processed to be deployed to real robot systems.

2.2 Deploying generated trajectories to robot system

The works mentioned above focus on improving and verifying the performance of collision-free motion planning algorithms in simulation environments rather than on how to deploy the planned robot motion in real robot systems. Bhardwaj et al. (2017); Jurgenson and Tamar (2019), Molina et al. (2020), and Shah and Srivastava (2022) only verify their algorithms in simulation environments. Huh and Lee (2018), Qureshi et al. (2019), Cheng et al. (2020), Qureshi et al. (2020), Gaebert and Thomas (2022), and Jinwook et al. (2022) deploy planned trajectories in real robot systems by using additional controllers to control the robot motion in real time, such as Robot Operation System (Quigley et al., 2009). In these methods, the robots usually run at low speeds to ensure that the robot can precisely track the planned collision-free motion.

Rahali et al. (2022) and Tan et al. (2023) use different algorithms to reduce the motion tracking errors of robots. However, these methods require the robot's dynamics to be identified and modeled. The algorithms in (Elhaki et al., 2022; Elhaki and Shojaei, 2022) are designed to control multibody systems, such as tractors and underwater vehicles, without requiring detailed system models. Different from these systems, industrial robots have own control systems. Any additional control

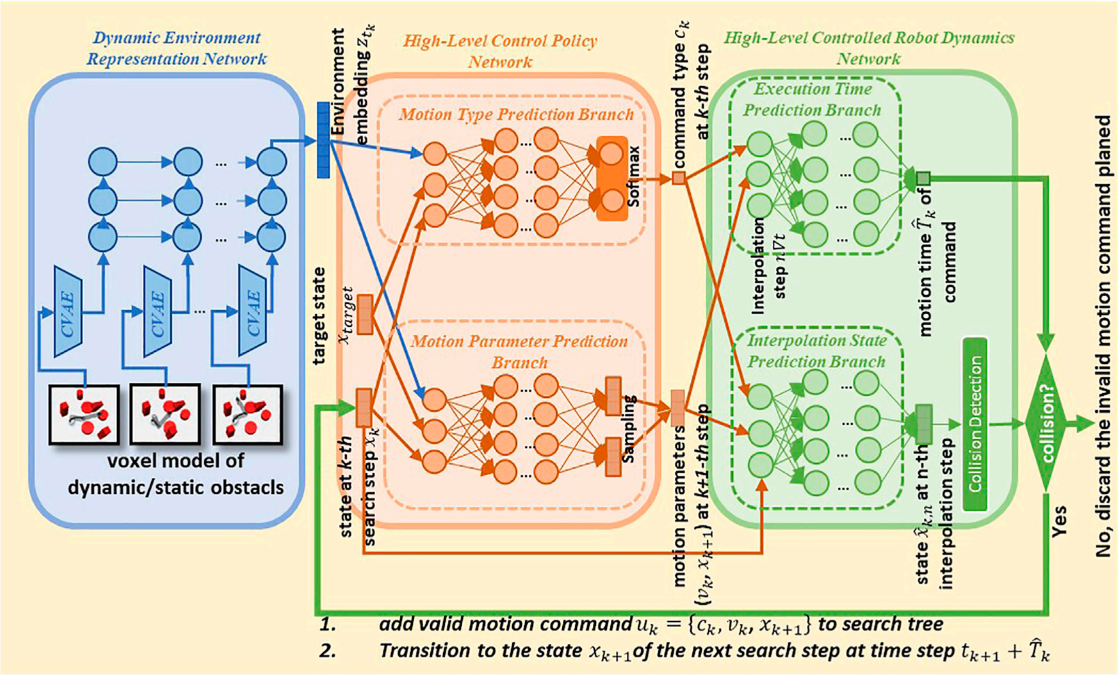

Dynamic Environment Representation Network
CVAE
CVAE
CVAE
voxel model of dynamic/static obstacls
Environment embedding z_{t_k}
High-Level Control Policy Network
Motion Type Prediction Branch
Softmax
target state x_{target}
Motion Parameter Prediction Branch
Sampling
state at k-th search step x_k
command type c_k at k-th step
motion parameters (v_k, x_{k+1}) at k+1-th step
High-Level Controlled Robot Dynamics Network
Execution Time Prediction Branch
Interpolation step $n\nabla t$
motion time $\hat{T}_k$ of command
Interpolation State Prediction Branch
state $\hat{x}_{k,n}$ at n-th interpolation step
Collision Detection
collision?
Yes
No, discard the invalid motion command planed
1. add valid motion command $u_k = \{c_k, v_k, x_{k+1}\}$ to search tree
2. Transition to the state $x_{k+1}$of the next search step at time step $t_{k+1} + \hat{T}_k$

FIGURE 3
Architecture and planning pipeline of the proposed approach.

algorithms must run on an additional controller and control the robot's motors through an interface provided by the robot own control system (Figure 2A). The stability of this control architecture cannot be guaranteed because some parameters of the internal control loop in the robot control system are not accessible. Furthermore, the communication time between the additional controller and the robot control system also affects the stability and performance of the entire control architecture. For example, the control systems of KUKA heavy-duty robots provide an Ethernet-based communication interface (Robot Sensor Interface- RSI) to control the robot motion using an additional control loop. The cycle time of this communication interface is 4 *ms*. Therefore, it limits the control algorithms to reduce tracking errors in higher control frequency. In some industrial applications, the high-speed and long-distance robot motions in 4 *ms* may lead to significant tracking errors.

Again, the methods mentioned above use additional control loops to control the robot in real time to track the motion planned and interpolated in the offline phase. In contrast, the proposed approach does not require an additional control loop to track the planned motion since the proposed approach generates collision-free motions as high-level commands, which can be executed directly by robot control systems with a small deviation from the planned motion of less than .5% on average.

3 Problem definition

This section describes the notations used in this work and formally defines the problem we consider.

Let $\chi \subseteq \mathbb{R}^d$ be the configuration space of a robot system with degrees of freedom $d \in \mathbb{N}$, $d > 2$. Let $\mathcal{U} \subseteq \mathbb{R}^d$ be the control input space of a robotics system. Let the discrete-time dynamics of the robot be defined by f_χ:

$$x_{k+1} = f_\chi(x_k, u_k) \tag{1}$$

where $x_k \in \chi$ and $u_k \in \mathcal{U}$ denote the state and control input of the system at k-th search step.

In contrast to the approaches described in (Bhardwaj et al., 2017; Huh and Lee, 2018; Jurgenson and Tamar, 2019; Qureshi et al., 2019; Cheng et al., 2020; Molina et al., 2020; Qureshi et al., 2020; Gaebert and Thomas, 2022; Jinwook et al., 2022; Shah and Srivastava, 2022), this work considers the high-level motion commands commonly used in robot handling applications as control inputs u_k. These commands typically consist of motion types (such as Point-to-Point, Linear and Circle motion) and motion parameters such as motion velocity and the desired state to be reached.

This work considers static obstacles and dynamic obstacles whose motions are known. For example, for the multi-robot system, motions of all robots are usually planned one by one. When planning the motion of a given robot, the motions of the other robots are known. Let $\chi_{feasible,t} \subseteq \chi$ define the feasible state space of the robotics system, in which the robot did not collide with static and dynamic obstacles at timestep t, $x_{init} \in \chi_{feasible,0}$ the initial state, and $x_{goal} \in \chi_{feasible,t}$ the goal state.

In this work, a trajectory π is defined as a series of states and high-level control commands:

$$\pi = \left(x_0, u_0, x_{t_0}, x_1, u_1, x_{t_1}, \ldots, x_k, u_k, x_{t_k}\right) \tag{2}$$

where t_k is the timestep of k-th *via* point in the trajectory.

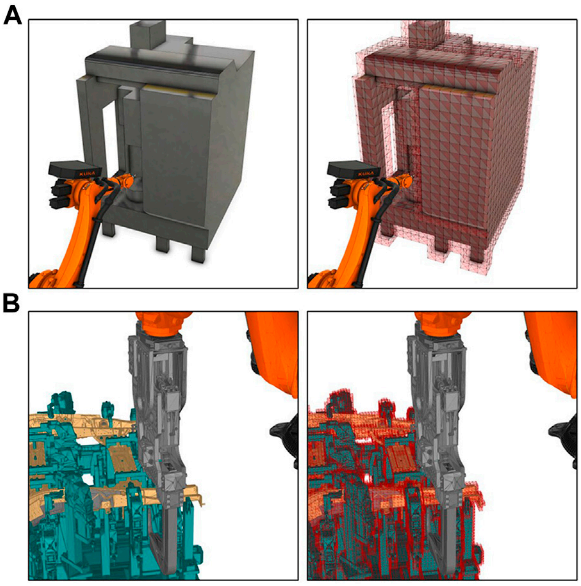


FIGURE 4
(A) Voxel model with low resolution (edge length of each voxel is 8 *cm*). **(B)** Voxel model with high resolution (edge length of each voxel is 1 *cm*).

3.1 Main problem

For complex environments and robot systems with high DOFs (degrees of freedom), the solution space of the motion planning problem is highly dimensional. Even if the solution space is represented implicitly using the sampling-based technique, it cannot be searched efficiently. In this work, the proposed approach focuses on learning the feasible solution space of motion planning problems from previous experience to improve search efficiency. In other words, the proposed approach first learns to perceive the environment surrounding the robot. Then the robot dynamics are learned to preciously simulate realistic robot motion. At last, the proposed approach learns which optimal high-level commands can move the robot toward to goal region with realistic dynamics in the perceived environment model at each search step.

3.2 Subproblem 1: Learning local feasible solution space of motion planning problem

Since learning the complete solution space is very difficult and does not scale well to other problems, our approach begins with learning the local feasible solution space $\mathcal{L}_{\text{local}}$:

$$\mathcal{L}_{\text{local}}\left(x_k, \phi_k \middle| x_{\text{goal}} \rightarrow u_k\right) \tag{3}$$

The locally feasible solution space consists of all feasible control policies that only consider the local system state (e.g., the state of the environment ϕ_k, the current state of the robot x_k and the target state χ_{goal}) and guide the robot from the current state toward the target area with control command u_k at k-th search step.

3.3 Subproblem 2: Suitable representation of dynamic environment

Since this work considers environments with static and dynamic obstacles, the geometric and temporal information of the environment should be represented as environment state ϕ_k at k-th search step and used in subproblem 1.

3.4 Subproblem 3: Learning robot dynamics f_χ controlled by high-level motion commands

The execution time and interpolation of robot motion between two states should be calculated to check the collision between the robot and the obstacles during the transition from one state to the next state at each search step. As mentioned before, the robot dynamics controlled by high-level motion commands are seen as a "black box." Therefore, the proposed approach learns the realistic robot dynamics controlled by high-level motion commands and uses it to calculate the realistic robot motion.

4 Methods

The core of the proposed approach is three neural networks (Figure 3) which solve the main problem described in Section 3. Sections 4.1–4.3 describe the functionality of each neural network and how they solve the corresponding subproblems. Then we give an overview of the entire pipeline of the proposed approach in Section 4.4.

4.1 Dynamic Environment Representation Network for subproblem 2

Since the motions of the dynamic obstacles are known, the environment can be discretized into a series of frames. The 3D model, such as the voxel model of the environment at each frame, can represent spatial and geometric information.

However, directly using this high-dimensional representation to learn control policy for subproblem 2 leads to a large-scale network that may be difficult to train. Therefore, a separate network struct is used to extract spatial and temporal features of the environment as low-dimensional representation.

Firstly, an encoder embeds the voxel model of the dynamic environment at each discrete timestep into a hidden vector s_t. Let denote this embedding as $h(\phi_t)$, which compresses the spatial state of the dynamic environment ϕ_t at the timestep t:

$$s_t = h(\phi_t) \tag{4}$$

Then an RNN-based encoder embeds the temporally ordered hidden vectors s_t, $s_{t+1}\ldots$, s_{t+n} into a hidden vector z_t, which represents the temporal information of the environment after current timestep t.

$$z_t = r(s_t, s_{t+1}, \ldots, s_{t+n}) \tag{5}$$

TABLE 1 Four categories of environments of the SCARA robot handling application used in experiment.

Categories of environments	Static obstacles	Dynamic obstacles
Simple Staitc Environments	1x Cylinder or Cubic	0
	1x Robot	
Complex Static Environments	3x Cylinders or Cubics	0
	1x Robot	
Simple Dynamic Environments	1x Cylinder or Cubic	1x Roboter
Complex Dynamics Environment	3x Cylinders or Cubics	1x Roboter

4.2 High-level control policy network for subproblem 1

The high-Level Control Policy Network is the core component of the proposed approach. Let denote the high-level control policy network q_θ with its parameter as

$$u_k = q_\theta\left(x_k, z_{t_k}, x_{goal}\right) \tag{6}$$

When the robot arrives a state x_k after k-th search step at timestep t_k, the network takes the current state x_k, the embedding of the dynamic environment z_{t_k} at timestep t_k and the target region χ_{goal} as inputs to generate a high-level motion command u_k. The high-level motion command u_k consists of command type c_k and corresponding motion parameters, such as the motion speed v_k and the desired state x_{k+1} to be reached.

4.3 High-Level Controlled Robot Dynamics Network for subproblem 3

We designed a neural network to predict the dynamics of robots controlled by high-level commands. This network predicates the execution time and interpolation of high-level motion commands.

$$\hat{T}_k = f_{execution\_time}(x_k, u_k) \tag{7}$$

$$x_{k,n} = f_{interpolation}(x_k, u_k, t_k + n\nabla t) \tag{8}$$

where $f_{execution\_time}$ denotes the function to predicate the execution time $\hat{T}_k$, x_k the current state and u_k the motion command, respectively. $f_{interpolation}$ denotes the function to predicate an interpolation state of the robot motion $\hat{x}_{k,n}$ at n-th timestep $t_k + n\nabla t$, where ∇t denotes the interpolation resolution.

4.4 Robot motion planning with learned feasible solution space

The entire pipeline of the proposed approach consists of the following procedures.

4.4.1 Data collection

Firstly, the expert data for training the above-described neural networks should be collected. The proposed approach requires plenty of realistic data, which are expensive in terms of time and resources. Therefore, the near-realistic simulation environment Visual Components (Visual Components, 2021) are used to generate realistic datasets and verify the planning results. Visual Components contains an offline programming system that can connect with VRC module (Virtual Robot Controller) (Bernhardt et al., 1994). The VRC module integrates the original robot controllers and provides a simulation accuracy of .00005 radians and 1% cycle time. In this work, following data are collected in the simulation environment.

4.4.4.1 Environment data

In offline robot programming, the geometry of the production cell is represented through 3D polygon mesh models in simulation software. The polygon mesh models of the obstacles in the environment are exported and collected as raw data for training Dynamic Environment Representation Network. Then the 3D polygon mesh of obstacles is rasterized into 3D voxel models because 3D voxel grids have a highly regular data format, which is suitable for representation learning. In contrast to representing the environment in point clouds, the resolution of the voxel model can be easily adapted to suit the diverse requirements for environment representation for different robot applications.

For example, in some high-speed handling tasks, the robot should keep a safe distance from the obstacles in the environment. In this case, the edge length of the voxel grids occupied by the obstacles should be increased to leave enough space between the robot and the obstacles (Figure 4A). However, for tasks that require the robot to perform delicate operations, such as spot welding tasks where the welding gun enters some narrow areas, we need to increase the resolution of the voxel models to represent more details of the narrow areas (Figure 4B).

4.4.4.2 Robot programs

Robot programs for scenarios of different applications are collected to learn high-level motion commands. The robot programs consist of high-level motion commands, which are programmed manually or automatically through other motion planners. These robot programs should be executed and verified on real robot systems or near-realistic simulation environments to ensure that the programmed robot motions are collision-free.

4.4.4.3 Realistic robot motions

Training the High-Level Controlled Robot Dynamics Network requires realistic robot motions executed by the robot control system. On the one hand, the collision-free robot motions generated in Section 4.4.1.2 are be reused. On the other hand, more high-level commands are randomly generated. These commands should also be executed on real robot systems or near-realistic simulation environments to collect realistic robot motions.

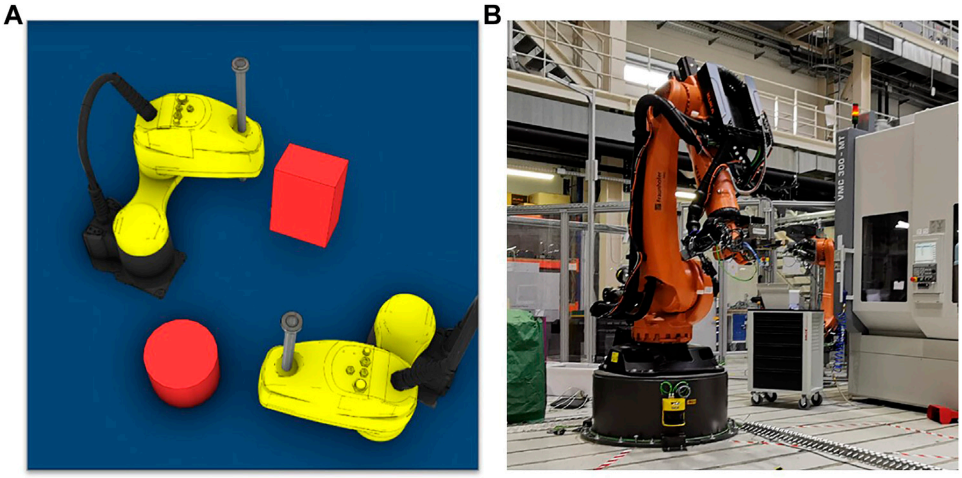


FIGURE 5
(A) SCARA robot handling application (only in simulation environment). **(B)** Machine tending application with a 6-axis heavy-duty industrial robot.

4.4.2 Model training

In the second procedure, the three neural networks described in Section 4.3 are trained on the data gathered in Section 4.4.1. All three neural networks are trained in an offline supervised fashion. The experiment settings for model training are detailed in Section 5.2.

4.4.3 Offline robot motion planning

In the offline planning procedure, the trained neural network models are used to search a collision-free robot motion from the initial state x_{init} to the goal state x_{goal}. The search process starts from the initial state x_k. At each search step k (at the timestep t_k), Dynamic Environment Representation Network embeds the dynamic environment into a low-dimensional hidden vector z_{t_k} (see the blue block in Figure 3). z_{t_k} is then fed into the High-Level Control Policy Network along with the current state x_k and the target state x_{target} to generate a high-level motion command u_k consisting of motion command type and motion parameters (see the orange block in Figure 3). The High-Level Controlled Robot Dynamics Network then takes u_k as input to predict all interpolation states $x_{k,n}$ and execution time $\hat{T}_k$ of the motion to the next search step. For each interpolation state, the collision between the robot and obstacles is checked using conventional forward kinematics and the collision check algorithm proposed in (Pan et al., 2012). If the robot motion is collision-free, u_k will be added to the search tree and the search process will transit to the next state x_{k+1} at the timestep $t_k + \hat{T}_k$ (see the green block in Figure 3). The planning pipeline repeats until the goal state is reached.

4.4.4 Deploying robot motion to robot system

The robot motions planned by the proposed approach are in a general format of high-level motion commands. Because the robot programming must follow robot manufacture-specific programming rules, the general format of high-level motion commands should be converted to robot manufacturer-specific programming language using a post-processor. Then the robot programs can be directly uploaded to the robot control system. It should be noted that the post-processing here is the syntactical conversation, which is different from the post-processing mentioned in Section 1.1.

5 Experiment design and implementation

This section reports the experiment settings and the implementation details of the proposed approach.

5.1 Experiment setup

We evaluate the proposed approach on two industrial applications: A handling application with two SCARA robots and a machine tending application with one 6-axis heavy-duty robot.

5.1.1 SCARA robot handling application

In this application, two SCARA robots perform pick-and-place tasks in different environments containing static and dynamic obstacles. It is important to note that although this application contains two SCARA robots, only the motions of one robot need to be planned, and the other robot is seen as a static or dynamic obstacle. Table 1 details the static and dynamic obstacles in four different categories of environments. In this application, we focus on evaluating the offline planning phase. Thus, the planned motions are only verified in the simulation environment (Figure 5A).

5.1.2 Machine tending application

This application evaluates the proposed approach for the problem domain of high-dimensional motion planning. A 6-axis heavy-duty robot loads and unloads a machine tool. Unlike the SCARA robot handling application, the planned robot motion in this application will be deployed and verified on a real robot-based machine tending system to evaluate the complete pipeline from planning until deploying robot motions (Figure 5B).

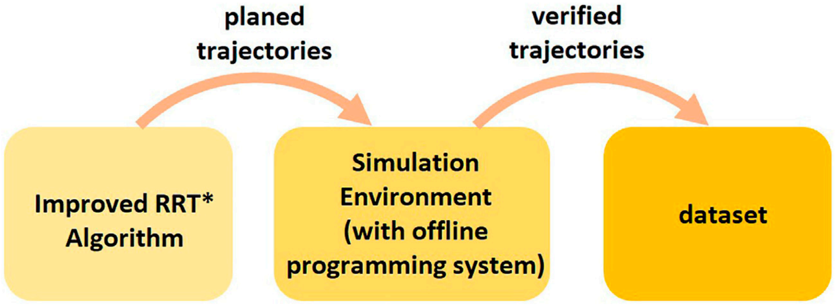


FIGURE 6
Procedure of collecting dataset for High-Level Control Policy Network.

