Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 31 (2019) 270–276

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

9th Conference on Learning Factories 2019

Development of an easy teaching and simulation solution for an autonomous mobile robot system

Vinu Vijayakumaran Nair[a,*], Daniel Kuhn[a], Vera Hummel[a]

[a]*ESB Business School, Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, Germany*

Abstract

With mass customized production becoming the mainstream, industries are shifting from large-scale manufacturing to flexible and customized production of small batch sizes. Agile manufacturing strategies adopted by SMEs are driving the usage of collaborative robots in today's factories. Major challenges in the adoption of cobots in the industry are the lack of a highly trained workforce to program the robot to perform complex tasks and integration of robot systems to other smart devices in the factory. In addition, the teaching and simulation by non-robotics experts of many industrial collaborative robot systems like the KUKA LBR iiwa is a major challenge, since these systems are designed to be programmed by robot experts and not by shop floor workers or other non-experts. This paper describes the research and development activities done for reducing the barriers in operation and ensure holistic integration of LBR iiwa cobot in the assembly on the example of the ESB Logistics Learning Factory. These include a visual programming solution for the easy teaching of various tasks. Robotics tasks are classified based on common robotics applications and application-specific blocks abstracting specific actions are implemented. A factory worker with no programming competency could create robot programs by combining these blocks using a Graphical User Interface. In addition, a simulation solution was developed to visualize, analyse, and optimize robotic workflow before deployment. An autonomous mobile robot is integrated with the LBR iiwa to improve reconfigurability and thus also the productivity. The system as a whole is controlled using an event-driven distributed control system. Finally, the capabilities of the system are analysed based on the design principles of Industrie 4.0 and potential future research ideas are discussed to further improve the system.

© 2019 The Authors. Published by Elsevier B.V.
Peer review under the responsibility of the scientific committee of the 9th Conference on Learning Factories.

Keywords: Collaborative robots; Easy teaching; Cyber physical systems; Visual programming; Virtualisation

1. Introduction

An autonomous mobile collaborative robot is capable of moving across the factory floor to specified locations operating as a flexible production assistant. These robotic systems are imperative in creating a flexible and cognitive manufacturing facility which can respond to changes in production requirements [1]. Integration of such complex robotic solutions to smart factories presents a challenge for novice operators if they are not designed in an adaptable

* Corresponding author. Tel.: +49 (0)7121 271-3142 .
E-mail address: vinu.vijayakumaran_nair@reutlingen-university.de

2351-9789 © 2019 The Authors. Published by Elsevier B.V.
Peer review under the responsibility of the scientific committee of the 9th Conference on Learning Factories.
10.1016/j.promfg.2019.03.043

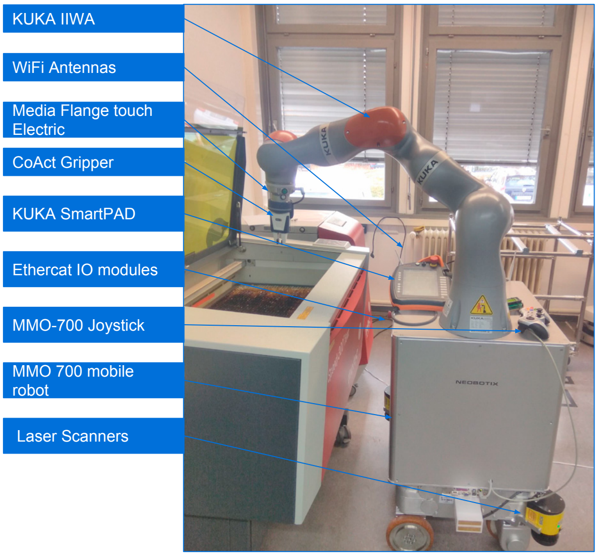

Fig. 1. Autonomous mobile robot system "Clara" at ESB LLF.

and easy to use manner. Development of an easy teaching and simulation solution for an autonomous mobile cobot system is part of the ongoing development and demonstration of future production scenarios at the ESB logistic-learning factory (LLF) [2]. Integration of collaborative robots into hybrid work systems improves the production flexibility, increases productivity and reduces the risk for workers. The developed solution is used at the ESB logistics-learning factory lectures for teaching the students about Human-Robot Collaboration(HRC) and smart factory concepts as well as serving as a demonstration platform of Industrie 4.0.

