



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

COPYRIGHT

© 2022 Wiese, Abicht, Friedrich,
Hellmich and Ihlenfeldt. This is an open-
access article distributed under the
terms of the [Creative Commons
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which does
not comply with these terms.

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

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 effortlessly. Therefore, programming experts must define all possible process changes in software that limits the flexibility to several 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

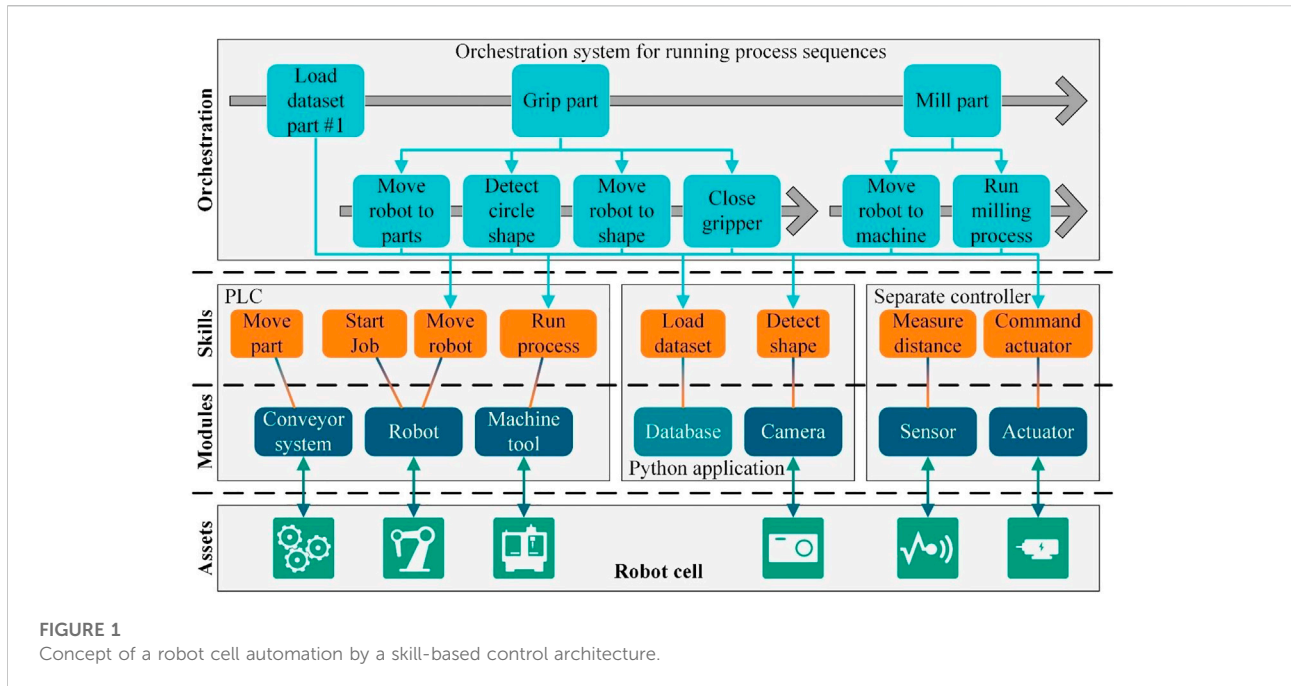
knowledge. The individual functions of the assets must be configurable *via* changeable parameters to be able to adapt them to specific process steps. This allows the operator to configure sequences with differently parameterized function calls of the assets without having control programming expertise.

- **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 für 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



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.