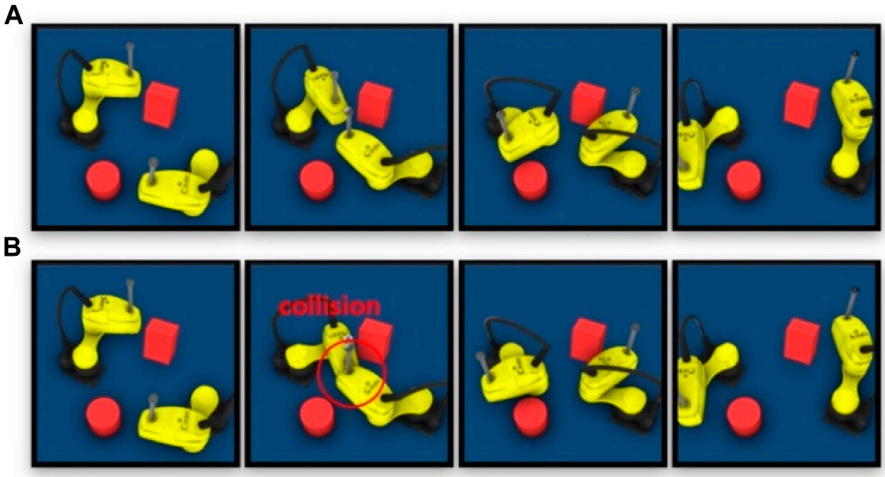


FIGURE 7
(A) Planned trajectory without collision. **(B)** Executed trajectory with collision.

5.2 Implementation

This section describes the structure of neural network models and the datasets.

5.2.1 Dynamic Environment Representation Network

The Dynamic Environment Representation Network uses the basis struct of Variational Autoencoders (VAE) (Kingma and Welling, 2014) with five 3D-CNN layers (Ji et al., 2013) to compress the static obstacles in the environment into a 20-dimensional embedding. For the dynamic environment, for example, in SCARA robot handling application, a 3-layer RNN encoder with ten units to embed the changes of the dynamic environment over time. Each of the ten units accepts a 20-dimensional embedding of one frame of the dynamic environment and the RNN encoder finally produces an embedding vector of the dynamic environment. (See the blue block in Figure 3).

For the SCARA robot handling application, we randomly generate 1000 environments. Each environment contains a varying number of static cylindrical or cubic obstacles and a SCARA robot seen as a dynamic obstacle. Then we recorded a frame of the dynamic environment every 50 milliseconds. The environment in each frame is voxelized and fed into VAE to produce an environment embedding. We take ten frames of environment embedding following the current timestep as a data tuple for training the RNN encoder.

For the machine tending application, 500 static environments are generated. In each environment, we select one of five different machine tools and place it randomly within the reachable workspace of the robot.

5.2.2 High-level control policy network

A high-level motion command consists of the motion type and motion parameters. In this work we consider Point-to-Point (PTP) and Linear motion of the SCARA robot and the 6-axis robot. The motion parameters of both motion commands are the motion speed v_k and the state x_{k+1} to be reached.

The High-level Control Policy Network contains two branches: one generates motion type (Motion Type Prediction Branch) and the other generates motion parameters (Motion Parameter Prediction Branch). These two branches take the same inputs: the goal state x_{Goal},

TABLE 2 Average prediction error of High-Level Controlled Robot Dynamics Network. The error in predicting execution time is defined as $\frac{\hat{T}_k - T_k}{T_k}$, where $\hat{T}_k$ and T_k are predicted execution time and actual execution time, respectively. The error in predicting motion interpolation is defined as $\frac{\sum_0^i |(\hat{x}_{k,i} - x_{k,i})|}{l_k}$, where $\hat{x}_{k,i}$ and $x_{k,i}$ are prediction and ground truth of robot state at interpolation step of robot motion, respectively. l_k is the euclidean distance along the robot motion executed. Because the 6-axis robot in the machine tending application does not need to avoid other dynamic obstacles in the machine tending application, the robot moves with 100% velocity override to achieve the shortest cycle time.

Motion speed override		Average prediction error of execution time		Average prediction error of interpolation	
		PTP motion (%)	Linear motion	PTP motion (%)	Linear motion
SCARA robot handling application	0 - 25%	2.7	4.6%	.23	.57 %
	25% - 50%	2.9	5.8%	.47	.79 %
	50% - 75%	4.3	6.6%	.48	.86 %
	75% - 100%	6.0	7.2%	.65	.91 %
Machine Tending Application	100%	1.4	—	.31	—

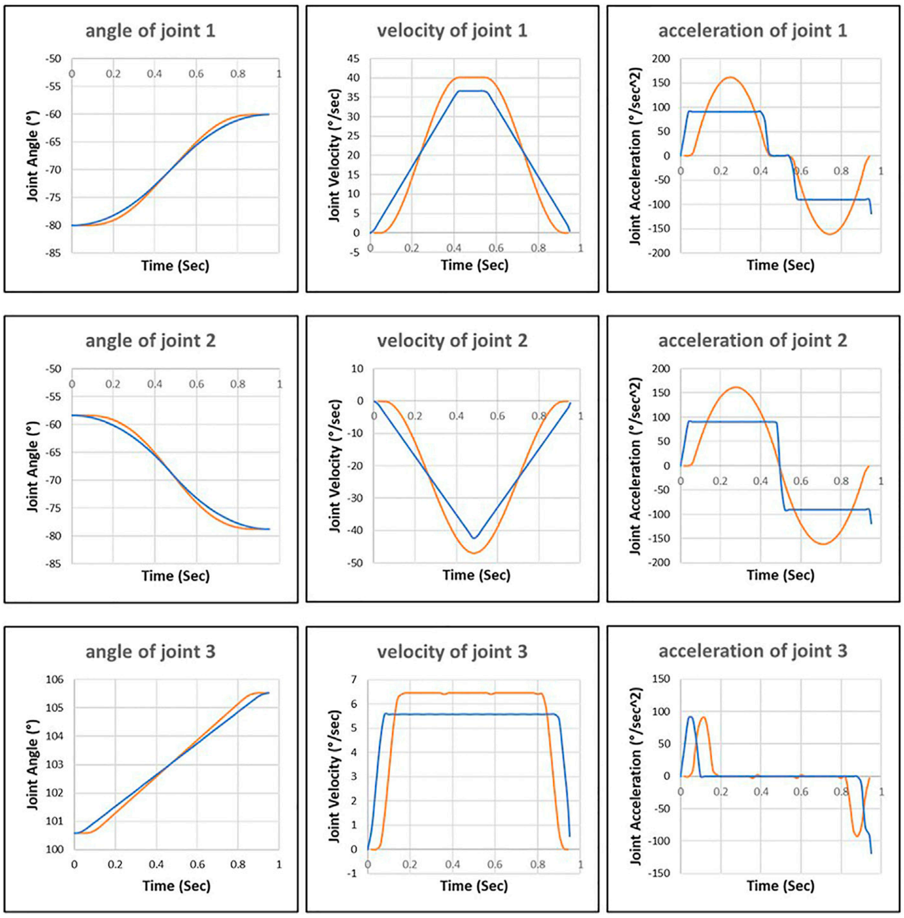


FIGURE 8
Joint motion planned by the improved RRT* approach (blue line) and joint motion executed by robot control system (orange line).

the current state x_k and the environment embedding z_{t_k} at search step k.

The Motion Type Prediction Branches for the SCARA robot handling and the machine tending applications consist of 10 and 12 fully connected hidden layers followed by a Softmax layer with a two-dimensional output, respectively. The Motion Parameter Prediction Branch is a 12-layer forward neural network for the SCARA robot handling application and a 15-layer forward neural

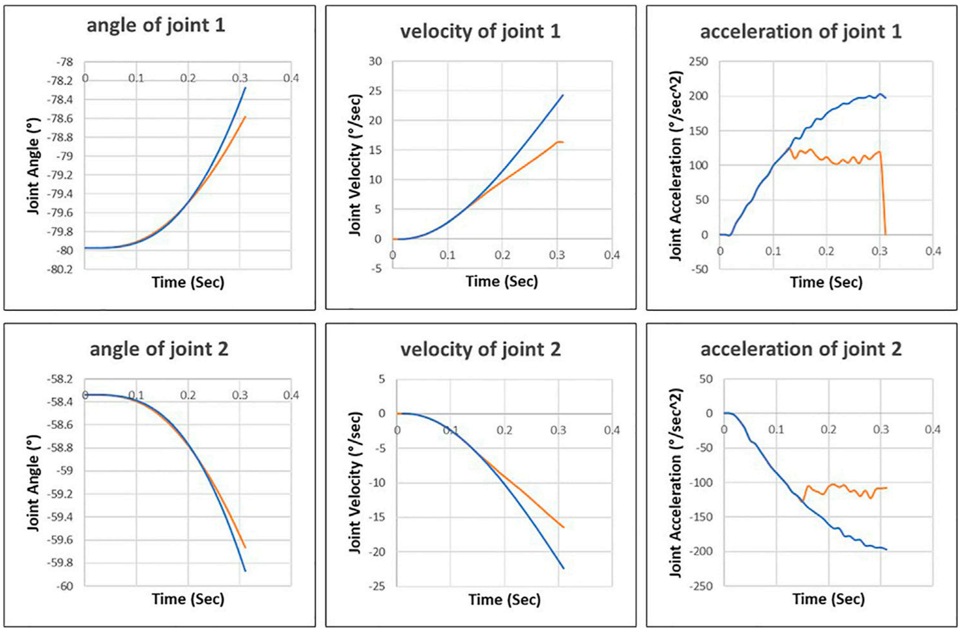


FIGURE 9
Control inputs of external control loop (blue line) and actual joint motion (orange line).

network for the machine tending application, respectively. Unlike the network structure that generates samples at every search step in (Bdiwi et al., 2018), we do not use dropout layers to achieve stochasticity in the Motion Parameter Prediction Branch because the dropout layer affects the convergence of the neural network. Inspired by the struct of VAE, we applied two hidden layers before the output layer to generate two vectors simultaneously: means and standard deviations vector of motion parameters. The output layer samples a final prediction of motion parameters from the means and standard deviations. (See the green block in Figure 3).

In the environments generated in Section 5.2, we collect data for training High-level Control Policy Network. Each environment of the SCARA robot handling application and the machine tending application contains 50 start-goal pairs. In order to make the data set closer to the real machine tending applications, the start position or the goal position of each start-goal pair must be located over the working table inside the machine tool.

An improved RRT* approach (Otto et al., 2021) is used to plan a trajectory as expert data. Unlike the basic RRT, the improved RRT* approach post-processes the planned robot motions using PTP and Linear interpolation and generates containing high-level motion commands. The robot motions post-processed by the improved RRT* algorithm may collide with static and dynamic obstacles during the execution due to the inaccurate estimation of robot dynamics and control parameters in the planning phase. Therefore, we execute all generated trajectories in the simulation environment Visual Components with the VRC module and only add the collision-free trajectories and corresponding environment models to the training set (Figure 6).

Both branches of the proposed network are trained in a supervised fashion. The loss of the first branch $L_T(\theta)$ is defined as:

$$L_T(\theta) = -\sum_{i=1}^{2} c_i \log(p_i) \tag{9}$$

where i indicates the category of motion command type c and p_i represents the predicted probability of the command type c_i.

The loss of the second branch $L_P(\theta)$ is defined as:

$$L_P(\theta) = ||\hat{v}_k - v_k|| + ||\hat{x}_{k+1} - x_{k+1}|| \tag{10}$$

where $\hat{v}_k$ and $\hat{x}_{k+1}$ are predicated motion parameters. v_k and x_{k+1} are the corresponding ground truth. We use adam optimizer (Kingma and Adam, 2015) with initial learning rate .001, momentum .9. The learning rate is decreased by half every 50 epochs.

5.2.3 High-Level Controlled Robot Dynamics Network

High-Level Controlled Robot Dynamics Network has two branches, the Interpolation State Prediction Branch and Execution Time Prediction Branch, to predict the interpolation states and execution time of realistic robot motion.

The Interpolation State Prediction Branches consist of 12 and 14 fully connected hidden layers for the SCARA robot handling and the machine tending applications, respectively. The Execution Time Prediction Branch consists of 10 and 11 fully connected hidden layers for the SCARA robot handling and machine tending applications, respectively. The Interpolation State Prediction Branch takes the current state x_k and motion command c_k with motion parameter (v_k and x_{k+1} for Point-to-Point and Linear motion) as input to predicate the execution time $\hat{T}_i$. A given interpolation step $t_k + n\nabla t$ along with the same input as Interpolation State Prediction Branch is fed into the Execution Time Prediction Branch to predict the interpolation state of the robot $\hat{x}_{k,n}$ at the given interpolation step $t_k + n\nabla t$ (See the orange block in Figure 3).

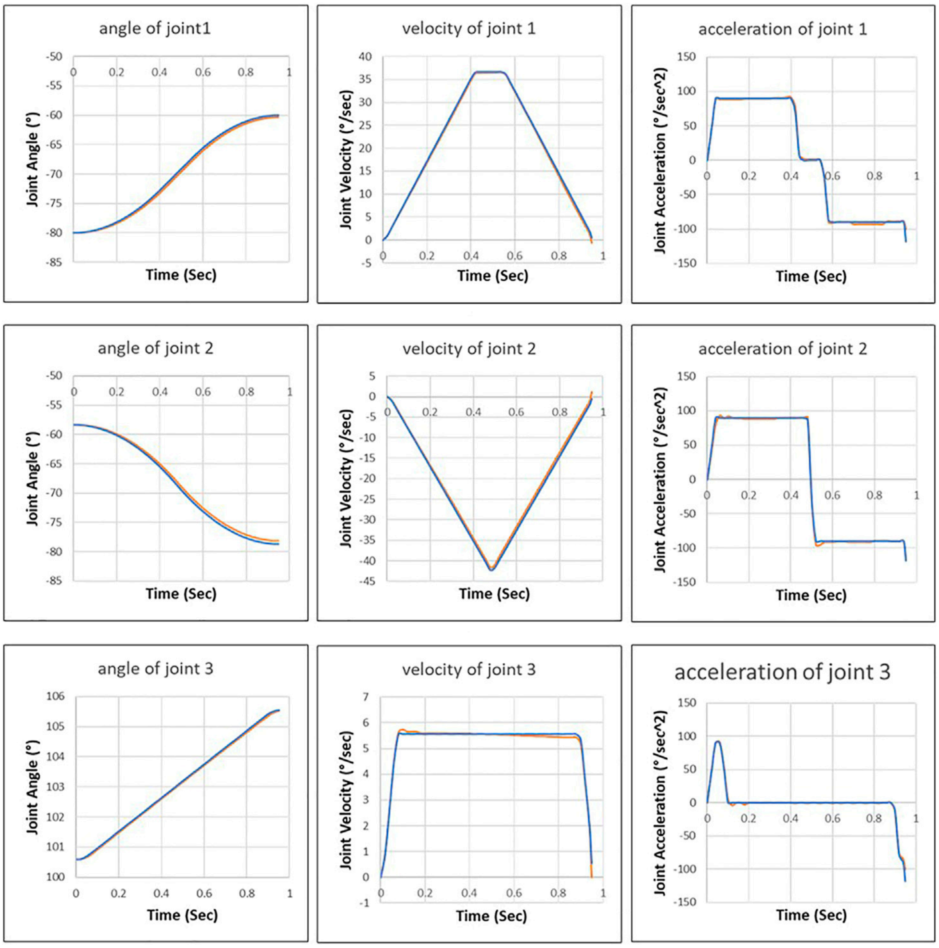


FIGURE 10
Joint motion planned by the proposed approach (blue line) and joint motion executed by robot control system (orange line).

The VRC Modul in Visual Components executes ten thousand motion commands of the SCARA robot and fifteen thousand motion commands of the 6-axis heavy duty robot. The execution time and the interpolation states of executed motion commands are recorded as the dataset.

The first and second branches are trained by using standard L2 loss function $L_{Interpolation}(\theta)$ and $L_{ExecutionTime}(\theta)$, respectively:

$$L_{Interpolation}(\theta) = \left\|\hat{x}_{k,n} - x_{k,n}\right\| \tag{11}$$

$$L_{ExecutionTime}(\theta) = \left\|\hat{T}_k - T_k\right\| \tag{12}$$

where $x_{k,n}$ and $\hat{x}_{k,n}$ denote the ground truth and prediction of robot state at the interpolation step n, respectively. T_k and $\hat{T}_k$ denote the ground truth and prediction of execution time, respectively. During training, we use stochastic gradient descent (SGD) [35] with initial learning rate .0005 and momentum .8.

6 Result and discussion

For each application, this section evaluates the proposed approach in 100 new environments, which are not used in the training phase. In each environment, 20 pairs of start and goal were randomly generated. The performance of the RRT, the improved RRT* and the proposed approach was analyzed in terms of validity, the execution time of trajectory and computation time.

6.1 Validity of trajectory

6.1.1 SCARA handling application

For the SCARA handling application, the robot motions planned offline by different planners are only verified in Visual Components. Figure 7 shows an example of an invalid trajectory generated by RRT. Figure 7A shows that when the SCARA robot follows the planned trajectory exactly, the robot on the right side passes the shared area before the robot on the left side. The executed motion of the robot is slower than computed in the planning phase and enters the shared area later than planned, resulting in a collision with a cubic obstacle (Figure 7B).

In all scenarios of SCARA handling application, only 5.2% of the trajectories generated by our approach are invalid because the trained High-Level Controlled Robot Dynamics Network can predicate the robot motion more accurately in the planning

TABLE 3 Average execution time of trajectories generated by proposed approach and the benchmark approaches.

Environment	Distance between start and goal	Average execution time in second		
		Proposed approach	RRT	Improved RRT*
Simple static environment of SCARA robot handling application	Near	.223	.210	.212
	Middle	.420	.544	.513
	Far	.661	.837	.702
Complex static environment of SCARA robot handling application	Near	.296	.370	.306
	Middle	.605	.801	.664
	Far	.736	.909	.759
Simple dynamic environment of SCARA robot handling application	Near	.246	.276	.266
	Middle	.495	.593	.563
	Far	.733	.948	.829
Complex dynamic environment of SCARA robot handling application	Near	.420	.464	.467
	Middle	.766	.978	.827
	Far	1.070	1.292	1.116
Machine tending application	Near	1.523	1.892	1.328
	Middle	2.034	2.367	2.249
	Far	3. 551	3.719	3. 406

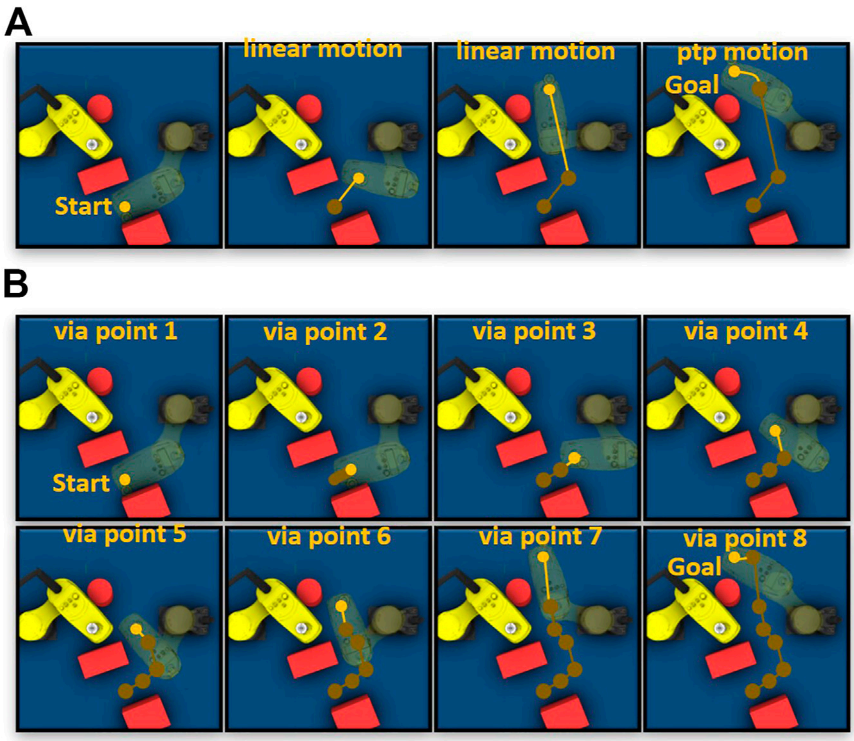


FIGURE 11
(A) Example trajectory generated by the proposed approach. **(B)** Example trajectory generated by RRT. The green points represent the *via* points of trajectory in Cartesian space.

TABLE 4 Average computation time of trajectories generated by the proposed approach and benchmark approaches.

Environment	Distance between start and goal	Average computation time in second		
		Proposed approach	RRT	Improved RRT*
Simple static environment of SCARA robot handling application	Near	.18	.37	1.82
	Middle	.26	.61	2.27
	Far	.31	.84	2.38
Complex static environment of SCARA robot handling application	Near	.62	1.50	2.47
	Middle	.74	2.81	4.77
	Far	.88	4.16	5.49
Simple dynamic environment of SCARA robot handling application	Near	.42	1.83	5.05
	Middle	.57	3.59	6.92
	Far	.59	4.24	7.22
Complex dynamic environment of SCARA robot handling application	Near	.67	3.12	5.34
	Middle	.89	5.94	8.03
	Far	.94	7.12	9.11
Machine tending application	Near	.90	.85	1.15
	Middle	1.08	.88	1.27
	Far	1.17	.92	1.63

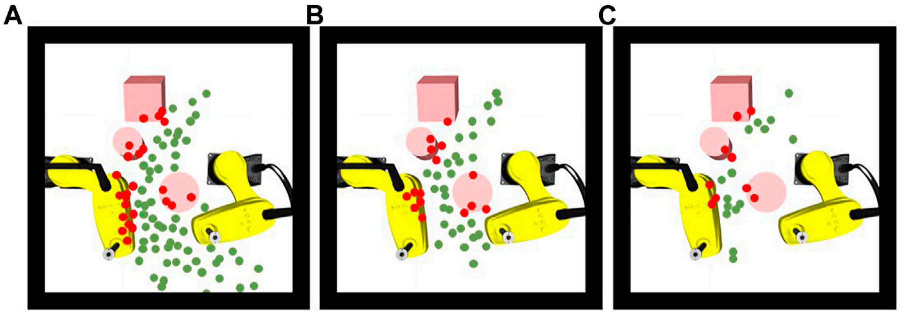


FIGURE 12
(A) Valid (green) and invalid (red) samplers generated by RRT, **(B)** RRT* and **(C)** by the proposed approach.

phase. Table 2 shows the relative error of the trained model in predicting motion interpolation and execution time. In all experiment scenarios, the average error between the actual and predicted execution time of the high-level motion commands is 5%. Furthermore, Table 2 shows that the error in predicting Point-to-Point motion is smaller than that in predicting linear motion. The reason is that predicting the dynamics of linear motion requires estimating the inverse kinematic model, which increases the prediction error.