An intuitive and easy programming interface is one of the most important requirements for a cobot designed to be operated by regular factory workers [3] [4]. Compared to the intuitive online programming approach used by Universal robots, Franka Emika etc, where operators can create programs using the teach pendent and teaching by demonstration, KUKA LBR iiwa uses a hybrid programming approach combining offline programming in Java programming language and online frame teaching. The existing programming solution for the KUKA iiwa is not suitable for a factory worker who is not an expert in programming. In order to make the robot teaching friendlier for a non-programmer user, an easy teaching solution based on a visual programming approach [5] [6] is developed as a proof of concept.

In addition to the easy teaching solution, a simulation of the required process is created using DELMIA apps of the Dassault Systemes 3D Experience Platform [7]. The simulation solution of the KUKA LBR iiwa enables factories to plan their processes ahead of time and validating them before implementation in the factory takes place.

2. System overview

The collaborative robot "Clara" at the ESB LLF consists of LBR iiwa cobot mounted on top of a custom built version of the Neobotix MMO-700 omnidirectional mobile robot as shown in figure 1. The LBR iiwa is equipped with a Schunk Co-Act EGP-C gripper which is inherently safe due to its current monitoring and is certified for HRC applications. The KUKA smartPAD teach pendant is used for axes jogging / reorientation, frame teaching, program execution control and debugging. This robot is equipped with a media flange touch electric with programmable status LEDs, a hand guidance enabling switch and electrical interfaces for end effectors. Laser scanners installed diagonally at the base of the MMO-700 platform ensures operator safety while movement and is used for navigation and localisation as well [8].

KUKA LBR iiwa is one of the most advanced collaborative robots available in the market. It is the world's first series-produced sensitive and human-robot compatible robot. Intelligent algorithms and force-torque sensors on each axis enable the robot to quickly react to external forces and take appropriate actions [9].In compliant mode, the robot acts as a spring-damper system with programmable stiffness [10] and can perform sensitive and delicate movements without creating crushing and shearing hazards. However utilizing these advanced features for useful applications in an easy to use manner still remains a challenge in terms of programming effort and robotics knowhow for regular factory workers.

“Clara” is integrated into the self-execution system (SES) of the LLF. The self-execution system at ESB LLF is an event-oriented, distributed, decentralized control method for autonomous decision making and distributed cooperation among smart entities in the factory [11]. The Neobotix mobile platform is configured for autonomous navigation through the vendor supplied GUI and provides and OPC-UA interface for remote controlling and monitoring. Map recording, path configuration, work station location, safe zone configuration and navigation task definition can be done using this GUI. A navigation task to a specific workstation is given to the platform through the SES.

3. Easy teaching solution for KUKA LBR iiwa

The desirable features of a powerful robotic programming solution are the capability of providing a high level of control, building flexible task plans from demonstrations, and support for low-level operations [12]. Creating a system which is easy and intuitive to use while keeping these properties is the a suitable solution for both advanced and normal users. Even though the existing KUKA programming solution utilising low-level Java API is powerful, it lacks the simplicity and easy programmability of high-level control.

3.1. Graphical modelling of robot tasks

The teaching solution developed at the ESB LLF is built upon the KUKA Application Framework (KAF) package, which assists users in modeling the robot's work sequences graphically using pre-programmed task blocks (see figure 2). These blocks make non-trivial robot operations easily accessible to the operator, enhance human-robot collaboration, and allow creation applications that cover the relevant application areas of cobots [13] [14].

Each KAF application has a start and end block. An application flow is controlled by connecting the results of one block to others. Data synchronisation and sharing of data among the blocks is achieved via input, output, and shared flow variables. Each block can have zero or many user configurable inputs. Inputs can be modified by the operator at runtime thus allowing the easy reconfiguration of an existing program. Results of a specific application block can be

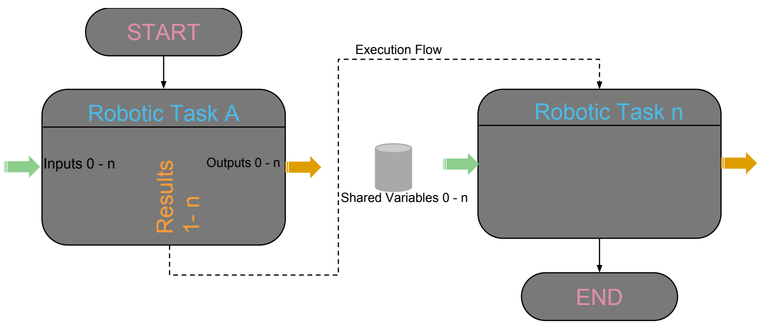

Fig. 2. Symbilic specification of KUKA application framework program.

used as in input of another block by connecting them via a shared variable.