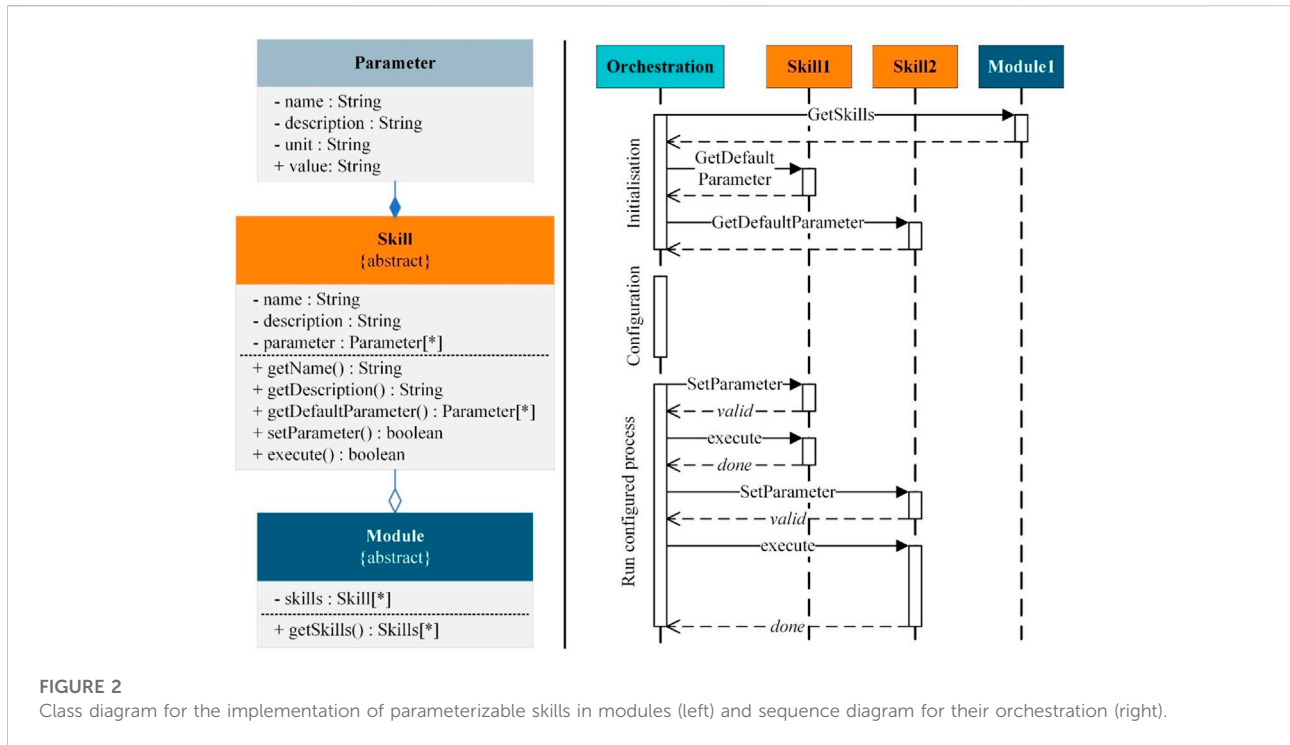


FIGURE 2 Class diagram for the implementation of parameterizable skills in modules (left) and sequence diagram for their orchestration (right).

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.

HMI for configuration

Sequence of skills

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

Skills

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

Editor for skill parameters

EditJobSkill

ID: 11
 Name: Gripper: GripRelease
 Config Group: Roboterzelle

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

Parameter wählen für Beschreibung

Dummy: PTOS

M: Asset modules S: Asset skills Q: Sequence P: Skill parameters R: Reusability

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

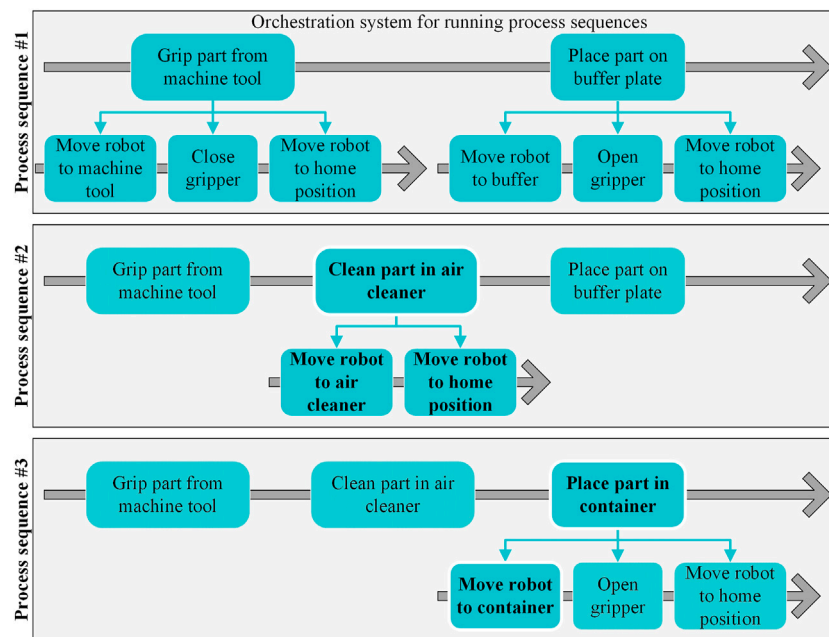


FIGURE 5

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 configure 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. Weißgraber, 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 für 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 German 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., Mysliwiec, 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. Kaliczynska (Cham: Springer International Publishing), 239–247.
- Froh, 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., Escada 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., Breitzkreuz, 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 connectors. in 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.