6.1.2 Machine tending application

For the machine tending application, we deployed the robot motions planned offline on the real robot in different ways. The proposed approach generates high-level motion commands that can be directly uploaded to the robot control system (see Figure 2C). Because the improved RRT* uses an interpolation algorithm to convert the planned robot motion to high-level motion commands, the generated motion commands can also be uploaded into the robot control system (see Figure 2B). The RRT generates low-level control inputs, which should be executed in an additional control loop (see Figure 2A).

In Figure 8, we can see that the robot motion planned by the improved RRT* deviates significantly from the robot motion executed by the control system. It is because the control algorithm of the improved RRT* used in the planning phase differs from the control algorithm used in the robot control system-in the offline planning phase, the improved RRT* assumes that the joints can reach the maximum acceleration. However, in reality, the robot control system only applies 60% and 45% of the maximum acceleration to the first and second joints, respectively.

The control inputs and actual values are recorded during the execution of robot motion controlled by the additional control loop (Figure 9). The additional controller tries to drive the first and second joints with maximum acceleration, but the internal motor controller limits the joints to reach the maximum value. Then the fluctuation of joint acceleration triggers the safety mechanism of the robot control system, which disconnects the communication interface (Robot Sensor Interface) between the additional controller and the robot control system.

Figure 10 shows that the robot motion planned by the proposed approach is close to the motion executed by the robot control system. We can see that the trained neural network has learned the control behavior (acceleration and deceleration) of the robot control system to predicate the interpolation of robot motion.

6.2 Execution time of trajectory

We compared the execution times of the trajectories generated by the proposed approach, RRT and improved RRT* (Table 3).

It is necessary to note that the trajectory's execution time varies significantly due to different distances between the start and goal states. To compare the performance of different approaches more reasonably, we classify the planning tasks into three categories according to the distance between the start and goal states: 1. near distance (smaller than 30% of the robot's range), 2. middle distance (bigger as 30% but smaller as 60% of the robot's range) and 3. far distance (bigger as 60% of the robot's range). It can be seen that the average execution time of the trajectories generated by the proposed approach is twenty percent faster than RRT in the SCARA robot handling application. Since improved RRT* optimizes the number of *via* points while expanding the search tree, the execution time of the trajectories generated by it is essentially the same as the proposed approach. However, the optimization increases computation time, as seen in section 6.3. Since all scenarios of the machine tending application are simple, the execution time of the motion planned by each approach varies slightly.

Figure 11 shows the valid trajectories generated by the proposed approach and RRT in an example scenario. The proposed approach generates a trajectory containing only three high-level motion commands (Figure 11A). The first and second linear motion commands guide the robot through a narrow area. After the robot leaves the narrow area, the proposed High-level Control Policy Network maps the empty surrounding area to a Point-to-Point motion command because the Point-to-Point motion is faster than the linear motion. RRT generates more *via* points (Figure 11B) in the narrow area, resulting in acceleration and deceleration of robot joints.

6.3 Computation time

We compared the computation time of the proposed approach with the benchmark approach in scenarios with different complexities. As the environment becomes more complex, the advantage of our approach in terms of computation time becomes obvious (Table 4). In particular, the proposed approach is up to 10 times faster than the improved RRT* approach in complex dynamic environments of SCARA robot handling application because the proposed approach reduces the computation time by efficiently exploring in learned feasible solution space. In Figure 12, we visualize all the samples generated by the different approaches for the same task. It has been found that the benchmark approaches spent much time to generate a large number of samples randomly. The proposed approach generates fewer samples in critical areas based on environment information.

7 Conclusion

We have proposed a novel deep neural network that generates collision-free trajectories as high-level motion commands. The generated trajectory can be directly deployed in the robot control system without post-processing. Furthermore, the experiment results show that the proposed approach outperforms the benchmark approaches in terms of validity, execution time of planned motion and computation time. One future direction is to extend our data collection procedure and generalize our network to handle more high-level commands for robots with higher degrees of freedom.

Data availability statement

The datasets presented in this article are not readily available because NDA is necessary. Requests to access the datasets should be directed to shuxiao.hou@iwu.fraunhofer.de.

Author contributions

SH developed the theory. He worked out almost all of the technical details and performed the experiments. SH wrote the manuscript with support from all authors. All authors discussed the results and commented on the manuscript.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bdiwi, M., Hou, S., and Delang, K. (2018). "Human-robot-cooperation real time robot path planning for dynamic HRC-applications," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 5542.

Bernhardt, R., Schreck, G., and Willnow, C. (1994). The realistic robot simulation (rrs) interface. *IFAC Proc. Volumes IFAC Workshop Intelligent Manuf. Syst.* 27, 321–324. doi:10.1016/s1474-6670(17)46044-7

Bhardwaj, M., Choudhury, S., and Scherer, S. (2017). "Learning heuristic search via imitation," in Conference on Robot Learning, 271–280.

Cheng, R., Shankar, K., and Burdick, J. W. (2020). "Learning an optimal sampling distribution for efficient motion planning," in IEEE International Conference on Intelligent Robots and Systems, IEEE, 7485–7492.

Chitta, S., Sucan, I., and Cousins, S. (2012). Moveit![ros topics]. *IEEE Robotics Automation Mag.* 19 (1), 18–19. doi:10.1109/mra.2011.2181749

Elhaki, O., Shojaei, K., and Mehrmohammadi, P. (2022). Reinforcement learning-based saturated adaptive robust neural-network control of underactuated autonomous underwater vehicles. *Expert Syst. Appl.* 197, 116714. doi:10.1016/j.eswa.2022.116714

Elhaki, O., and Shojaei, K. (2022). Output-feedback robust saturated actor–critic multi-layer neural network controller for multi-body electrically driven tractors with *n*-trailer guaranteeing prescribed output constraints. *Robot. Aut. Syst.* 154, 104106. doi:10.1016/j.robot.2022.104106

Gaebert, C., and Thomas, U. (2022). "Learning-based adaptive sampling for manipulator motion planning," in IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 715–721.

Hauser, K., and Ng-Thow-Hing, V. (2010). "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in 2010 IEEE international conference on robotics and automation (IEEE), 2493–2498.

Huh, J., and Lee, D. D. (2018). Efficient sampling with q-learning to guide rapidly exploring random trees. *IEEE Robotics Automation Lett.* 3 (4), 3868–3875. doi:10.1109/lra.2018.2856927

Janson, L., Schmerling, E., Clark, A., and Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. Robotics Res.* 34 (7), 883–921. doi:10.1177/0278364915577958

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Analysis Mach. Intell.* 35 (1), 221–231. doi:10.1109/tpami.2012.59

Jinwook, H., Lee, D. D., and Isler, V. (2022). *Neural cost-to-go function representation for high dimensional motion planning. Workshop: Motion planning with implicit neural representations of geometry*. ICRA.

Jurgenson, T., and Tamar, A. (2019). "Harnessing reinforcement learning for neural motion planning," in *Robotics: Science and systems*, 1–13.

Karaman, S., and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *Int. J. Robotics Res.* 30 (7), 846–894. doi:10.1177/0278364911406761

Kingma, D. P., and Adam, J. Ba. (2015). "A method for stochastic optimization," in International Conference on Learning Representations.

Kingma, D. P., and Welling, M. (2014). "Auto-encoding variational bayes," in 2014 International Conference on Learning Representations, Banff, Canada, 1–14.

Kuffner, J., and LaValle, S. M. (2000). "RRT-connect: An efficient approach to single-query path planning," in Proceedings of the IEEE International Conference on Robotics and Automation (San Francisco, CA, United States), 995–1001.

LaValle, S. M. (1998). *Rapidly-exploring random trees: A new tool for path planning*. Computer Science Dept., Iowa State University. tR 98-11.

Molina, D., Kumar, K., and Srivastava, S. (2020). "Learn and link: Learning critical regions for efficient planning," (Paris, France: in IEEE International Conference on Robotics and Automation).

Otto, A., Hou, S., Ahrens, A., Frieß, U., Todtermuschke, M., and Bdiwi, M. (2021). "Combining safe collaborative and high-accuracy operations in industrial robots," in *Advances in automotive production Technology – theory and application*. Berlin: Springer Vieweg, 451–459. doi:10.1007/978-3-662-62962-8_52

Pan, J., Chitta, S., and Manocha, D. (2012). "Fcl: A general purpose library for collision and proximity queries," in 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, United States, 3859–3866. doi:10.1109/ICRA.2012.6225337

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T. B., and Leibs, J. (2009). "Ros: An open-source robot operating system," in 2009 International Conference on Robotics and Automation Workshop Open-Source Software, Kobe, Japan.

Qureshi, A. H., Bency, M. J., and Yip, M. C. (2019). "Motion planning networks," in 2019 International Conference on Robotics and Automation, Montreal, Canada, 2118–2124.

Qureshi, A. H., Miao, Y., Simeonov, A., and Yip, M. C. (2020). Motion planning networks: Bridging the gap between learning-based and classical motion planners. *IEEE Trans. Robotics* 37, 48–66. doi:10.1109/tro.2020.3006716

Rahali, H., Zeghlache, S., and Benyettou, L. (2022). Fault tolerant control of robot manipulators based on adaptive fuzzy type-2 backstepping in attendance of payload variation, *International Journal of Intelligent Engineering and Systems*, Japan, 14 (4), 312–325.

Ratliff, N., Zucker, M., Andrew Bagnell, J., and Srinivasa, S. (2009). "Chomp: Gradient optimization techniques for efficient motion planning," in Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 489–494.

Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., et al. (2014). Motion planning with sequential convex optimization and convex collision checking. *Int. J. Robotics Res.* 33 (9), 1251–1270. doi:10.1177/0278364914528132

Schulman, J., Ho, J., Lee, A. X., Awwal, I., Bradlow, H., and Abbeel, P. (2013). Finding locally optimal, collision-free trajectories with sequential convex optimization. *Robot. Sci. Syst.* 9 (1), 1–10.

Shah, N., and Srivastava, S. (2022). "Using deep learning to bootstrap abstractions for hierarchical robot planning," in Proc. of the 21st International Conference on Autonomous Agents and Multi-agent Systems (AAMAS).

Tan, S., Yang, J., and Ding, H. (2023). A prediction and compensation method of robot tracking error considering pose-dependent load decomposition. *Robotics Computer-Integrated Manuf.* 80, 102476. doi:10.1016/j.rcim.2022.102476

Visual Components (2021). *Visual components*. Espoo, Finland. 4.3.

Zucker, M., Ratliff, N., Dragan, A. D., Pivtoraiko, M., Klingensmith, M., Dellin, C. M., et al. (2013). Chomp: Covariant Hamiltonian optimization for motion planning. *Int. J. Robotics Res.* 32 (9), 1164–1193. doi:10.1177/0278364913488805

frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 15 February 2023
DOI 10.3389/frobt.2023.1028329

OPEN ACCESS

EDITED BY
Jose Luis Sanchez-Lopez,
University of Luxembourg, Luxembourg

REVIEWED BY
Yong-Guk Kim,
Sejong University, Republic of Korea
Hang Su,
Fondazione Politecnico di Milano, Italy

*CORRESPONDENCE
Sebastian Krusche,
✉ sebastian.krusche@iwu.fraunhofer.de

SPECIALTY SECTION
This article was submitted to Robotic Control Systems, a section of the journal Frontiers in Robotics and AI

RECEIVED 25 August 2022
ACCEPTED 30 January 2023
PUBLISHED 15 February 2023

CITATION
Krusche S, Al Naser I, Bdiwi M and Ihlenfeldt S (2023), A novel approach for automatic annotation of human actions in 3D point clouds for flexible collaborative tasks with industrial robots. *Front. Robot. AI* 10:1028329. doi: 10.3389/frobt.2023.1028329

A novel approach for automatic annotation of human actions in 3D point clouds for flexible collaborative tasks with industrial robots

Sebastian Krusche*, Ibrahim Al Naser, Mohamad Bdiwi and Steffen Ihlenfeldt

Department of Production System and Factory Automation, Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany

Manual annotation for human action recognition with content semantics using 3D Point Cloud (3D-PC) in industrial environments consumes a lot of time and resources. This work aims to recognize, analyze, and model human actions to develop a framework for automatically extracting content semantics. Main Contributions of this work: 1. design a multi-layer structure of various DNN classifiers to detect and extract humans and dynamic objects using 3D-PC preciously, 2. empirical experiments with over 10 subjects for collecting datasets of human actions and activities in one industrial setting, 3. development of an intuitive GUI to verify human actions and its interaction activities with the environment, 4. design and implement a methodology for automatic sequence matching of human actions in 3D-PC. All these procedures are merged in the proposed framework and evaluated in one industrial Use-Case with flexible patch sizes. Comparing the new approach with standard methods has shown that the annotation process can be accelerated by 5.2 times through automation.

KEYWORDS

data labeling, human activity recognition, deep learning, robotics, point cloud annotation

1 Introduction

Recognition and prediction of human actions are increasingly crucial in industrial production. Flexible and agile machine systems should be able to recognize their environment, detect persons in the workspace and predict human intentions. Based on the future human action information, the machine systems adapt the production sequence in real-time and optimize the production process situationally. Such an activity prediction would allow the human to work very closely with the robot in specific product steps in collaboration without the danger of a collision. On the other hand, there is no loss of machine utilization because the system can increase the speed again when a permissible safety distance is reached (Rashid et al., 2020). By ensuring the safety, interactions such as teaching a heavy-duty robot by gestures very close to the robot can be enabled without requiring complex approvals by the operator The No-Code approach allows the human to teach the robot directly by guiding, showing, or demonstrating without the need for knowledge of complex programming languages (Bdiwi et al., 2016; Halim et al., 2022). Working directly with the robot on the production component enables the human to use his sensitive skills in very complex activities and thus drive the level of automation forward, even in non-industrial areas such as surgery (Su et al., 2021; Su et al., 2022). To make these possible, efficient algorithms are needed, that can robustly recognize and predict human behavior in all its variations. Many approaches have been implemented to deal with different video data (Baradel et al., 2016; Feichtenhofer et al., 2016; Gkioxari et al., 2017; Liang et al., 2019; Morais et al., 2020). Most convert the input video data into spatio-temporal representations and infer labels from these representations. Different types of information are used in these works, such as human posture, interaction with objects, and appearance features.

A large number of published data sets with daily and sports activities are available for the method development (Shahroudy et al., 2016; Kay et al., 2017; Carreira et al., 2018; Carreira et al., 2019; Jang et al., 2020; Liu et al., 2020; Lucas et al., 2020; Shao et al., 2020). Annotating these datasets is very time-consuming and labor-intensive. Human annotators must define and describe spatial regions associated with an image frame from a video or delineate temporal segments in conjunction with the video. Standard shapes such as rectangles, circles, points, or polygons frequently characterize the spatial regions. In contrast, marking the temporal segments requires only the start and end timestamp. These spatial regions and temporal segments are described by textual metadata.

Activities in an industrial context are usually very complex and consist of a combination of simple actions. In most cases, items are used to perform the activity. Up to that, it comes to interactions with other persons to accomplish extended, more complex action sequences. The activity duration is usually longer than 1 s, but the duration of action is mostly only up to 0.5 s (Das Dawn and Shaikh, 2016; Trong et al., 2017; Dang et al., 2021). In very rare cases there are crowds of people or crowded scene, but more often there is occlusion by industrial plant parts and machinery in the scene. Furthermore, many items, such as ladders or chairs, are often classified as humans by 3D sensor systems, depending on their shape. In addition to these static objects, there are dynamic objects such as robots or AGVs, whose position changes continuously, and these temporarily provide occlusions in the workspace.

In order to face these challenges, it is necessary to apply a 3D multi-sensor system that observes the industrial workspace from multiple perspectives and avoids the risk of occlusion. Each 3D sensor provides a 3D point cloud and an RGB image with a frame rate of 10 fps ~ 30 fps, which leads to a vast amount of data for an action sequence with a duration of about 1s, if at least 4 sensors are used. Manual annotation of action and activity sequences is impossible because of this amount of data and the complexity of such a multi-sensor system. Furthermore, manual annotation of objects in 3D space requires different modeling tools than those required for annotation 2D images.

The automatic annotation approach presented in the paper can fulfill all these requirements. The manual effort of the annotation process would be reduced to a significant amount, and training data from different perspectives can be generated due to the multi-sensor technology. By using deep learning models for skeleton-based recognition of human activities (Pavlakos et al., 2016; Barsoum et al., 2017; Ruiz et al., 2018; Li et al., 2020; Mao et al., 2020; Yuan and Kitani, 2020; Dang et al., 2021; Mao et al., 2021; Martínez-González et al., 2021), the action sequences can be classified and tracked very easily. The annotator no longer needs to focus on the elaborate annotation of the human pose and can take care of tracking multiple people.

In our work, we designed a multi-layered structure of different DNN classifiers to recognize humans and dynamic objects in the 3D point clouds of a multi-sensor system. To do this, we combined several available AI classifiers to distinguish humans from robots or other objects accurately. We developed and implemented a methodology for automatically matching human actions in 3D point clouds for human activity sequence detection. To operate these methods and verify the results, we designed an intuitive user interface that allows the user to correct the automatic annotation or improve the process by optimizing the classifiers. To finally evaluate the approach, we created extensive datasets based on empirical experiments with ten subjects performing various simple and complex activities in an industrial environment. As part of the experiments, we addressed human-robot cooperation scenarios where humans and robots coexist in a workspace very close.

2 Related work

Annotation of human activities in video data is very time and labor-intensive work. It requires a massive amount of human and hardware resources. There are two general approaches for generating data sets with human actions.

1) Data sets like NUCLA, SYSU, NTU-RGB + D, PKU-MMD (Bdiwi et al., 2016; Rashid et al., 2020; Su et al., 2021; Halim et al., 2022; Su et al., 2022) (Wang et al., 2014; Shahroudy et al., 2016; Hu et al., 2017; Liu et al., 2017; Liu et al., 2020) were generated under laboratory-like conditions, the activities were controlled, and the sensors had optimal perspectives on the scene. Based on this boundary condition, the human activity in the video sequences can be very well recognized, annotated, and quickly separated. In this case, annotation by hand is very easy and requires less effort. However, human activities' variance is minimal, meaning that the data sets do not represent reality. Performing such predefined laboratory experiments is labor intensive and time-consuming.
2) In contrast, data sets such as Fine-Gym, UAV-Human, HOMAGE (Shao et al., 2020; Li et al., 2021; Rai et al., 2021) generated in real-world environments (such as road traffic and crowds in public places) with uncontrolled action are more challenging to annotate because the environment is too cluttered, people may be obscured, camera perspectives are not optimal, or the variance of human action is too different.

Datasets like as ActivityNet, AVA, Babel (Heilbron et al., 2015; Gu et al., 2018; Punnakkal et al., 2021) have been labeled *via* commercial crowdsourcing platforms such as Amazon Mechanical Turk (AMT) (Amazon, 2021) for a charge to the dataset creators. In some cases, the annotators from crowdsourcing platforms influence the annotation quality negatively due to a lack of expertise. The crowdsourcing method may compromise confidentiality.

There are different open-source tools for annotating objects and features in image videos (da Silva et al., 2020; Dutta and Zisserman, 2019; Biresaw et al., 2016; Riegler, 2014; Yuen et al., 2009). Most of them require a manual annotation by an annotator in every frame. Only some of them provide the ability to track humans or objects across multiple image frames using tracking functions (David, 2000; Vondrick et al., 2012; Bianco et al., 2015; Intel, 2021). This feature makes it easier for annotators to save time by automatically tracking the annotations instead of labeling them frame by frame. Usually, the marking is done by an Annotator manually or by an object recognition algorithm before the tracking function tracks the object or human over several frames. In ViPER, ground truths are stored as sets of descriptors (David, 2000). Each descriptor annotates an associated range of frames by instantiating a set of attributes for that range. However, these attributes are not simple and flexible enough to annotate time-varying (appearing and disappearing) behaviors. VATIC is a simple, reusable, open-source platform for labeling research videos (Vondrick et al., 2012). To annotate human behavior, third parties have extended VATIC with additional features. iVAT (Bianco et al., 2015) presents a tool that allows the user to extract target states and categorize the targets. To significantly minimize human effort, iVAT uses automatic tracking and other computer vision methods combined with interpolation to support manual annotation. However, human interventions and verifications are necessary to validate the quality of the annotation results. JABBA (Kabra et al., 2013) is a semi-automatic machine learning-based behavioral annotator that takes the already annotated states (trajectories) as input to perform the task. The purpose of these tools is to reduce human effort and time and to preserve the annotation quality. Manual labeling effort is reduced by automatically estimating states between selected keyframes using linear interpolation and homography-preserving techniques (Yuen et al., 2009; Vondrick et al., 2012). The annotation quality of these tools depends on the individual annotators or object detectors. It is very challenging to mark objects correctly in crowded scenes, and annotators may easily miss important details. Furthermore, no other information, such as the human body pose or current activity, is provided. In addition, it is necessary to perform a pose estimation to get the skeleton data (Cao et al., 2016; Andriluka et al., 2017; Güler et al., 2018; Kocabas et al., 2018; Xiao et al., 2018; Cai et al., 2019; Cheng et al., 2019; Sun et al., 2019; Jin et al., 2020; Contributors, 2021; Kreiss et al., 2021). The HAVPTAT tool allows the annotation of body poses (“Walking”, “Standing”, “Sitting”) and simple activities (“WalkingWhileCalling”, “StandingWhileWatchingPhone”, “SittingWhileEating”) of several people over a sequence of 2D images (Quan and Bonarini, 2022). A separate algorithm OpenPifPaf (Kreiss et al., 2021), whose results are reloaded and played in parallel with the video, does the estimation of body poses. The annotator has to do the body pose assignment manually.

Besides the above drawbacks, it is challenging to perform the detection and tracking of multiple people in a video when dealing with crowded and cluttered scenes. None of the approaches uses a multi-sensor concept to fuse, and plausible the body pose estimation and tracking, which guarantees that the results can be classified and annotated much more clearly.

3 Annotation framework

The realization of the annotation approach required the development of an extensive framework, its architecture consisting of a powerful AI server and an intuitive GUI, as seen in Figure 1. The basic workflow is structured into three steps.

1) The user selects the desired action dataset *via* the GUI and passes it to the automatic annotation step, where the data is automatically segmented, tracked, and classified.
2) The raw results are then analyzed, filtered, and optimized in the following automatic post-annotation step. The goal is to determine and correct correlations based on a complete view of the entire sequence of actions.
3) In the final manual post-annotation step, the user checks whether the results are correct or whether a further manual correction is necessary, based on the visualization that displays the annotation results in the context of the 3D point clouds and RGB images.

The development and implementation of this structure were realized using the OpenCV library (Bradski, 2000) for the backend framework, VTK library (Schroeder et al., 2006) for 3D visualization and the Qt library (Nasdaq Helsinki: QTCOM, 2021) for the GUI. The integration of the open-source AI algorithms was done using Python since most approaches are based on AI frameworks PyTorch (Paszke et al., 2019) or Tensorsflow (Abadi et al., 2021), which are implemented in the Python programming language. Each AI classifier is encapsulated in a Docker container to avoid possible inferences between the specific software dependencies. The Docker containers are run on a separate Linux-based AI server with two parallel GPU cards to enhance performance and ensure parallel usage. A ZMQ interface based on UDP is implemented between the Annotation Viewer and Docker Container on the AI Server for data exchange (Hintjens, 2011).

3.1 Data structures

Because of the high raw data volume of the multi-sensor system and the output of the multi-person tracking, it is necessary to structure the data so that the system can clearly distinguish between input data and the intermediate and end results. For this purpose, a data structure with specific data types was developed, which lists all data and information types and allows the user to select the available visualization option. The data structure differs in two basic categories, which are specified as follows:

Streams: The Streams category contains all image and 3D point cloud data sets of the multi-sensor system for the entire acquisition time of an action sequence, forming the basis for the automated annotation process. One stream includes the acquired sensor data of the scene from the perspective view of the single sensor. The intrinsic

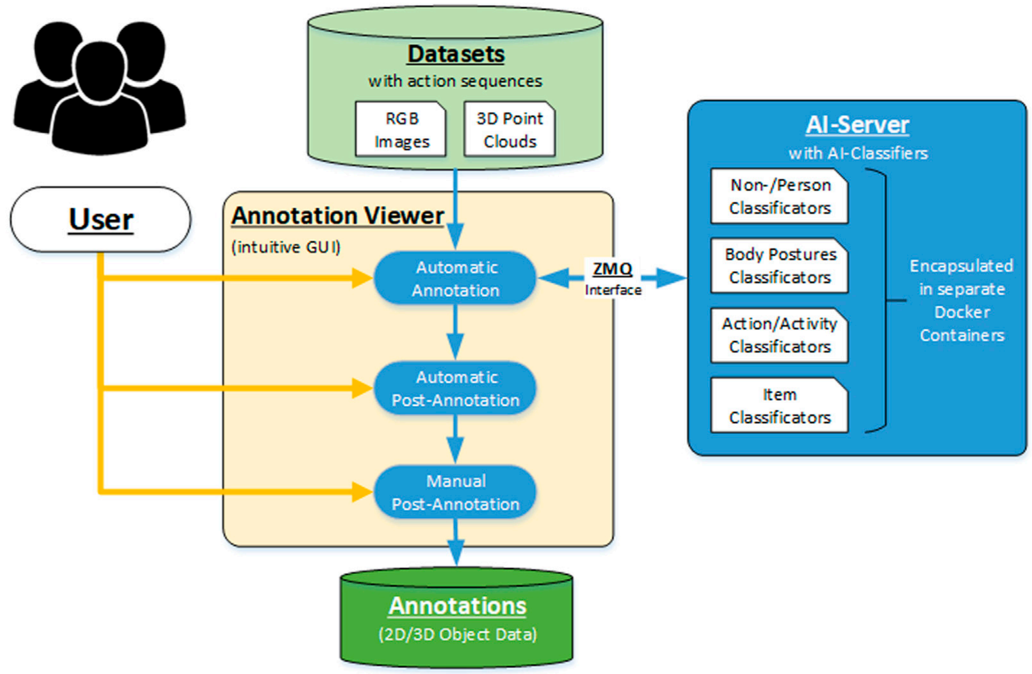
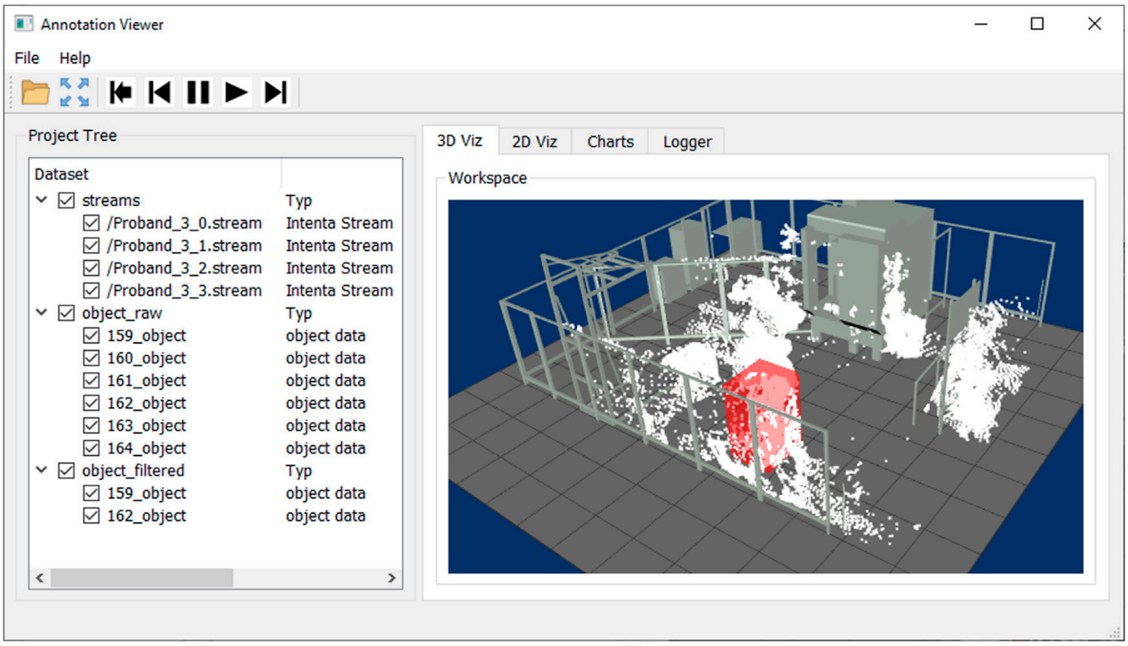


FIGURE 1
Schematic chart of the annotation framework.



Dataset	
☑ streams	Typ
☑ /Proband_3_0.stream	Intenta Stream
☑ /Proband_3_1.stream	Intenta Stream
☑ /Proband_3_2.stream	Intenta Stream
☑ /Proband_3_3.stream	Intenta Stream
☑ object_raw	Typ
☑ 159_object	object data
☑ 160_object	object data
☑ 161_object	object data
☑ 162_object	object data
☑ 163_object	object data
☑ 164_object	object data
☑ object_filtered	Typ
☑ 159_object	object data
☑ 162_object	object data

FIGURE 2
Graphical user interface for displaying sensor data sets and annotation results.

and extrinsic calibration parameters of every single sensor are required to establish the relation between the 2D RGB camera images, the 3D point clouds, and the sensor world coordinate system. In addition to the reference to the sensor world coordinate system, the reference to the world coordinate system should also be given for a holistic view of the scene from different perspectives.

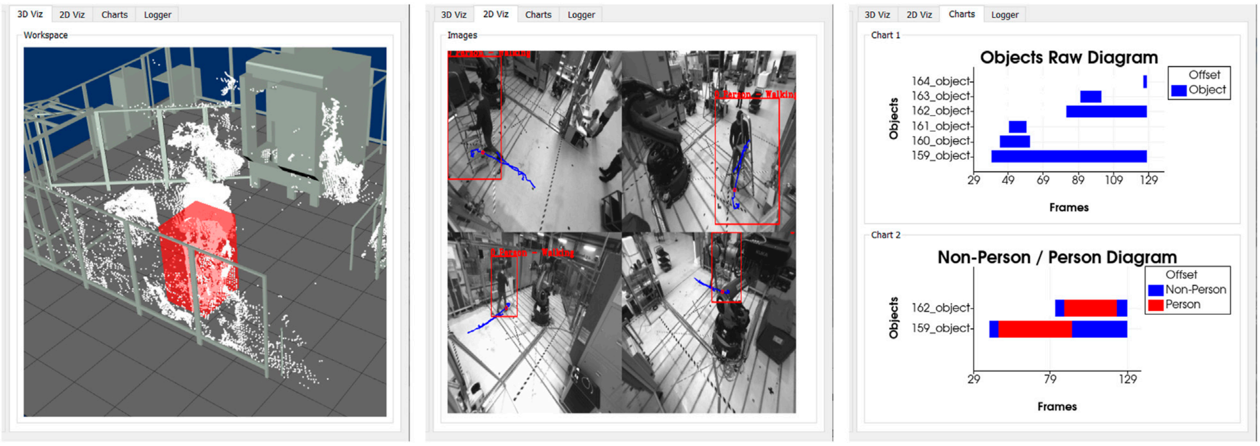


FIGURE 3
Workspace viewer (left), image viewer (center), chart viewer (right).

Object List: This category allows the summary of all objects detected, tracked, and classified throughout the action sequence. Besides the result representation of the automatic annotation process, this structure is also used for the following steps of automatic post- and manual annotation. The list contains a separate data set (object data) for each object, which enables a view of the temporal and spatial movement concerning the whole sequence for the single object. The dataset contains spatial information in the form of 2D and 3D bounding boxes and the results of person/non-person classification and human posture estimation for each frame, respectively. Furthermore, global information about the object is also stored, such as walking paths or the execution location of the action.

Annotation Data: For training and verification of AI algorithms for action recognition and action prediction, datasets are required that represent human actions in a temporal context. The annotation sets include point clouds and image patches with corresponding labels related to human action, generated automatically based on object data and specifications by the user.

In the user interface, the loaded and generating data structures are visualized in the form of a tree model, shown in Figure 2 on the right side of the user interface. The user can quickly distinguish between raw data (streams), object data (object list), and annotation data based on the structure and select the corresponding visualization forms *via* the checkboxes.

3.2 Annotation viewer

An intuitive user interface has been designed and implemented to merge annotations and raw data sets, allowing the user to review, verify and adjust the action sequences. In the interface, the 2D image data, 3D point clouds, and tracking and classification results of an action sequence can be visualized in correspondence to each other frame by frame. Various display forms are implemented as separate visualizations for data representation, defined as follows, as shown in Figure 2.

Workspace Viewer: For displaying the 3D object information of the detected human to the 3D point cloud, a 3D visualization environment is implemented, representing the multi-sensor system surveillance space. The user can load and visualize 3D CAD models of the plant with the machines, robots, and protective fences to reference the sensor information to the environment, as shown in Figure 3 (left). The precondition for correct mapping of the 3D object information, 3D point clouds, and CAD models is a precise extrinsic calibration and temporal synchronization to each other. With the 3D workspace visualization, the user can quickly check and verify the results of the segmentation, tracking, and classification algorithms. In complex scenes with many dynamic objects, the information representation in 3D can be better than in 2D.

Image Viewer: Parallel to the 3D workspace visualization, the user can review and analyze the action scene from every single sensor perspective of the multi-sensor system. For this purpose, the RGB images are visualized in combination in a separate tab, as shown in Figure 3 (center). In addition to the perspective view of the scene, the results of the tracking and classification are plotted in the individual images. The detected persons are marked by a 2D bounding box and a label so that the persons can be identified over several frames. Furthermore, additional information about the course of action can be displayed, such as the entire walking path or the location where the action was performed. It is important to note that the detected person may be covered due to the sensor perspective, resulting in an incorrect display.

Chart Viewer: To summarize the whole sequence of actions, the temporal series of the objects are displayed in the form of bars on a time axis. The diagrams are available in a third tab, as seen in Figure 3 (right). In the Raw Object chart, all detected objects in the action sequence are displayed, allowing the quality of the segmentation, and tracking to be evaluated. The algorithm could not accurately segment and track the objects over the entire sequence if there are many objects with short time segments. The second chart details the objects by person and non-person. The user can see here which objects can be assigned to the acting persons or interaction objects and which objects are misdetections.

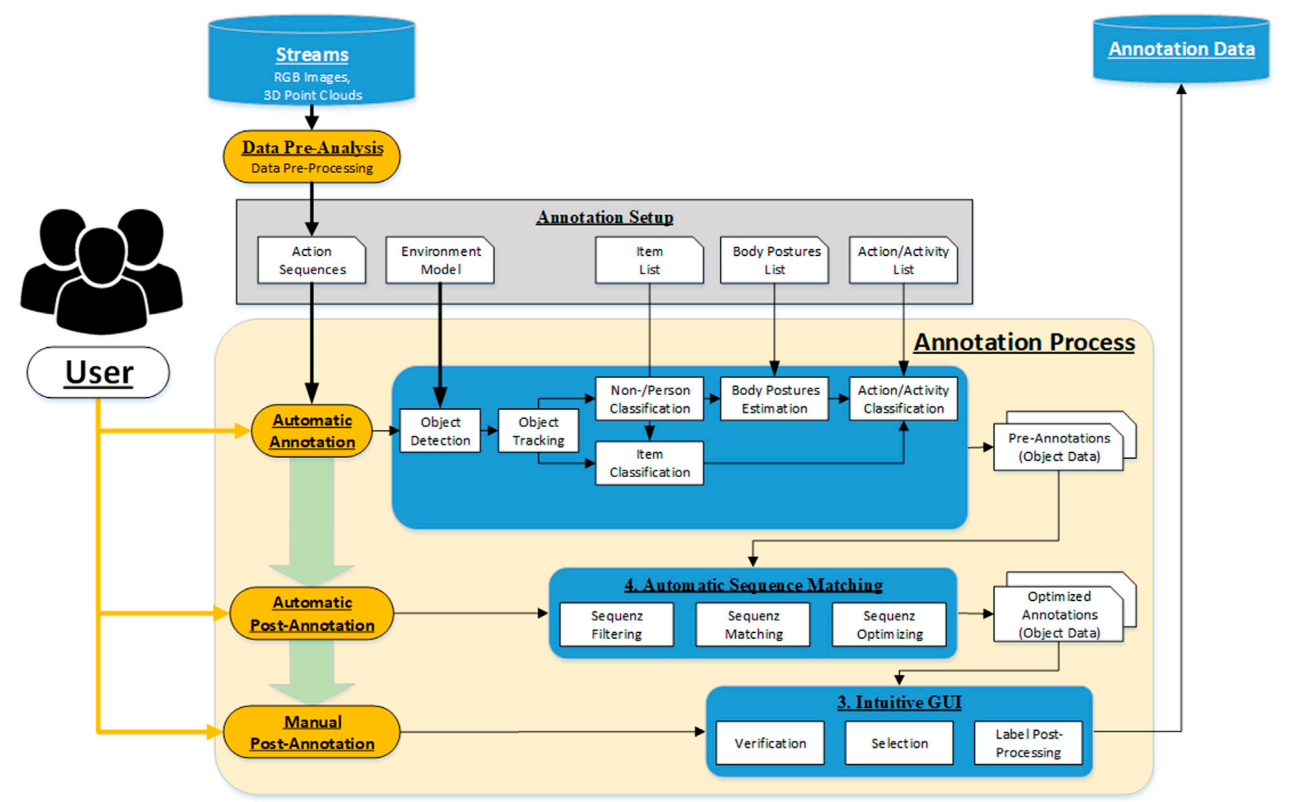


FIGURE 4
Schematic workflow of the annotation approach.

Project Tree: In general, to give the user an overview of the data being loaded or generated, the data structure is visualized as a dynamic tree model, as shown in Figure 2 on the right side of the user interface. In addition to the listing of the individual sensor data sets and the detected objects of the action sequence, the data types described in Section 3.1 are also displayed. By using a dynamic tree model, additional data, information, and control elements can be added or adapted very quickly, making the user interface flexible depending on the size and type of the data set.

Data Logger: A text browser was placed in an additional tab to display status messages or system information on the annotation process. The user can check which current step the process is or which files have been loaded.

The user can replay the action sequence in its full context through the intuitive user interface or analyze the scene in more detail frame by frame. There are also functions for importing raw data, exporting annotation results, and printing analysis results as diagrams.

3.3 Annotation approach

A workflow approach was developed to ensure that the automatic annotation tool generates action- and context-based annotation data consistent with temporal and spatial relationships. Figure 4 shows the schematic context of the annotation approach. The single steps are explained in detail in the following subsections.

The minimum requirement for the automatic action annotation is that each dataset contains a complete action or a complex activity with a series of actions of at least one person. This action or activity should be represented as a sequence of corresponding RGB images and 3D point clouds in the dataset. A further advantage is if multiple sensors from different spatial perspectives capture the scene with the action sequence time-synchronously. The condition for processing these multi-recordings is that the setup between the sensors is known. The extrinsic parameters must be precisely determined to merge the 3D point clouds of the individual sensors and to ensure the assignment of the detected persons and items. The action sequences must be loaded into the annotation tool as a complete, time-synchronous data set to guarantee successful processing and annotation.

Data pre-analysis: The tool treats each annotation of an action dataset as a session, whose results are stored and evaluated separately. After the action sequence is loaded into the annotation tool, a pre-data analysis is performed to check if there is an equal number of RGB images and 3D point cloud or if there is a large number of frame drops. In case of incompleteness or low quality, the data set must be discarded or cropped so that a successive action annotation is possible. In addition, data pre-processing can also be optionally performed, such as the rectification of RGB images or the conversion of depth images from 3D point clouds.

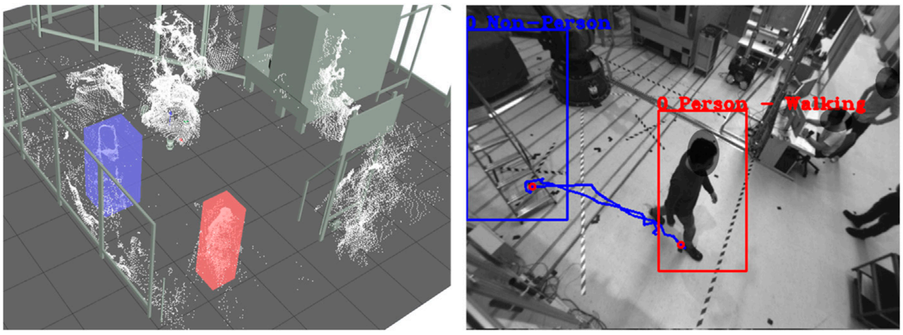


FIGURE 5
Results of the automatic annotation step visualized in the 3D workspace (left) and RGB image (right) of Sensor 1. People are marked with (red) and objects with (blue).

Annotation Setup: At the beginning of the annotation session, the user has to provide additional information regarding the action, the environment, and the interaction objects besides the raw data set. The environment model contains all spatial information regarding the action, like where the action takes place, what are possible accesses or walkways in the monitoring area or where are the interaction objects placed, etc. For this purpose, the corresponding objects or regions will be defined using standard shapes such as rectangles, circles, points, or polygons in a predefined XML format. The action inference is based on body posture estimation and object interaction recognition. It is necessary to define the common postures and objects and to submit them as lists to the tool. The final action labeling is done by specifying the action sequence or the action type, which is to be predefined by the user as an action/activity list. In this main list, links are used to refer to the information in the sub-lists. It results in a tree structure with different levels, describing the expected action in detail and forms the basis for the automated annotation algorithm. The user must configure all this information utilizing a parameter catalog and provide it using parameter files.

Automatic Annotation: After all boundary conditions regarding the annotation task have been set, and the pre-analysis of the datasets is positive, the datasets can be passed to the automatic annotation step. In the beginning, static objects and environment structures must be removed from the 3D Point Cloud. For this purpose, the background segmentation method is used, which removes all 3D points from the point cloud that are not included in the static background model. This background model should be learned for each scene so that all static objects and environmental structures are precisely removed from the 3D point cloud. A prerequisite is that no dynamic object is in the field of view of the sensors during the teach-in or that there is no further change in the working space of the system. After the segmentation of the static background in the 3D point cloud of the current frame follows the segmentation of all dynamic objects in the sensor's field of view. All 3D points with a certain Euclidean distance are combined into a cluster, separated from the remaining part of the point cloud by a 3D bounding box, and declared as an object. These segmented objects are then tracked over single frames using Kalman filters until they leave the field of view or the data set with the action sequence is finished. The result of the 3D object segmentation can be seen in Figure 5 (left). Due to background segmentation and point cloud fusion, dynamic objects can be segmented very well from the point cloud.

In order to enable the classification of the segmented objects in the RGB images, it is necessary to project the 3D bounding box from 3D space into the 2D camera plane of the single sensors. For this purpose, the 3D object must be transformed from the world coordinate system into the sensor coordinate system using extrinsic parameters and then projected into the sensor plane using intrinsic sensor parameters. The 3D to 2D projection result can be seen in Figure 5 (right). The object can be segmented from the rest of the RGB image using the 2D bounding boxes. Based on the segmentation, human and object classification in 3D/2D is feasible. The extracted 3D and 2D patches are transferred to the AI server *via* the ZMQ interface. Various AI classifiers such as OpenPose (Cao et al., 2018), Alpha Pose (Li et al., 2018), or DarkNet (Redmon and Farhadi, 2018) can distinguish non-persons from persons or identify specific items. Once a person has been confidently classified, human body pose estimation is performed based on the skeleton model of the OpenPose and AlphaPose classifiers. With the help of the estimator, human postures such as walking, standing, sitting, bending, and kneeling can be detected and additionally used later to generate action-related annotation data.

Automatic Post-Annotation: The objective of the automatic post-annotation is to view the results of the automatic annotation step over the entire sequence of actions and the overall workspace. Input is the classified objects whose distribution over all sequence frames can be seen in the form of a bar chart in Figure 6 (left). Mainly there are two large objects with a history spanning several frames, from which the plot can be derived. All other objects are short-lived and have been classified as undefined. The appearance can be reduced to artifacts in the point cloud, which are caused by an asynchrony of the sensor data.

The second step of Automatic Post-Annotation includes filtering objects according to different criteria. Besides undefined objects and objects with a small lifetime, all objects

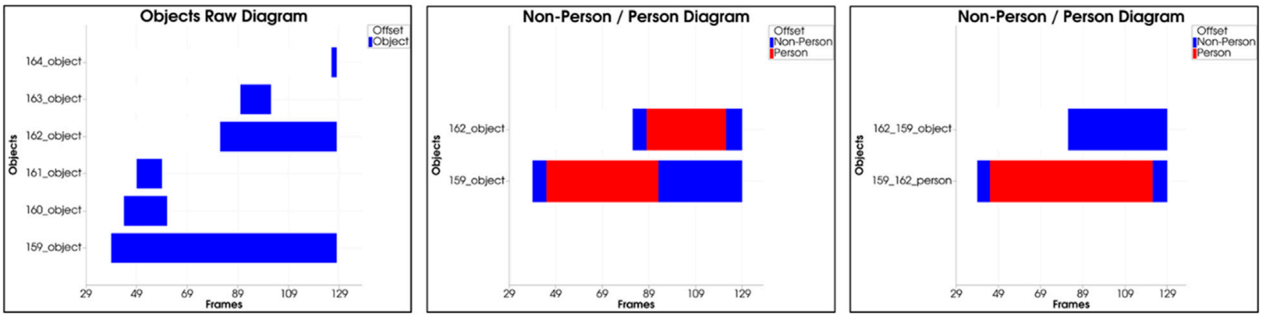


FIGURE 6
(left) Input: distribution of raw objects over the entire sequence, (center) Filtering: removal of all small undefined objects, (right) Optimization: object decompensation and reordering by Non-Person/Person.

with small sizes are removed. The threshold value here is set to a length of 0.3 m per bounding box edge. The object is filtered out if all three edges are below this threshold. The result of the filtering is Figure 6 (center). All undefined objects with a too short lifetime were removed.

After the list of objects has been roughly filtered, the single objects are analyzed in detail. It is examined whether the classification results of the objects are constant over the entire sequence or whether there is a relationship between them. Besides the classification results, the motion sequence in 3D space and the object's volume is another input for detailed investigation. In the present case, the second object results from the first object, only that the class assignment is incorrect. To correct this, both objects are decompensated in the third stage, rearranged, and reassembled. The result is shown Figure 6 (right). The reordering shows that both object courses are consistent, and a clear distinction between non-person and person can be made.

Manual Post-Annotation: Finally, after the object data has been generated and optimized, the final step is verifying and selecting the annotation data to be exported. The user reviews the results over the entire sequence of actions using the 3D Workspace Viewer, 2D Image Viewer, and Chart Viewer by replaying the data or examining it frame by frame. Objects can be manually removed if it becomes apparent that the data has been incorrectly segmented, classified, and mapped. Furthermore, in case of inaccurate results, it is desirable to adjust the segmentation, tracking, or classification parameters and repeat the automatic annotation process.

The automatic annotation approach aims to generate specific action and contextual object data whose spatial and temporal changes are coherent. It ensures that the segmented, tracked, and classified objects are based on the 3D/2D sensor data corresponding to the natural dynamic objects represented by an object or a person. The user can automatically create training and verification datasets based on consistent object data for AI algorithm development. For that purpose, the tool automatically extracts the point cloud or image patches of the selected object or person with the corresponding label using the 3D and 2D bounding boxes.

4 Experiments

4.1 Experiment setup

The performance of the annotation tool was examined and verified in 6 specific test scenarios according to different criteria. In these scenarios, a test person performs different complex activities ranging from actions such as walking with an object to interactions between two people. The test scenarios cover expected human behaviors in industrial activities in the automotive industry. One of the first actions in the event of malfunctions in industrial robot systems is usually for the worker to enter the robot cells to clear a fault. Therefore, almost every test scenario includes walking with or without an object.

Table 1 summarizes the 6 scenarios with the main actions and the number of persons. The test scenarios range from very simple to complex. Depending on the sequence, one or more persons are in the robot cell, interacting with objects such as ladders, suitcases, or transport carts. The test scenarios cover expected human behaviors in industrial activities in the automotive industry. One of the first actions in the event of malfunctions in industrial robot systems is usually for the worker to enter the robot cells to clear a fault. Therefore, almost every test scenario includes walking with or without an object.

The sensor data is collected in the HRC cell at Fraunhofer IWU, whose design corresponds to a robot cell without a protective fence in the industrial production environment. The interior of the cell is large and barrier-free and allows human activities and interactions in the robot environment. The open area of the cell allows for the optimal alignment of sensor technology to the scenery and a promising field of view without being obscured by additional machine or plant parts. Based on the described infrastructure, the access area and the front working area of the cell were selected for data acquisition of the multi-sensor system. The front view of the HRC cell can be seen in Figure 7 (Left). In order to completely cover the workspace and to avoid obscuring the person acting,

TABLE 1 Test scenarios to investigate the performance of the annotation tool.

Scenario-Nr.:	Scenario title	Action types	Active subjects
1	Person walks into robot cell	Standing (static), walking	1
2	Person walks with item	Standing (static), walking, setting up ladder	1
3	Person pushes a transport cart	Standing (static), walking, pushing transport cart	1
4	2 persons walk into robot cell	Standing (static), walking	2
5	2 persons hand over an item	Standing (static), walking, handing over item	2
6	2 persons with a transport cart	Standing (static), walking, Pushing transport cart	2

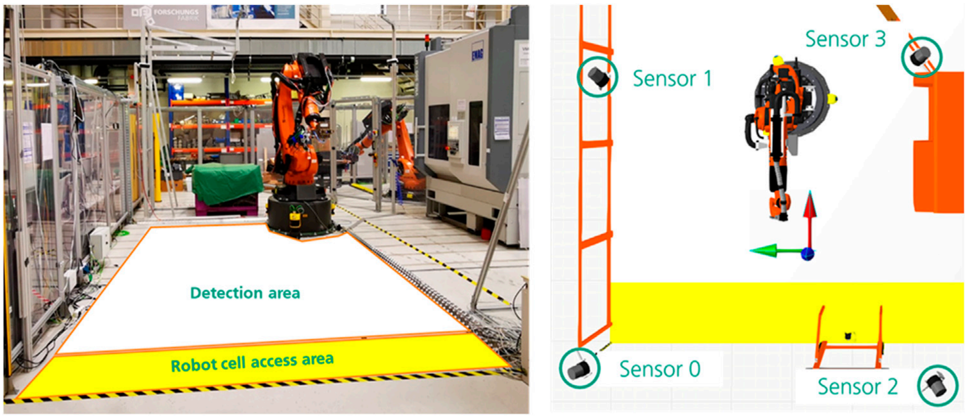


FIGURE 7
(Left) Front view of HRC test cell at Fraunhofer IWU, (Right) Sensor layout with 4 sensors.

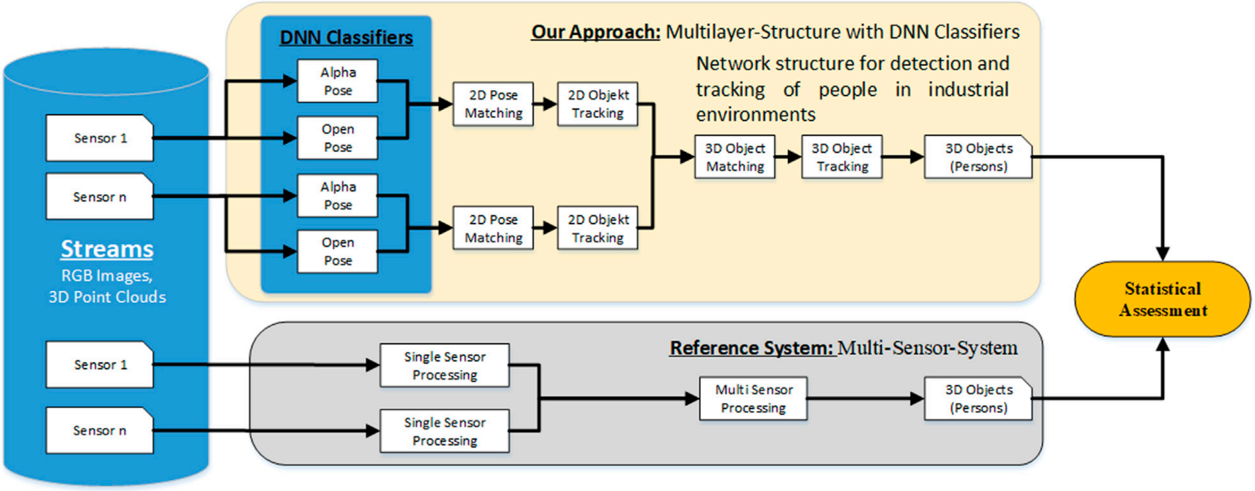


FIGURE 8
Architecture for comparing Results of Annotation Tool and Reference System.

four sensors were installed in the room, which records the scene from various perspectives. The sensor layout can be seen in Figure 7 (right).

In the execution of the experiment, data sets with 10 different test subjects with five scenarios each were created based on the test description. Because three scenarios were performed in

TABLE 2 Final summary of the statistical evaluation.

Dataset information					3D person tracking (average number of processed objects)			Reference (average number of processed objects)			
Scenario Number	Number of Datasets	Scenario Mode	Number Subjects	Frames	Total	Valid (Person)	Invalid	Raw	Filtered	Person	Non Person
Scenario 1	11	single	1	167	4,0	1.5	2.5	1.5	1.1	1.1	0.0
Scenario 2	10	single	1	182	3.9	1.4	2.5	4.8	1.8	1.4	0.4
Scenario 3	10	single	1	236	8.4	1.9	6.5	5.3	2.0	0.7	1.3
Scenario 4	9	multi	2	179	8.6	2.9	5.7	2.8	2.2	2.2	0.0
Scenario 5	8	multi	2	182	10.4	3.5	6.9	4.8	2.5	2.3	0.3
Scenario 6	8	multi	2	259	13.8	3.8	10.0	7.9	3.3	1.8	1.6

cooperation with another subject, it is possible that in some data sets, the same subject interacted several times or that the test series of some subjects were combined. The data sets were divided into single- and multi-subject groups for further processing and analysis. Because of the experimental scenarios with two subjects, the number of data sets in both groups is not the same. The recordings resulted in 31 data sets for single-person and 27 for multi-person scenarios.

4.2 Statistical assessment

Based on the data collection, an extensive static study was conducted to demonstrate the performance of the automatic annotation tool. The purpose of this evaluation was to determine how accurately the Multilayer Structure of the various DNN classifiers can classify and track human subjects in 2D and 3D. For comparison, result data from a Multi-Sensor Reference System was used, which provides similar results in 3D space. The Architecture used for the evaluation is shown in Figure 8. The reference system works with the same sensor data as the automatic labeling tool but is limited in its function, the evaluation point cloud data. The recognition of additional objects is not provided, which is necessary for the further development and annotation of more complex scenarios. Therefore, only the recognition of persons is used to compare the systems.

The execution of the tests was automated so that the data sets were replayed. Based on the image data and point clouds, the algorithms classified and tracked people and objects in parallel. Each frame counted the number of objects for each processing step, and the intermediate results were stored.

Supplementary Appendix S1 shows the complete evaluation for scenario 1 with all four sensors. For traceability, the entire process was broken down into individual processing steps, and the number of input and output data was listed for each step. Each row of the table represents the summarized evaluation of a data set. The individual cells represent the cumulative number of input and output data of a processing step. The unit of the cell values is the number of objects processed during the data set's application. At the table's end, each column's mean and median is calculated for comparison against the other scenarios.

Supplementary Appendix S2 summarizes the results of the separate evaluation of all six scenarios and is directly compared with the reference system. In the beginning, average values of the individual processing steps for each sensor are listed, which are then merged in the multi-sensor fusion part. Based on this compact representation, anomalies and high error rates of the individual processing steps can be detected depending on the complexity of the scenario.

The results of the automatic annotation tool and the multi-sensor reference system for each scenario were summarized in Table 2 for the final evaluation of the statistical assessment. The values in the cells are the average number of processed objects. The crucial columns (marked in green) for comparing the systems reflect the number of detected persons. The average values show no significant large differences between the systems. In the case of the more complex scenarios, the number of detected persons is higher for the entire scenario, which is not necessarily due to incorrect classification. Instead, these differences can be attributed to tracking errors or random persons at the edge of the test environment, such as the recording supervisor. For a further analysis of the error causes, the data sets must be looked through randomly. For this purpose, various functions for replaying the data sets are provided in the GUI of the annotation tool. The qualitative assessment section will provide a detailed description of the classification and tracking errors.

4.3 Qualitative assessment

The recorded data sets were then processed with the developed annotation framework. Based on the results, an initial qualitative assessment of performance can be made. Basically, it can generally be concluded that the distinction between non-person and person is accurate in 90% of the scenarios. However, in the case of more complex actions, this leads to inaccurate tracking and classification results. Table 3 summarizes the most significant assessment for the corresponding scenario.

TABLE 3 Summary of the qualitative assessment.

Scenario-Nr.:	Scenario title	Segmentation and tracking	Classification
1	Person walks into robot cell	Segmentation and Tracking is correct	Classification is correct
2	Person walks with item	Object ID of human changes to ladder and human is recognized as new object (Figure 9)	Ladder is recognized as a person if the bounding box includes the person (Figure 10)
3	Person pushes a transport cart	Person is not detected in the point cloud (Figure 11)	Person is classified as non-person because only the transport cart is segmented (Figure 11)
4	2 persons walk into robot cell	Segmentation and Tracking is correct	Classification is correct
5	2 persons hand over an item	Segmentation and Tracking is correct	Classification is correct
6	2 persons with a transport cart	Change of object IDs when turning and handing over the transport cart	Transport cart is recognized as a person if the bounding box includes the person

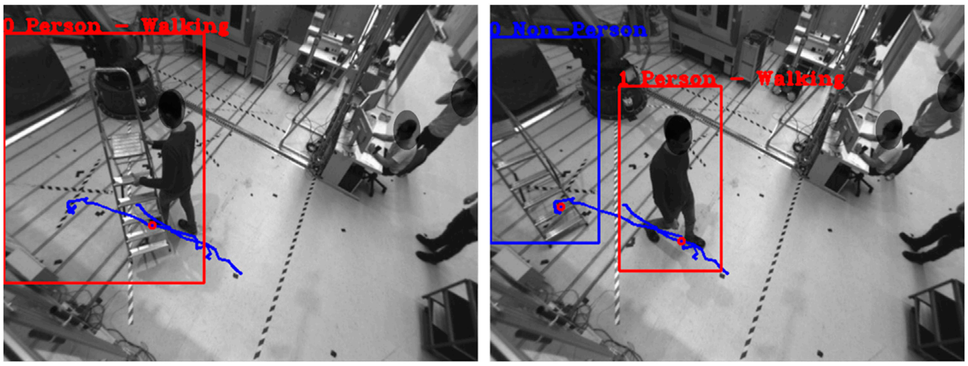


FIGURE 9
Object ID of human changes to ladder and human is recognized as new object: (Left) human and ladder interact before ID change, (Right) human and ladder separate after ID change.

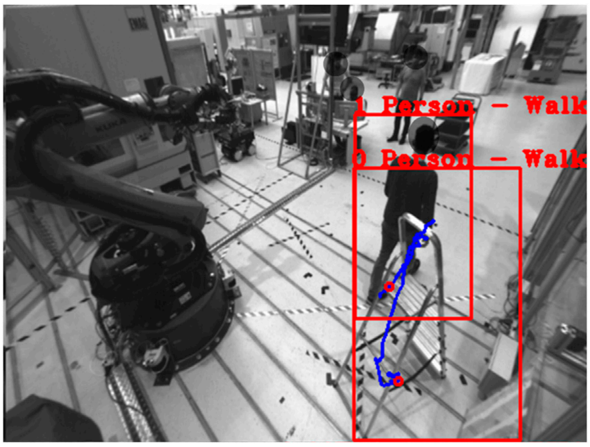


FIGURE 10
Ladder is recognized as a person if the bounding box includes the person.

4.4 Manual annotation vs. automatic annotation

For a direct comparison of the automatic annotation *versus* manual annotation, the elapsed times of the individual processing steps during the automatic run were recorded and accumulated for all data sets. Table 4 summarizes the results according to the scenarios. The values from the columns for 2D Pose Estimation refer to the processing of the entire image by the pose classifier because the time is independent of the number of objects. The remaining values in the table always refer to the elapsed time per automatically annotated object. Table 4 shows that the classification of the human pose takes the most time. The times per image are about 84 ms for OpenPose and about 94 ms for AlphaPose. In contrast, matching the 2D poses takes only a short time of about 2 ms per object. The projection of the 2D results into the 3D space requires an average time of 17–35 ms. The reason for this is the additional object segmentation in the 3D point cloud, which lead to different times depending on the

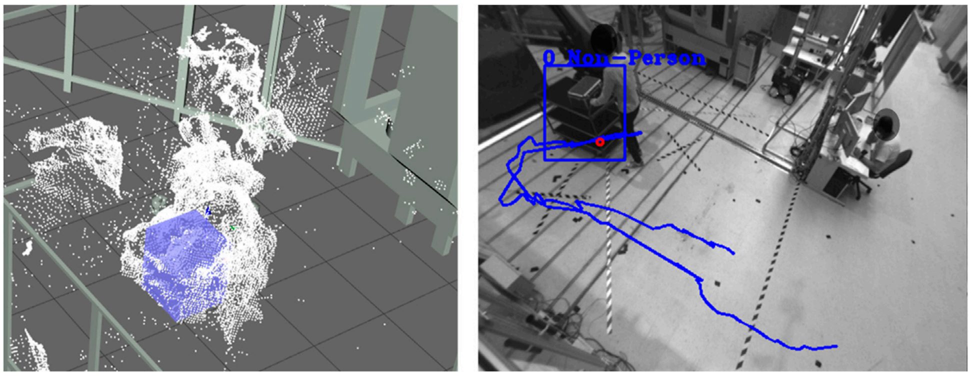


FIGURE 11
(Left) undetected Person in point cloud, (Right) unclassified Person in RGB Image.

TABLE 4 Summary of the average time measured for each processing step per scenario.

Dataset information					Elapsed time in ms					
					2D pose estimation		2D person matching	3D projection	3D object fusion and tracking	Total time
Scenario number	Number of datasets	Scenario mode	Number subjects	Frames	Open pose	Alpha pose				
Scenario 1	11	single	1	167	84.0	92.9	0.9	25.6	2.3	205.6
Scenario 2	10	single	1	182	83.9	92.7	1.3	37.2	3.1	218.2
Scenario 3	10	single	1	236	84.0	93.2	1.4	30.9	3.0	212.5
Scenario 4	9	multi	2	179	83.7	95.4	1.4	18.7	5.4	204.7
Scenario 5	8	multi	2	182	83.8	95.6	1.1	17.5	3.1	201.1
Scenario 6	8	multi	2	259	83.9	95.6	1.2	19.4	3.1	203.2
									Mean	**207,6**

Bold values are the average time over all scenarios.

TABLE 5 Qualitative comparison of the presented automatic annotation approach with standard methods.

	Two-stage approach (Gygli and Ferrari, 2020)	Box & speak (Gygli and Ferrari, 2020)	Ours (DNN classifiers + human verification (Papadopoulos et al., 2016))
Time/box	12.5 s	6.5 s	2.4 s (0.207 s + 2.2 s)
Acceleration of our approach compared to standard methods	x5,2	x2,7	-

object's size. Similar to matching the 2D poses, the 3D object tracking requires a short time of about 3 ms. In total, the average elapsed time per object is 207 ms.

We use previously confirmed results from standard methods for labeling images with object bounding boxes (Russakovsky et al., 2014; Kuznetsova et al., 2020) or outlines (Lin et al., 2014) to evaluate manual annotation, which is typically done in two steps. In the first stage, annotators are asked to mark the presence or absence of object classes in each image. In the second stage, the annotators draw 2D bounding boxes corresponding to the class labels in the image to segment the object. Another approach to fast annotation uses speech and mouse interaction. By combining them, the annotator can simultaneously draw a bounding box around the object and specify its class by speech (Gygli and Ferrari, 2020). A qualitative comparison is shown in Table 5 to estimate how efficient the automatic annotation approach is. The values for both standard approaches were taken from the existing publication (Gygli and Ferrari, 2020) and were not quantified in an experiment. For the estimation of the manual verification, as provided in the approach, a time of 2.2 s was chosen, which was taken from the publication

(Papadopoulos et al., 2016). The proposed automatic annotation approach can be estimated to be x5.2 faster in providing the class and bounding box, including human verification, than the two-stage approach.

5 Conclusion

We introduced the automatic annotation framework, an approach capable of cost-effectively generating high-quality annotations for 3D multi-sensor datasets with complex action sequences.

1) Our work focused on designing a multi-layer structure with various DNN classifiers to detect humans and dynamic objects using 3D point clouds. The action sequences can be classified and tracked very easily by using deep learning models for skeleton-based human activity recognition. The annotator no longer needs to focus on the complex annotation of the human pose and can take care of tracking multiple people.
2) The empirical experiments with more than 10 subjects to capture datasets of human actions and activities in an industrial environment allowed us to have a reasonable basis for developing and verifying the whole annotation framework. The various complex scenarios allowed us to specify the requirements for the annotation tool very well.
3) By developing an intuitive graphical user interface (GUI), the user gets a tool to verify and correct the results of the automated annotation process. The annotated action sequences can be referenced over the entire sequence or frame by frame using various 3D and 2D visualizations.
4) The design and implementation of a methodology for automatic matching of human actions in 3D point clouds enable the automatic correction of tracking and classification errors resulting from the multi-layer structure. The decompensation and rearrangement by non-person/person ensure that 3D objects are consistent.

A limitation of the approach is the presence of dynamic, non-human objects such as robots or AGVs, which may need to be clarified parts of the scene or lead to incorrect recognition. Implementing additional AI classifiers or creating a complex kinematic model for contextualization is necessary, especially when annotating human-robot cooperation scenarios where humans and robots work very closely together. This weakness needs to be compensated in the future by using robotic AI classifiers that extract accurately from the scene. To make the approach robust against the described errors in Section 4.3. Qualitative Assessment, several optimizations and tunings are required, which are prioritized as follows.

1) 3D point cloud segmentation: This requires accurately examining the sensor data for errors such as missing 3D points or adjusting segmentation parameters to ensure the segmentation of finer objects.
2) Tracking behavior: In addition to segmentation results, classification results such as human body pose should also be included in tracking to ensure that objects from the previous frame are correctly assigned to the current frame.
3) Multi-layer structure: by using additional DNN classifiers, additional object features should be detected, such as whether the person is carrying or holding an item.
4) Methodology for automatic sequence matching: In addition to the classification results, the interaction with the environment should also be considered, e.g., whether the action is performed at a specific location.

By using the tool, records from multi-sensor systems can be processed synchronously to detect and track the activity of acting individuals seamlessly. Observation from multiple perspectives creates the advantage of having sufficient samples of the human from various views in the sets of annotation data, ensuring that the AI being trained covers a high variance of human behavior. By using multi-modal data such as RGB images and point clouds from multiple sensors, a larger workspace can be covered, and tracking of multiple people can be guaranteed throughout the activity. Especially in human-robot cooperation, where safety has to be ensured during direct interaction in very confined spaces, annotating multimodal sensor data observing a scene from multiple perspectives can lead to a significant optimization of the database for training AI classifiers. Furthermore, through fusion and synchronization, annotation of the multi-modal data in 2D and 2D is possible. The annotation framework was developed to speed up the process of annotating action records and reduce the manual task of the annotator. The proposed approach can accelerate the process by up to x5.2 through automation. The tool is intended to shift the focus from viewing single images to viewing the whole scenario and include the interaction with the environment during the action. This paper is intended to stimulate the creation of more large action datasets and lead to innovations in data-driven computer vision in the coming years.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Ethics statement

Ethical review and approval was not required for the study involving human participants in accordance with the local legislation and institutional requirements. Written informed consent to participate in this study was not required from the participants in accordance with the national legislation and the institutional requirements.

Author contributions

SK: Conceptualization, Methodology, Development, Experiments, Assessments and Writing. IA: Conceptualization, Supervision and Technical Supervision. MB: Conceptualization, Supervision, Reviewing and Editing. SI: Supervision and Reviewing.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/femat.2023.1061269/full#supplementary-material

References

Abadi, M., Agarwal, A., Chen, Z., and Citro, C. (2021). TensorFlow: Large-Scale machine learning on heterogeneous systems. Available at: https://www.tensorflow.org/.

Amazon (2021). Amazon mechanical turk (MTurk). Available at: https://www.mturk.com/ (accessed Jul 20, 2022).

Andriluka, M., Iqbal, U., Milan, A., and Gall, J. (2017). PoseTrack: A benchmark for human pose estimation and tracking: arXiv.

Baradel, F., Wolf, C., Mille, J., and Taylor, G. W. (2016). Glimpse clouds: Human activity recognition from unstructured feature points. ArXiv. Available at: http://arxiv.org/pdf/1802.07898v4.

Barsoum, E., Kender, J., and Liu, Z. (2017). "HP-GAN: Probabilistic 3D human motion prediction via gan,"arxiv. doi:10.48550/arXiv.1711.09561

Bdiwi, M., Rashid, A., and Putz, M. (2016). "Autonomous disassembly of electric vehicle motors based on robot cognition," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May. 2016, 2500–2505.

Bianco, S., Ciocca, G., Napoletano, P., and Schettini, R. (2015). An interactive tool for manual, semi-automatic and automatic video annotation. *Comput. Vis. Image Underst.* 131, 88–99. doi:10.1016/j.cviu.2014.06.015

Biresaw, T., Habib Nawaz, T., Ferryman, J. M., and Dell, A. I. (2016). "ViTBAT: Video tracking and behavior annotation tool," in 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 295–301.

Bradski, G. (2000). *The OpenCV library*. Dr. Dobb's Journal of Software Tools.

Cai, Y., Ge, L., Liv, J., and Cai, J. (2019). "Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2272–2281.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y., (2016). Realtime multi-person 2D pose estimation using Part Affinity fields: arXiv.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2018). "OpenPose: Realtime multi-person 2D pose estimation using Part Affinity fields,"Available at: http://arxiv.org/pdf/1812.08008v2.

Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., and Zisserman, A., "A short note about kinetics-600," ArXiv, abs/1808.01340, 2018.

Carreira, J., Noland, E., Hillier, C., and Zisserman, A., "A short note on the kinetics-700 human action dataset," ArXiv, abs/1907.06987, 2019.

Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., and Zhang, L. (2019). HigherHRNet: Scale-Aware representation learning for bottom-up human pose estimation: arXiv.

Contributors, M. (2021). OpenMMLab pose estimation toolbox and benchmark. Available at: https://github.com/open-mmlab/mmpose (Accessed Jul19, 2022).

da Silva, J. L., Tabata, A. N., Broto, L. C., Cocron, M. P., Zimmer, A., and Brandmeier, T. (2020). "Open source multipurpose multimedia annotation tool," in image analysis and recognition," in International Conference on Image Analysis and Recognition, 356–367.

Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G., and Msr-Gcn (2021). Multi-Scale residual graph convolution networks for human motion prediction. arxiv. Available at: http://arxiv.org/pdf/2108.07152v2.

Das Dawn, D., and Shaikh, S. H. (2016). A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector. *Vis. Comput.* 32 (3), 289–306. doi:10.1007/s00371-015-1066-2

David, S. (2000). "Doermann and David Mihalcik, "Tools and techniques for video performance evaluation," in Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, 167–170.

Dutta, A., and Zisserman, A. (2019). *The VGG image annotator (VIA)*. CoRR, abs/1904.10699.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). *Convolutional two-stream network fusion for video action recognition*. ArXiv. Available at: http://arxiv.org/pdf/1604.06573v2.

Gkioxari, G., Girshick, R., Dollár, P., and He, K. (2017). *Detecting and recognizing human-object interactions*. ArXiv. Available at: http://arxiv.org/pdf/1704.07333v3.

Gu, C., Sun, C., Ross, D., and Li, Y. (2018). "Ava: A video dataset of spatio-temporally localized atomic visual actions," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6047–6056.

Güler, R. A., Neverova, N., and Kokkinos, I. (2018). DensePose: Dense human pose estimation in the wild: arXiv.

Gygli, M., and Ferrari, V. (2020). Efficient object annotation via speaking and pointing. *Int. J. Comput. Vis.* 128 (5), 1061–1075. doi:10.1007/s11263-019-01255-4

Halim, J., Eichler, P., Krusche, S., Bdiwi, M., and Ihlenfeldt, S. (2022). No-Code robotic programming for agile production: A new markerless-approach for multimodal natural interaction in a human-robot collaboration context. *Front. robotics AI* 9, 1001955. doi:10.3389/frobt.2022.1001955

Heilbron, F. C., Escorcia, V., Ghanem, B., and Niebles, J. C. (2015). "ActivityNet: A large-scale video benchmark for human activity understanding," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 961–970.

Hintjens, P. (2011). 0MQ - the guide.

Hu, J., Zheng, W., Lai, J., and Zhang, J. (2017). Jointly learning heterogeneous features for RGB-D activity recognition. *IEEE Trans. pattern analysis Mach. Intell.* 39, 2186–2200. doi:10.1109/tpami.2016.2640292

Intel (2021). Computer vision annotation tool. Available at: https://github.com/openvinotoolkit/cvat (Accessed Jul19, 2022).

Jang, J., Kim, D., Park, C., Jang, M., Lee, J., and Kim, J. (2020). "ETRI-Activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 10990–10997.

Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., et al. (2020). Whole-body human pose estimation in the wild: arXiv.

Kabra, M., Robie, A., Rivera-Alba, M., Branson, S., and Branson, K. (2013). Jaaba: Interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* 10, 64–67. doi:10.1038/nmeth.2281

Kay, W., Carreing, J., Zhang, B., and Hillier, C., "The kinetics human action video dataset," ArXiv, abs/1705.06950, 2017.

Kocabas, M., Karagoz, S., and Akbas, E. (2018). MultiPoseNet: Fast multi-person pose estimation using pose residual network: arXiv.

Kreiss, S., Bertoni, L., and Alahi, A., (2021). OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association: arXiv.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., et al. (2020). The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.* 128 (7), 1956–1981. doi:10.1007/s11263-020-01316-z

Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.-S., and Lu, C. (2018). CrowdPose: Efficient crowded scenes pose estimation and A new benchmark: arXiv

Li, M., Chen, S., Zhao, Y., Zhang, Y., Wang, Y., and Tian, Q. (2020). Dynamic multiscale graph neural networks for 3D skeleton-based human motion prediction. arxiv. Available at: http://arxiv.org/pdf/2003.08802v1.

Li, T., Liu, J., Zhang, W., Ni, Y., Wang, W., and Li, Z. (2021). "UAV-human: A large benchmark for human behavior understanding with unmanned aerial vehicles," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16261–16270.

Liang, J., Jiang, L., Niebles, J. C., Hauptmann, A., and Fei-Fei, L. (2019). Peeking into the future: Predicting future person activities and locations in videos. ArXiv. Available at: http://arxiv.org/pdf/1902.03748v3.

Lin, T.-Y., Maire, M., and Hays, J. (2014). Microsoft COCO: Common objects in context. Available at: http://arxiv.org.pdf/.1405.0312v3.

Liu, C., Hu, Y., Li, Y., Song, S., and Liu, J. (2017). PKU-MMD: A large scale benchmark for skeleton-based human action understanding. *VSCC* 17.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L. Y., and Kot, A. C. (2020). Ntu RGB+D 120: A large-scale benchmark for 3D human activity understanding. *IEEE Trans. pattern analysis Mach. Intell.* 42, 2684–2701. doi:10.1109/tpami.2019.2916873

Lucas, S., Carreira, J., Noland, E., Clancy, E., Wu, A., and Zisserman, A., "A short note on the kinetics-700-2020 human action dataset," ArXiv, abs/2010.10864, 2020.

Mao, W., Liu, M., and Salzmann, M. (2021). Generating smooth pose sequences for diverse human motion prediction. arxiv. Available at: http://arxiv.org/pdf/2108.08422v3.

Mao, W., Liu, M., and Salzmann, M. (2020). History repeats itself: Human motion prediction via motion attention. arxiv. Available at: http://arxiv.org/pdf/2007.11755v1.

Martínez-González, A., Villamizar, M., and Odobez, J.-M. (2021). Pose transformers (POTR): Human motion prediction with non-autoregressive transformers. arxiv. Available at: http://arxiv.org/pdf/2109.07531v1.

Morais, R., Le, V., Tran, T., and Venkatesh, S. (2020). *Learning to abstract and predict human actions*. ArXiv. Available at: http://arxiv.org/pdf/2008.09234v1.

Nasdaq Helsinki: QTCOM (2021). Nasdaq Helsinki: QTCOM(Qt group). Available at: https://www.qt.io/(Accessed Aug23, 2022).

Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., and Ferrari, V. (2016). *We don't need no bounding-boxes: Training object class detectors using only human verification*. Available at: http://arxiv.org/pdf/1602.08405v3.

Paszke, A., Gross, S., Massa, F., Lerer, A., Fang, L., Lin, Z., et al. (2019). "PyTorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems 32*. Editors H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett (Red Hook, NY: Curran Associates, Inc), 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2016). Coarse-to-Fine volumetric prediction for single-image 3D human pose. arxiv. Available at: http://arxiv.org/pdf/1611.07828v2.

Punnakkal, A. R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., and Michael, J. (2021). "Black max planck institute for intelligent systems, and universität konstanz, "BABEL: Bodies, action and behavior with English labels," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 722–731.

Quan, H., and Bonarini, A. (2022). Havptat: A human activity video pose tracking annotation tool. *Softw. Impacts* 12, 100278. doi:10.1016/j.simpa.2022.100278

Rai, N., Chen, H., Ji, J., and Adeli, E. (2021). "Home action genome: Cooperative compositional action understanding," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11179–11188.

Rashid, A., Peesapati, K., Bdiwi, M., Krusche, S., Hardt, W., and Putz, M. (2020). "Local and global sensors for collision avoidance," in 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, Sep. 2020 - Sep, 354–359.

Redmon, J., and Farhadi, A. (2018). YOLOv3: An incremental improvement. Available at: http://arxiv.org/pdf/1804.02767v1.

Riegler, M. (2014). "Mathias lux, vincent charvillat, axel carlier, raynor vliegendhart, and martha larson, "VideoJot: A multifunctional video annotation tool," in Proceedings of International Conference on Multimedia Retrieval.

Ruiz, A. H., Gall, J., and Moreno-Noguer, F. (2018). Human motion prediction via spatio-temporal inpainting. arxiv. Available at: http://arxiv.org/pdf/1812.05478v2.

Russakovsky, O., Deng, J., Su, H., and Ma, S. (2014). ImageNet large scale visual recognition challenge. Available at: http://arxiv.org/pdf/1409.0575v3.

Schroeder, W., Martin, K., and Lorensen, B. (2006). *The visualization toolkit-an object-oriented approach to 3D graphics*. Kitware, Inc.

Shahroudy, A., Liu, J., Ng, T-T., and Wang, G. (2016). "Ntu RGB+D: A large scale dataset for 3D human activity analysis," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1010–1019.

Shao, D., Zhao, Y., Dai, B., and Lin, D. (2020). "FineGym: A hierarchical video dataset for fine-grained action understanding," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2613–2622.

Su, H., Mariani, A., Ovur, S. E., Menciassi, A., Ferrigno, G., and de Momi, E. (2021). Toward teaching by demonstration for robot-assisted minimally invasive surgery. *IEEE Trans. Autom. Sci. Eng.* 18 (2), 484–494. doi:10.1109/TASE.2020.3045655

Su, H., Qi, W., Chen, J., and Zhang, D. (2022). Fuzzy approximation-based task-space control of robot manipulators with remote center of motion constraint. *IEEE Trans. Fuzzy Syst.* 30 (6), 1564–1573. doi:10.1109/TFUZZ.2022.3157075

Sun, K., Xiao, B., Liu, D., and Wang, J., (2019). Deep high-resolution representation learning for human pose estimation: arXiv.

Trong, N. P., Nguyen, H., Kazunori, K., and Le Hoai, B. (2017). "A comprehensive survey on human activity prediction," in *Lecture notes in computer science, computational science and its applications – iccsa 2017*. Editor O. Gervasi (Cham: Springer International Publishing), 411–425.

Vondrick, C., Patterson, D. J., and Ramanan, D. (2012). Efficiently scaling up crowdsourced video annotation. *Int. J. Comput. Vis.* 101, 184–204. doi:10.1007/s11263-012-0564-1

Wang, J., Nie, X., Yin, X., Wu, Y., and Zhu, S-C. (2014). "Cross-view action modeling, learning, and recognition," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2649–2656.

Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking: arXiv.

Yuan, Y., and Kitani, K. (2020). DLow: Diversifying latent flows for diverse human motion prediction. arxiv. Available at: http://arxiv.org/pdf/2003.08386v2.

Yuen, J., Russell, B. C., Liu, C., and Torralba, A. (2009). "LabelMe video: Building a video database with human annotations," in 2009 IEEE 12th International Conference on Computer Vision, 1451–1458.

TYPE Original Research
PUBLISHED 06 April 2023
DOI 10.3389/frobt.2023.1028411

Check for updates

OPEN ACCESS

EDITED BY
Alessandro Ridolfi,
University of Florence, Italy

REVIEWED BY
Alexander Arntz,
Ruhr West University of Applied Sciences, Germany
Sabrine Kheriji,
Chemnitz University of Technology, Germany

*CORRESPONDENCE
Aquib Rashid,
✉ aquibrash87@gmail.com

SPECIALTY SECTION
This article was submitted to Robotic Control Systems, a section of the journal Frontiers in Robotics and AI

RECEIVED 26 August 2022
ACCEPTED 23 January 2023
PUBLISHED 06 April 2023

CITATION
Rashid A, Alnaser I, Bdiwi M and Ihlenfeldt S (2023), Flexible sensor concept and an efficient integrated sensing controlling for an efficient human-robot collaboration using 3D local global sensing systems. *Front. Robot. AI* 10:1028411.
doi: 10.3389/frobt.2023.1028411

Flexible sensor concept and an efficient integrated sensing controlling for an efficient human-robot collaboration using 3D local global sensing systems

Aquib Rashid*, Ibrahim Alnaser, Mohamad Bdiwi and Steffen Ihlenfeldt

Department of Cognitive Human Machine Systems, Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany

Human-robot collaboration with traditional industrial robots is a cardinal step towards agile manufacturing and re-manufacturing processes. These processes require constant human presence, which results in lower operational efficiency based on current industrial collision avoidance systems. The work proposes a novel local and global sensing framework, which discusses a flexible sensor concept comprising a single 2D or 3D LiDAR while formulating occlusion due to the robot body. Moreover, this work extends the previous local global sensing methodology to incorporate local (co-moving) 3D sensors on the robot body. The local 3D camera faces toward the robot occlusion area, resulted from the robot body in front of a single global 3D LiDAR. Apart from the sensor concept, this work also proposes an efficient method to estimate sensitivity and reactivity of sensing and control sub-systems The proposed methodologies are tested with a heavy-duty industrial robot along with a 3D LiDAR and camera. The integrated local global sensing methods allow high robot speeds resulting in process efficiency while ensuring human safety and sensor flexibility.

KEYWORDS

collision avoidance, human–robot collaboration, intrusion distance, sensor concept, distance sensors

1 Introduction

Traditional industrial robots better complement human workers with their large range and high payload capabilities. These capabilities are required in multiple manufacturing processes. Furthermore, Gerbers et al. (2018), Huang et al. (2019), Liu et al. (2019), and Zorn et al. (2022) also proposed a human–robot collaborative disassembly as a means of sustainable production. More than 95% of the total robots installed in the world between 2017 and 2019 are traditional industrial robots (Bauer et al., 2016). Moreover, there is a gradual increase in the single human–single robot collaborative processes (IFR, 2020).

Touchless and distance-based collision avoidance systems are required to enable traditional industrial robots to collaborate efficiently with humans. Increased efforts toward e-mobility and sustainability have opened new challenges in the waste disposal sectors. For example, the shredding of batteries and cars decreases engineering production value. Disassembly, however, can help further save engineering and energy costs while reducing carbon emissions. Full automation of the disassembly would require a large amount of data for AI engines, which can

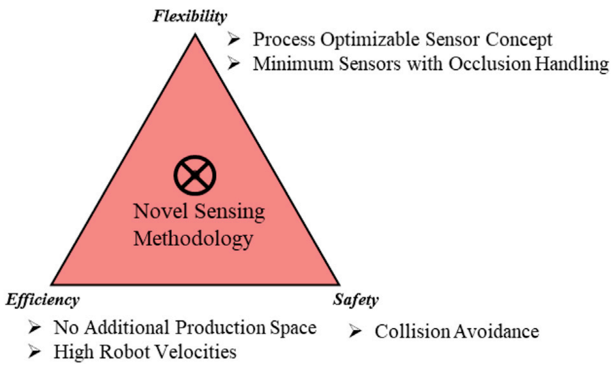

Flexibility
Process Optimizable Sensor Concept
Minimum Sensors with Occlusion Handling
Novel Sensing Methodology
Efficiency
No Additional Production Space
High Robot Velocities
Safety
Collision Avoidance

FIGURE 1
Research target.

be collected with an intermediate solution of a constant human–robot collaboration. Ensuring operational efficiency and varied requirements for disassembly processes requires new collision sensing methodologies.

Current safety standards aim to completely stop the robot before the human comes in contact. These standards enabled fenceless robot cells, where safety sensors like the laser scanner/2D LiDAR sensor are installed to ensure a protective separation distance (PSD) between the human and robot as follows:

$$PSD(t_0) \geq C + S_h + S_r + S_s + Z_R + Z_S, \qquad (1)$$

where C is the linear intrusion distance inside the sensor field of view, after which a detection is triggered. S_h& S_r are the distances covered by the human and robot before the actuating system reacts to the signal from the sensing system. S_s is the distance covered by the robot before stopping while being controlled by the actuating system. Z_R and Z_S are the inaccuracies in position estimation from the robot and sensor, respectively. This strategy is termed speed and separation monitoring (SSM) and is applicable for presence detection-based sensors with occasional operator presence.

Eq. 1 can be used inversely (Byner et al., 2019) when the actual separation distance (ASD) between the robot and human is known to determine the maximum safety robot velocity of the robot (v_r^{safe}) as follows:

$$v_r^{safe} \leq \sqrt{v_h^2 + (T_r a_s)^2 - 2a_s (C - ASD)} - T_r a_s - v_h, \qquad (2)$$

where T_r is the reaction time of the actuation system, a_s is the maximum negative deceleration of the robot, and v_h is the expected human velocity. This approach is termed dynamic speed and separation monitoring (DSSM).

The main focus of this work is to develop a novel sensing methodology, which can be used in the context of traditional industrial robots with constant human presence. The state of the art is evaluated toward this goal with three main parameters, as shown in Figure 1. Agile production requires a flexible sensor concept, which can be adjusted to the need of the process. Furthermore, as each process may require different levels of complexity and human intervention, the sensing methodology should be flexible and scalable while ensuring occlusion handling with minimum sensors. Two main conditions are considered for efficiency. Previously set-up robotic cells may have limited production space as resources. Moreover, the operational efficiency of the process should be achieved by high robot velocities. Finally, collision avoidance would be ensured for the complete human body, which requires 3D sensing.

The main contribution of this work is the methodologies proposed in the design and implementation of the sensor concept for a human–robot collaboration with traditional industrial robots. These contributions are highlighted as follows:

1) Co-existence cell design, which discusses the LiDAR-based sensor concept with limited resources, variable need for shared space, and utilization of the entrance area for prior detection.
2) An efficient method to estimate the intrusion distance and reaction time parameters of a collision avoidance system for speed and separation monitoring.

2 State of the art

2D LiDAR-based sensing approaches have been discussed. This approach approximates the human position based on a cylindrical model (Som, 2005). The safety approach implemented here is based on Tri-mode SSM (Marvel, 2013), which includes not only PSD (the stop area) but also slow and normal speed areas. Nevertheless, the approach results in low operational efficiency due to the constant presence of humans in the slow area. Byner et al. (2019) ensured higher efficiency by proposing dynamic speed and separation monitoring. Nevertheless, the 2D LiDAR approach limits the applicability in the constant operator presence scenario, where the upper limbs are not detected. Human upper limbs can move at twice the speed of the estimated human velocity (Weitschat et al., 2018). The 3D LiDAR approach with a higher vertical field of view and accuracy than the 2D LiDAR approach and fixed field of view (FOV) 3D depth cameras. Moreover, fixed FOV-based 3D depth cameras require additional production space to capture the complete robot workspace area, as discussed by Morato et al. (2014).

Flacco et al. (2012) proposed efficient and high robot velocities for a limited workspace area. The depth camera is installed outside of the robot body, looking toward the human workspace area. The approach, however, suffers from occlusion from a large traditional robot. Moreover, no method is proposed to ensure compliance with the safety standards as no intrusion distance measurements are provided.

The single sensor-based approach by Kuhn and Henrich.,(2007) projected an expanding convex mesh from a virtual robot model on images. The minimum distance was estimated by performing a binary search until an unknown object intersects the projected hull. Similar to Flacco, the approach is not applicable for large traditional industrial robots as the sensor concept would require additional space to ensure covering the large robot body.

Two important research problems have been identified:

1) A generic sensor concept needs to be proposed, which discusses the 2D or 3D LiDAR sensor concept from the aspects of flexibility and occlusion handling.
2) Furthermore, means to measure the intrusion distance for 3D cameras on the robot body need to be discussed.

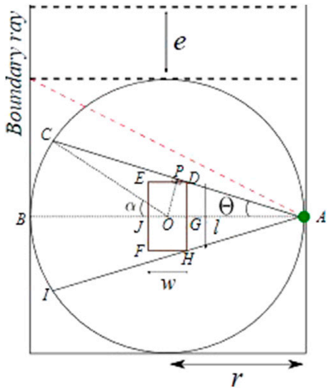


FIGURE 2
Different types of occlusions.

Sensor concept and design aspects have been proposed by Flacco and De Luca (2010), addressing presence and detection-based sensors. Nevertheless, they are not applicable for 2D LiDAR with a large traditional robot causing the main occlusion. Moreover, the intrusion distance measurement for sensors on the robot body is extended from our previous work (Rashid et al., 2022).

3 Flexible and efficient sensor concepts for LiDAR and 3D camera

The work proposes a novel local global sensing framework, which comprises 1) a flexible global (LiDAR-based) sensor concept and 2) an efficient local (3D cameras) intrusion distance measurement method. The integrated approach provides an efficient and flexible collaboration with traditional industrial robots using the novel local global sensing methodology. The local-global sensing in the previous work used a stationary LiDAR and camera sensor (Rashid et al., 2021). This work further introduces co-moving cameras as local sensors to ensure safety from grippers and objects gripped, as illustrated in Figure 2.

3.1 Flexible global sensor concept

This section discusses an occlusion-aware sensor concept applicable to 2D and 3D LiDAR. The sensor concept is discussed by proposing a standardized co-existence cell model in 2D based on three parameters. These parameters are maximum robot reach r, space available toward the entry area e, and the number of entry sides n.

Occlusion with a single LiDAR sensor can be caused due to static objects in the field of view or a dynamic robot body. For the LiDAR sensor placed without any orientation, the robot base link results in the most evident occlusion in the cell. The green circle represents the LiDAR sensor, and the robot base link is represented with a rectangle with maximum length and width (l x w). The base rectangle is placed at the center of the robot workspace, which is enclosed by three sides (n = 1) with safety fences. The entrance area e is assumed to be free from any static occlusions. A ray from LiDAR, represented by a dotted red line, which when intercepted, results in entry occlusion, is termed a boundary ray, as shown in Figure 3.

For the sensor position in scenario A, illustrated in Figure 3, occlusion is constrained by the safety fence, thus having a lower risk of possible human collision. Moreover, for scenario B, the occlusion is unconstrained toward the entrance area and is at higher risk of a possible collision. Unconstrained occlusions are avoided by allowing LiDAR placement only on the adjacent entrance walls. Furthermore, the boundary ray polar coordinates are used to set constraints on robot motion. These measures allow safety by design. The constrained occlusion caused by the robot base requires to be mathematically formulated.

In the 6D pose for the 360° HFOV LiDAR, no orientations are assumed to exist for the robot reference frame. Any yaw or pitch orientations would result in non-uniform coverage of the production area. Moreover, 3D LiDAR comprises multiple 2D laser channels, which rotate at a certain orientation, as illustrated in Figure 4. The height parameter has a direct relationship with the area of the circle with the radius $r1$. The circle represents the blind spot in the 2D floor space of the co-existence cell and can be expressed as follows:

$$r1 = h^{*}\tan(\alpha), \tag{3}$$

thus lowering the height of LiDAR results in a decreased blind spot area. On the other hand, increasing the height of the LiDAR, to a specific extent, results in increasing the number of rays falling inside the robot cell. The higher number of rays corresponds to a higher accuracy of human localization estimations. An optimal height would be related to average worker heights and sensor vertical resolution. The risks from the blind spot area can be reduced by ensuring that the human is localized minimum by the 0° measuring plane. This leaves the XY plane, on which the Y-axis is constrained to avoid unconstrained occlusion. Thus, only 1D degree of freedom is available for the LiDAR sensor concept. Nevertheless, this 1D is enough to cover a variety of process requirements, while incurring no additional production space.

Let the LiDAR sensor be placed at A, representing the middle of the safety fence. Maximum occlusion is caused when the robot base is perpendicular to the global sensor. This occlusion area is represented by an area of polygon IHFEDC, as illustrated in Figure 5. The occluded area, in this configuration, can be computed in robot parameters, by drawing perpendicular OP on AC and joining OC, as shown in Figure 5. The occluded area CPEFHI is computed by first computing its half area, constituting the area of Δ AOC and sector BOC. Then, removing half of the base rectangle and Δ AGD gives a symmetric one-sided constrained occlusion area.

The sector BOC inscribes the same arc BC as that of BAC. This results in the angle (α) at the sector BOC, being twice the angle (θ) at the sector BAC. Using this, the area of the sector BOC can be defined as follows:

$$Area\ BOC = \frac{2\theta}{360}{}^{*}\pi r^{2}\,, \tag{4}$$

where r represents the maximum reach of the robot used.

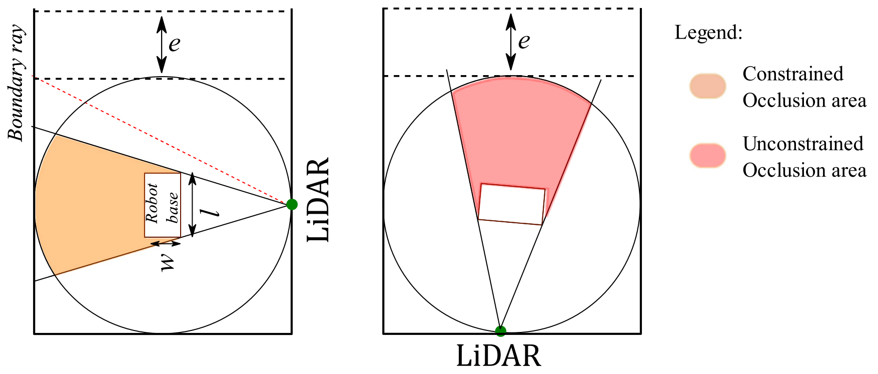
Furthermore, the area of ΔAOC can be computed as follows:

$$\Delta\mathrm{AOC} = \frac{1}{2}\,(\mathrm{OP}^{*}\mathrm{AC}). \tag{5}$$

Using the property of isosceles triangle ΔAOC,

$$AC = 2AP\,. \tag{6}$$

Sides AP and OP can be expressed as follows:



Scenario A: Sensor on wall adjacent to entrance

Scenario B: Sensor on wall opposite to entrance

FIGURE 3
Proposed flexible and efficient sensor system.

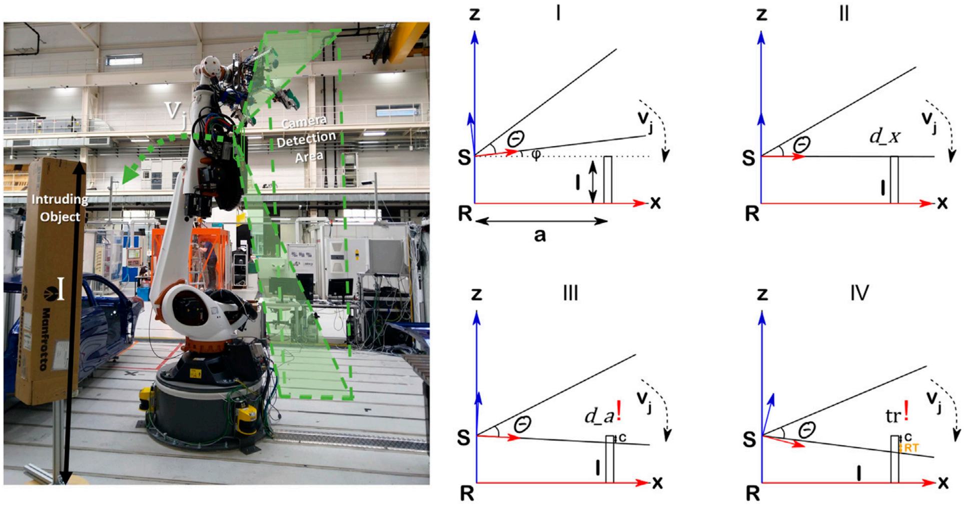


FIGURE 4
LiDAR sensor concept.

$$AP = PC = r^{*}\cos\theta, \tag{7}$$

$$OP = r^{*}\sin\theta. \tag{8}$$

Substituting Eqs 6, 7, and 8 in Eq. 5, we obtain the following equation:

$$\Delta \mathrm{AOC} = \frac{1}{2}\left(\left(r^{*}\sin\theta\right)^{*}2\left(r^{*}\cos\theta\right)\right). \tag{9}$$

Using Eqs 4 and 9, the area of BAC can be defined as follows:

$$Area\ of\ BAC = \frac{2(\theta)}{360}{}^{*}\pi r^{2} + \frac{1}{2}\left(\left(r^{*}\sin\theta\right)^{*}2\left(r^{*}\cos\theta\right)\right). \tag{10}$$

The symmetric half occlusion area can be computed using Eq. 10, by removing the area of the robot body and ΔAGD as follows:

$$Area\ of\ \mathrm{BJEDC} = Area\ BAC - \frac{1}{2}\left(Area\,of\ Rectangle\ EFHD\right) - Area\Delta\mathrm{AGD}. \tag{11}$$

Using Eqs 9–11, we get

$$\begin{aligned} Area\ of\ BJEDC = {} & \frac{2(\theta)}{360}{}^{*}\pi r^{2} + \frac{1}{2}\left(\left(r^{*}\sin\theta\right)^{*}\left(2r^{*}\cos\theta\right)\right) \\ & - \frac{1}{2}\left(\mathrm{l}^{*}w\right) - \frac{1}{2}\left(\frac{l}{2}{}^{*}\left(r - \frac{w}{2}\right)\right). \end{aligned} \tag{12}$$

Eq. 12 can be used to compute the overall occlusion area IHFEDC as follows:

$$\begin{aligned} Area\,of\ \mathrm{IHFEDC} &= 2\left(Area\,of\ BJEDC\right) \\ &= 2\left(\frac{2(\theta)}{360}{}^{*}\pi r^{2} + \frac{1}{2}\left(\left(r^{*}\sin\theta\right)^{*}\left(2r^{*}\cos\theta\right)\right)\right. \\ &\quad \left. - \frac{1}{2}\left(\mathrm{l}^{*}w\right) - \frac{1}{2}\left(\frac{l}{2}{}^{*}\left(r - \frac{w}{2}\right)\right)\right). \end{aligned} \tag{13}$$

Eq. 1 gives a generalized equation to compute the occlusion area, with the distance between the sensor and robot r, robot base link dimensions, and occlusion angle (2θ) with the robot body.

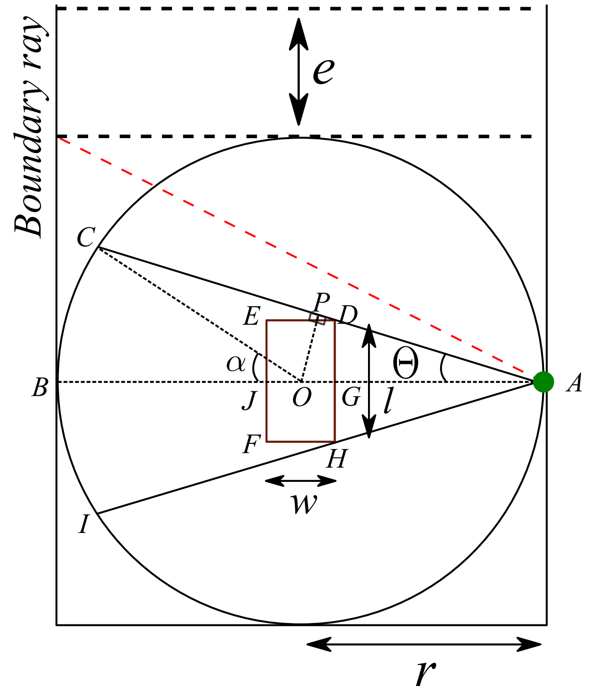


FIGURE 5
Constrained occlusion with the sensor at the middle of the fence.

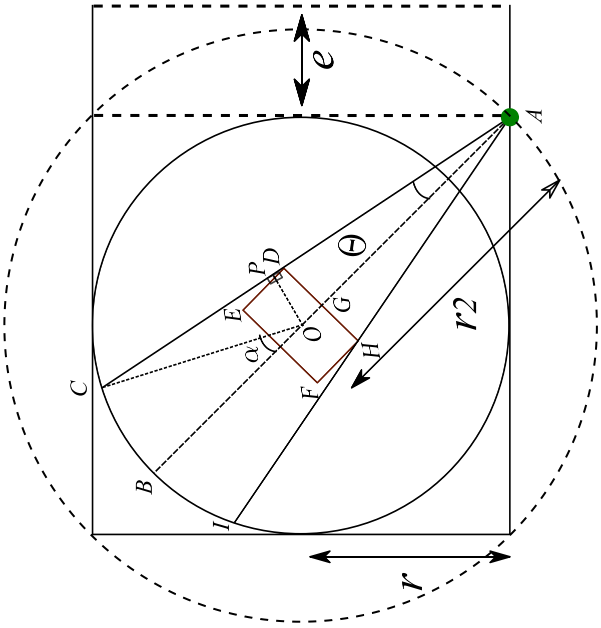


FIGURE 6
Constrained occlusion with the sensor at the entrance.

For a specific sensor position on the safety fence in a co-existence cell model, the overall occlusion area can also be defined by representing the value of angle ∠GAD (θ) as follows:

$$\theta = \tan^{-1}\left(\frac{GD}{AG}\right) = \tan^{-1}\left(\frac{\frac{l}{2}}{r-\frac{w}{2}}\right). \tag{14}$$

Substituting Eq. 13 into Eq. 14, the overall generalized calculation of the occlusion area, with the sensor at the middle point of the co-existence cell fence, as shown in Figure 5, is given by the following:

$$Area\,of\ \mathrm{IHFEDC} = 2^{*}\left(\begin{matrix} \dfrac{2\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{r-\frac{w}{2}}\right)\right)}{360}{}^{*}\pi r^{2}+ \\ \dfrac{1}{2}\left(\begin{matrix}\left(r^{*}\sin\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{r-\frac{w}{2}}\right)\right)\right)^{*} \\ \left(2r^{*}\cos\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{r-\frac{w}{2}}\right)\right)\right)\end{matrix}\right)- \\ \frac{1}{2}\,(l^{*}w)-\frac{1}{2}\left(\frac{l}{2}{}^{*}\left(r-\frac{w}{2}\right)\right)\end{matrix}\right). \tag{15}$$

Similarly, for the sensor placed at the entrance corner start position, as illustrated in Figure 6, the radius $r2$ can be computed as half of the length of the diagonal of the square as follows:

$$r2 = \sqrt{2}r\,. \tag{16}$$

The aforementioned Eq. 16 is substituted in 13, resulting in the overall occluded area (Area of IHFEDC) at the start position as follows:

$$= 2^{*}\left(\begin{matrix} \dfrac{2\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{\sqrt{2}\,r-\frac{w}{2}}\right)\right)}{360}{}^{*}\pi r^{2} \\ +\dfrac{1}{2}\left(\left(r^{*}\sin\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{\sqrt{2}\,r-\frac{w}{2}}\right)\right)\right)^{*}\left(2r^{*}\cos\left(\tan^{-1}\left(\dfrac{\frac{l}{2}}{\sqrt{2}\,r-\frac{w}{2}}\right)\right)\right)\right) \\ -\frac{1}{2}\,(l^{*}w)-\frac{1}{2}\left(\frac{l}{2}{}^{*}\left(\sqrt{2}\,r-\frac{w}{2}\right)\right)\end{matrix}\right),$$

where *w* and *l* represent the width and length of the base link. It is evident that the constrained occlusion for the scenario with LiDAR at the entrance is minimum. Nevertheless, the optimal position of the LiDAR (global) sensor could be adjusted based on process requirements and available resources.

The 1D variation of LiDAR between the start and midpoint of the fence fulfills varied perception requirements for varied processes. The final placement of LiDAR would divide the overall workspace into 1) constrained occlusion and 2) shared workspace. The occupancy of these workspaces is checked in the real-time collision avoidance system, using polar coordinate limits. Extrinsic calibration between the robot and LiDAR is assumed to be known (Rashid et al., 2020). Finally, the constrained occlusion area can further be covered using a local 3D camera on the robot body, facing toward the shadow area. The local collision avoidance setup is already discussed in our previous work (Rashid et al., 2021).

3.2 Intrusion distance and reaction time estimation for co-moving local sensors

The proposed method provides the intrusion distance estimation for co-moving local sensors on the robot body. This method comprises

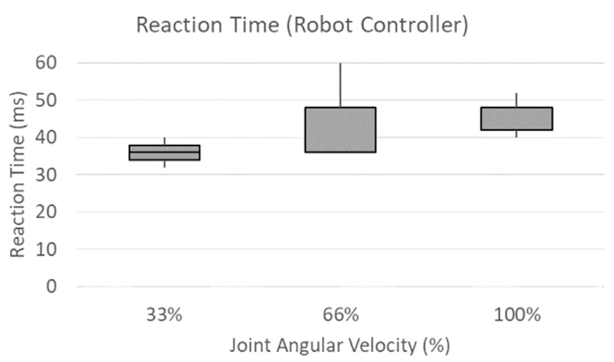


FIGURE 7
Data capturing experiment for safety parameter estimation.

three main steps. The first step involves setting up an external intruding object, which can be detected from the presence detection algorithm for local sensing. This is followed by multiple controlled robot experiments at a defined angular velocity. The data coming from the robot joints and sensing detections are recorded in an external processing system. The final step involves offline processing of the recorded data, with some a priori data as input to provide safety parameters as output. These sub-steps are discussed in detail in Section 3.2.2.

3.2.1 Controlled experiments' overview from illustrations

A simplistic concept for the estimation of the parameter for local sensing can be understood in Figure 7. The intruding object of height I is placed in the robot work cell at a distance of a vector. A 3D camera is mounted on the robot body with reference frame S and a field of view Θ, as illustrated in Section 1 of Figure 7. A robot trajectory with the tool center point velocity of v_j is performed, ensuring that the intruding object is piercing almost perpendicularly into the sensor field of view, as illustrated in section II of Figure 7. The known system and intruding object features ensure the pre-estimation of the expected detection (d_x), where the sensor field of view first touches the intruding object. However, the sensor data being processed in an external system results in a delay. Thus, the actual detected (d_a) is flagged at the future position, giving an estimate for intrusion distance (C), as illustrated in part III of Figure 7. The flagged detection issues a stop signal from the external processing system to the robot controller. The stopping trigger (tr) is perceived by the start of unplanned deceleration in the external system capturing the robot velocities, as illustrated in part IV of Figure 7. The delay in communication to an external processing system is assumed to be negligible. Moreover, most robot manufacturers provide communication interfaces running at 250–1000 Hz. Finally, the motion of the sensor is assumed to result in linear displacement of the intruding object into the field of view.

3.2.2 Setting up of the intruding object

A simple cardboard box placed on a stand is used as an intruding object. The setup of this intruding object requires positioning the intruding object at a known position with respect to the robot reference frame. The second step of the robot experiment is illustrated in Figure 7. The detection of the intruding object, which is not part of the environment with respect to a moving sensor on the robot body, requires unknown object detection based on local sensing (Rashid et al., 2021).

3.2.3 Offline data processing method for intrusion distance and reaction time measurements

This step involving the offline post-processing of the recorded data is detailed in the following algorithm. The developed software tool parses through multiple iterations at a specific robot velocity. The expected detection (d_x) for a specific set of positions is taken and processed sequentially. In a single iteration for a known intruding object position, the robot position is searched in angular joint coordinates, where the detection flag is active. The corresponding joint angular position JointAngle_(d_a)ĩ is compared to the actual detection (∝_(d_x)) to estimate the intrusion distance. The time stamp at the position of the flagged detection is recorded (t_(d_x)). The joint angular velocities are searched for a deceleration trigger for which the corresponding time stamp is captured (t_tr). The difference between the two time stamps provides an estimate of controller reaction time.

```
Input: Time-stamped Joint Angles, Joint Angular velocity, and Detection Flag over multiple iterations (i) and over a complete set represented by j,k,l, or event
$\{time_t^i, JointAngle_j^i, Joint\%age_k^i,, \propto_{d\_x}\}$
Output: Reaction time (B) of robot in ms; Intrusion distance $\propto_{in}$ in mm
Algorithm:
A
$\leftarrow\{(time_t^i, JointAngle_j^i, Joint\%age_k^i, \Theta_{eventExpected})\}$
$B \leftarrow \varnothing$
While $A \neq \varnothing$ do//Search for the event expected position
    if $JointAngle_j^i \geq \propto_{d\_x}$
      $t_{d\_x} \leftarrow time_{te}^i$//Save time stamp
if $eventDetect = 1$//Actual event detected
$\propto_{in} = JointAngle_{d\_a}^i - \propto_{d\_x}$
While $Joint\%age_n^i \neq \varnothing$//Search for deceleration trigger
if $Joint\%age_{k-9}^i >$
$Joint\%age_{k-8}^i > \ldots Joint\%age_k^i$
      $t_{tr} \leftarrow time_{ta}^i$
       $B \leftarrow t_{tr} - t_{d\_x}$
return B; $\propto_{in}$
```

Algorithm. Reaction time and Intrusion distance

4 Experimental setup for efficient intrusion distance estimation

The experimental setup comprises a stereo camera connected to a processing system over a USB3 connection. The processing system is connected over Ethernet to a robot controller. In this work, ZED1 from Stereo Labs is used as a 3D camera. A heavy-duty industrial robot Kuka KR180 with a range of 2.9 m is used, as illustrated in Figure 8. The KRC4 robot controller is used, which allows 250 Hz UDP communication with an external computer. Joint angular displacement along with a tool

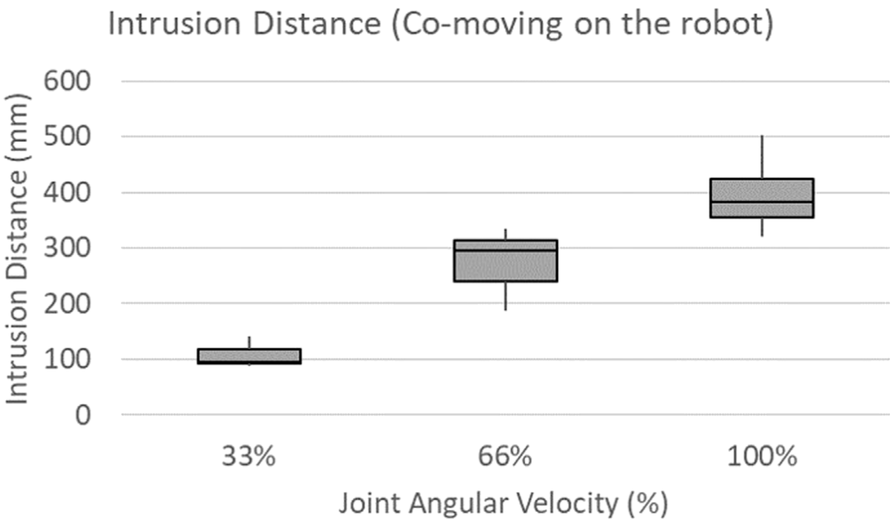


FIGURE 8
Simplified co-existence cell.

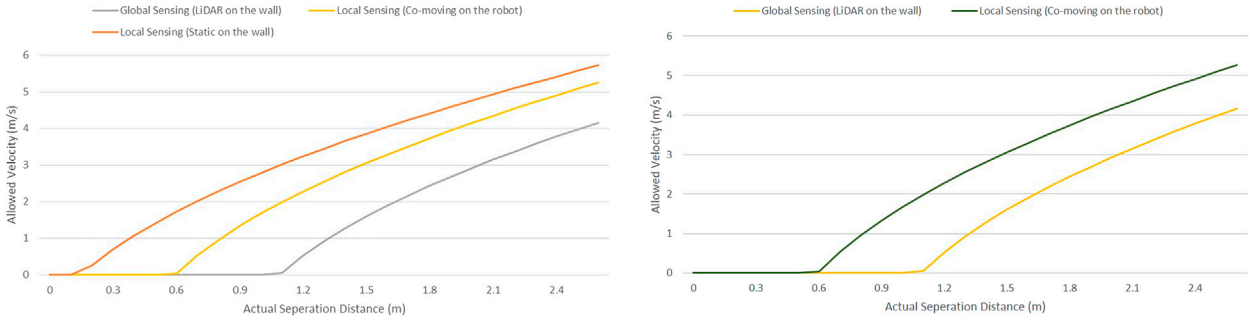


FIGURE 9
Complete trajectory for collision sensing and stop trigger for a single iteration.

center point (TCP) in the base coordinate is provided at the external processing system. In order to compare the intrusion distance of local sensing (Schrimpf, 2013) with that of global sensing (Morato et al., 2014), an identical processing system is used with Intel i7-6700K CPU, Nvidia TITAN Xp GPU, and 12 GB GDDR5 memory.

At 4 ms, as for Kuka communication speed, the robot joint values are used to estimate the position of the local sensor on the robot. A cardboard box is placed in a robot cell of 4 x 4 m. The position of the intruding object is estimated by moving the TCP to the top of the object. More accurate estimations can be performed by using ArUco markers (Garrido-Jurado et al., 2014) on the intruding object. The intruding position for the known sensor is estimated with a ±10 mm precision.

4.1 Experimental results

The real-time experiment with more than five iterations for a single robot velocity is captured for statistical variations. The robot velocities need not be running at a constant velocity before an event, compared to the state of the art (Rashid et al., 2022). This can be challenging for achieving a high robot velocity with a large sensor field of view. This work rather uses a constant accelerating profile, with specific top velocities. An important aspect here is to capture linear robot velocities, which are comparable to or higher than the nominal human speed (1.6 m^2). A constant acceleration profile can be seen in Figure 9. The figure gives a dual vertical graph for joint angular velocity and a detection flag for A2 joint-based motion. The approx. linear velocity for the sensor mounted on A3 amounts to be

$$v_l = r^{*}\varphi/t, \tag{17}$$

where r is the approximate length of the A2 link, which is 1.35 m for Kuka KR180. φ is angular displacement in radians, and t is the total time duration in seconds, Thus, the linear velocity was found to be 1.12 m/s2.

Figure 9 gives a zoomed-in view into the recorded velocity profile for the experiment data from Figure 10, with a target speed of 87% of A2 joint speed. The sensing flag and deceleration trigger are observed at 81,228 and 81,276 timestamps, respectively. The intrusion distance and reaction time is measured in spatial and temporal dimensions.

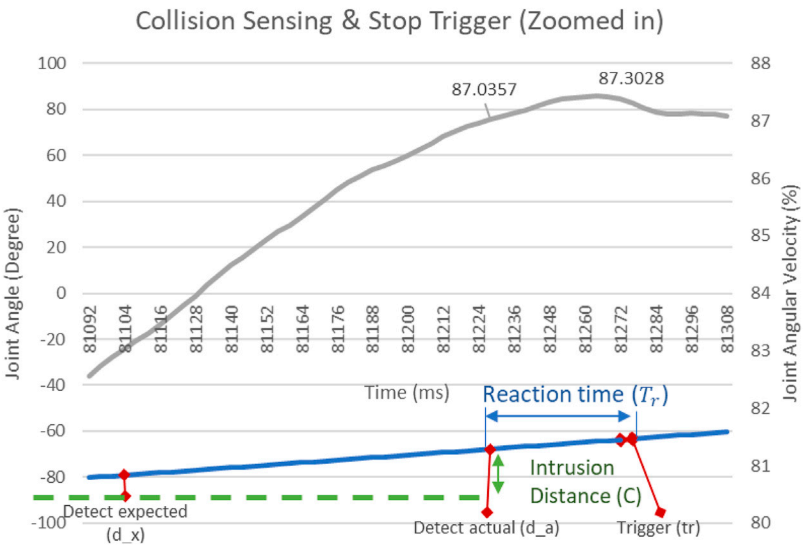


FIGURE 10
Zoomed in trajectory for intrusion distance and reaction time estimation.

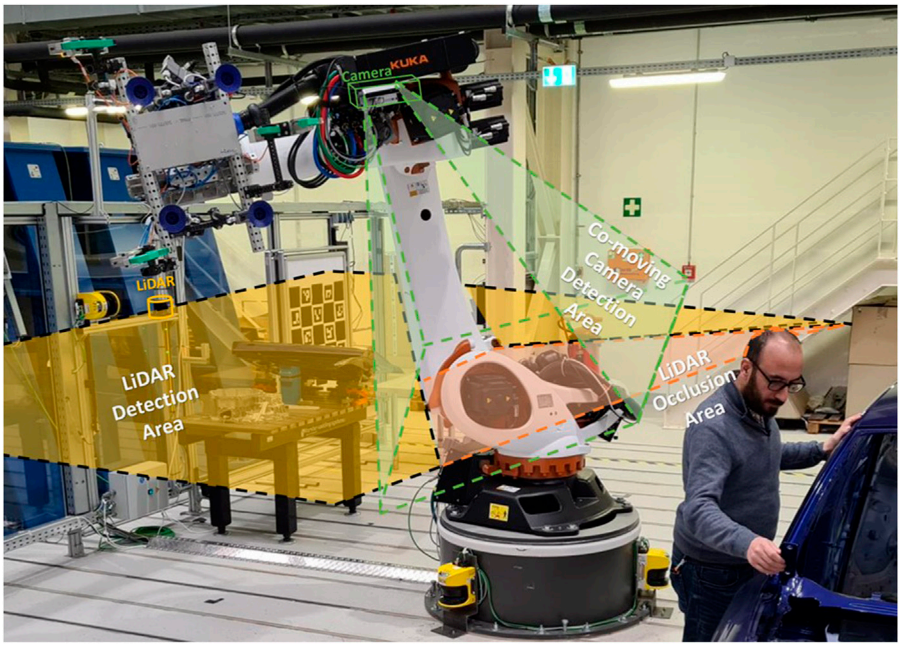


FIGURE 11
Box plot on multiple iterations at multiple robot velocities.

The implementation of local sensing is discussed in Mandischer et al. (2022). The multiple iterations on three robot velocities are processed to estimate the worst-case intrusion distance and reaction time, as illustrated in Figure 11. The worst-case reaction time is calculated as 60 ms, which is comparable to the state-of-the-art calculation of 56 ms and 40 ms (Schrimpf, 2013) for different robot controllers. The worst-case intrusion distance captured at multiple iterations and velocities equals to be 502 mm.

4.2 Discussion on the efficient and flexible constant human–robot collaboration with traditional industrial robots

The method proposed allows safety distance measurement for co-moving or dynamic local sensors on the robot body. The method can be used by system integrators or safety sensor developers, aiming to use distance-based sensors for an efficient collaborative system.

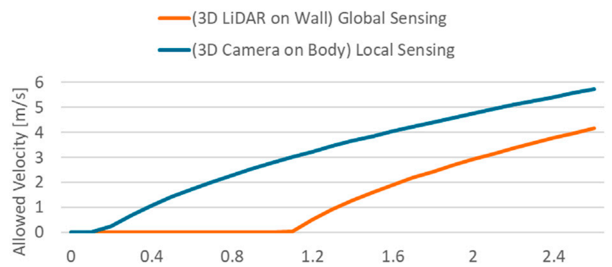


FIGURE 12
Safety distance and robot velocity for local (co-moving) and global collision sensing.

Plotting the worst-case intrusion distance of local sensing with global sensing in Eq. 2, we obtain two different relations between the separation distance and maximum allowed robot velocity, as illustrated in Figure 12. The co-moving local sensor or dynamic local sensing is more efficient, allowing higher robot speed within the proximity human operator. However, the co-moving local sensor is only relevant for the safety of the tool or object in the robot's hand. The LiDAR sensor on the wall, acting as a global sensor, ensures safety from the complete robot body without the constrained occlusion area. This occlusion area can be constantly monitored based on the LiDAR sensor concept. Furthermore, safety from design can be implemented using the LiDAR sensor concept by allotting human workspace away from the constrained occlusion. Future steps would include determining an optimal combination of different local (co-moving/dynamic (Mandischer et al., 2022) or static (Rashid et al., 2021)) and global sensing (Rashid et al., 2021) systems for a given agile or disassembly process in a safety digital twin.

5 Conclusion

The work proposes a method for not only utilizing distance-based sensors for an efficient and flexible collision avoidance system for a human–robot collaboration. The global sensor concept with LiDAR addresses a minimalistic and reduced complexity approach while addressing the occlusion from the robot body. Furthermore, an efficient method is proposed that simultaneously determines the intrusion distance and reaction time for 3D cameras on the robot body and robot controller, respectively. The work proposed can be used to compute safety parameters for a wide variety of distance-based sensors on robot bodies. These methodologies can be implemented for lightweight industrial robots or co-bots. Increased efforts toward resource efficiency and sustainability will require human–robot collaboration. The methodologies proposed will enable the development of close-proximity human–robot collaboration.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

AR contributed to the main method and multiple data collection and data visualization. IA and MB contributed to the abstract and overall review.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bauer, W., Manfred, B., Braun, M., Rally, P., and Scholtz, O. (2016). *Lightweight robots in manual assembly–best to start simply*. Stuttgart: Frauenhofer-Institut für Arbeitswirtschaft und Organisation IAO.

Byner, C., Matthias, B., and Ding, H. (2019). Dynamic speed and separation monitoring for collaborative robot applications–concepts and performance. *Robotics Computer-Integrated Manuf.* 58, 239–252. doi:10.1016/j.rcim.2018.11.002

Flacco, F., and De Luca, A. (2010). "Multiple depth/presence sensors: Integration and optimal placement for human/robot coexistence," in 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 03-07 May 2010 (IEEE).

Flacco, F., Kroger, T., De Luca, A., and Khatib, O. (2012). "A depth space approach to human-robot collision avoidance," in 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14-18 May 2012 (IEEE).

Garrido-Jurado, S., Munoz-Salinas, R., Madrid-Cuevas, F., and Marin-Jimenez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* 47, 2280–2292. doi:10.1016/j.patcog.2014.01.005

Gerbers, R., Wegener, K., Dietrich, F., and Droder, K. (2018). *Safe, flexible and productive human-robot-collaboration for disassembly of lithium-ion batteries, Recycling of Lithium-Ion Batteries*. Berlin, Germany: Springer, 99–126.

Huang, J., Pham, D., Wang, Y., Ji, C., Xu, W., Liu, Q., et al. (2019). A strategy for human-robot collaboration in taking products apart for remanufacture. *Fme Trans.* 474, 731–738. doi:10.5937/fmet1904731h

IFR (2020). *Demystifying collaborative industrial robots*. Frankfurt: Germany International Federation of Robotics.

Kuhn, S., and Henrich, D. (2007). "Fast vision-based minimum distance determination between known and unkown objects," in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October 2007 - 02 November 2007 (IEEE).

Liu, Q., Liu, Z., Xu, W., Tang, Q., Zhou, Z., and Pham, D. T. (2019). Human-robot collaboration in disassembly for sustainable manufacturing. *Int. J. Prod. Res.* 5712, 4027–4044. doi:10.1080/00207543.2019.1578906

Mandischer, N., Weidermann, C., Husing, M., and Corves, V. (2022). Non-contact safety for serial manipulators through optical entry DetectionWith a Co-moving 3D camera sensor. *VDI Mechatronik* 2022, 7254309. doi:10.5281/zenodo.7254309

Marvel, J. A. (2013). Performance metrics of speed and separation monitoring in shared workspaces. *IEEE Trans. automation Sci. Eng.* 10, 405–414. doi:10.1109/tase.2013.2237904

Morato, C., Kaipa, K. N., Zhao, B., and Gupta, S. K. (2014). Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *J. Comput. Inf. Sci. Eng.* 14, 1. doi:10.1115/1.4025810

Rashid, A., Bdiwi, M., Hardt, W., Putz, M., and Ihlenfeldt, S. (2021). "Efficient local and global sensing for human robot collaboration with heavy-duty robots," in 2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Fl, USA, 28-29 October 2021 (IEEE).

Rashid, A., Bdiwi, M., Hardt, W., Putz, M., and Ihlenfeldt, S. (2022). "Intrusion distance and reaction time estimation for safe and efficient industrial robots," in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23-27 May 2022 (IEEE).

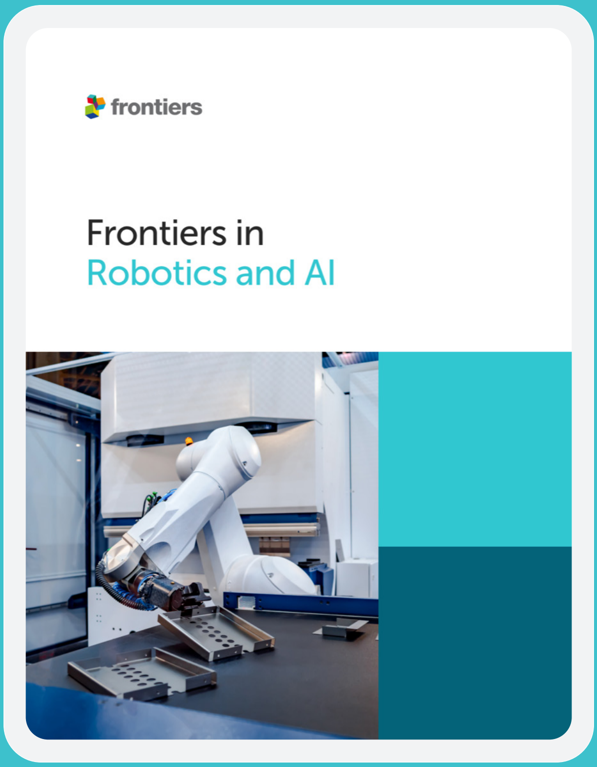
Rashid, A., Bdiwi, M., Hardt, W., Putz, M., and Ihlenfeldt, S. (2020). "Open-box target for extrinsic calibration of LiDAR, camera and industrial robot," in 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA), Shanghai, China, 16-18 October 2020 (IEEE).

Schrimpf, J. (2013). "Sensor-based real-time Control of industrial robots," PhD Thesis (Trondheim: Norwegian University of Science and Technology).

Som, F. (2005). *Sichere Steuerungstechnik für den OTS-Einsatz von Robotern in 4, Workshop für OTS-Systeme in der Robotik*. Stuttgart: Sichere Mensch-Roboter-Interaktion ohne trennende Schutzsysteme.

Weitschat, R., Ehrensperger, J., Maier, M., and Aschemann, H. (2018). "Safe and efficient human-robot collaboration part I: Estimation of human arm motions," in 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, Australia, 21-25 May 2018 (IEEE).

Zorn, M., Ionescu, C., Klohs, D., Zahl, K., Kisseler, N., Daldrup, A., et al. (2022). An approach for automated disassembly of lithium-ion battery packs and high-quality recycling using computer vision, labeling, and material characterization. *Recycling* 4, 48. doi:10.3390/recycling7040048



frontiers

Frontiers in
Robotics and AI

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us
+41 (0)21 510 17 00
frontiersin.org/about/contact