3.2. Block level overview of configurable blocks

Application specific blocks are programmed with KUKA sunrise API by the robot engineers and abstract a specific robotic operation. Blocks are categorised into general robot operations, end effector control, coordination, and force feedback. See table 3.2 for description and application of each blocks. Coordination blocks implement a collaborative interaction between the operator and the robot. Error handling and specification of reactions is crucial in robot programming [15]. The robot engineer defines Block specific error outputs for each block. An error control flow is triggered in case of an error. This feature allows the robot operator to configure safe reactions to error states and create robust robotic applications as shown in figure 3.

Table 1. Details of developed blocks and their applications.

Block Name	Category	Description	Application
GripperOpen/Close	End effector Control	Control compatible grippers, perform error handling	Pick and Place, Assembly, logistics tasks.
MoveLin,MovePTP,MoveRel	General Robot movements	Move the robot to a waypoint taught by the operator.	All applications
HandGuiding	Coordination	Enables hand guidance mode till user button pressed	Assistant tasks, Robot aided Handling of heavy parts,Ergonomic positioning of tools and parts
PositionHold	Coordination	Robot keeps a certain pose compliant mode till user button is pressed or end effector is pushed.	Collaborative assembly, Inspection, Wait for Operator, Ergonomic positioning of tools and parts
ScrewDrive	End effector Control	Control block for Schunk screw driver system	Fixing, Screwing
SearchLinear	Force Feedback	Move TCP linearly in compliant mode till an insertion is detected	Parts insertion, Assembly, Tactile inspection
SearchSpiral	Force Feedback	Moves TCP in spiral shape in compliant mode till an insertion is detected.	Assembly, Parts insertion
SeekAttention	Coordination	Blinks status LED on media flange and wait for user interaction to continue.	Error Handling, Status display
HandleIO	General Robot operation	Generic block for controlling the ethercat IO module	External sensor, actuator integration

3.3. Teaching workflow

The KAF programs are created on a PC using eclipse based Sunrise workbench IDE. The operator creates the program by dragging and dropping the required blocks from the task palette, configuring input values and finally connecting the blocks to achieve high-level control flow and error handling. Created programs must be synchronised with the robot controller via a network connection. Now the operator can test the program and make adjustments to the waypoints using the KUKA smartPAD. Updated waypoints and user-defined inputs parameters must be synced back to Sunrise workbench PC.

4. Simulation using DELMIA

DELMIA is a software solution created and distributed by Dassault Systèmes. The goal of the software is to support companies worldwide in creating and redesigning their plant layout and factory flows. The primary focus is on quality, development and operating sequence. Modern companies heavily rely on a functioning coordination

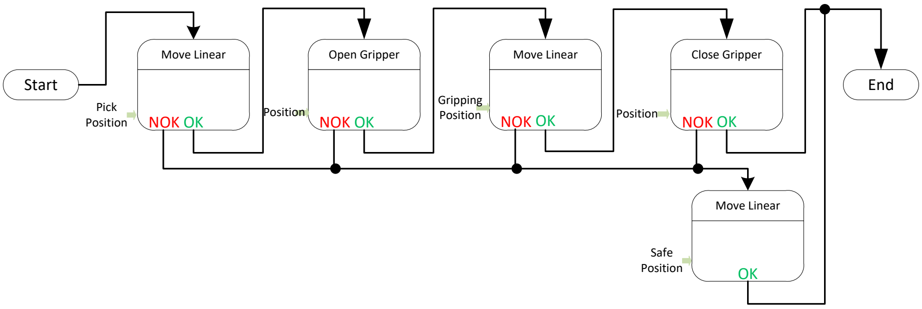

Fig. 3. Simplified representation of a pick operation using the developed blocks.

Fig. 4. Process overview of simulation using Delmia and easy teaching.

between construction, production, distribution, workforce, and processes. Those points are covered with the DELMIA software solution, therefore, enabling an efficient planning process starting with the workers and ending with the delivery process to the customer. A process is simulated and validated using the Delmia apps then it is programmed using the easy teaching approach and finally verified on the physical robot (see figure 4).

4.1. Setting up the Simulation

In the first step it is described how to set up the environment for your simulation. In the DELMIA App "Plant Layout Design" the "Resource Catalog" option is used to implement the required objects. The "Smart Snap" option allows to snap the objects in an easy manner and reduce the set up time. Once the objects are implemented the KUKA LBR iiwa can be imported using the "Import" function and the catalog containing the models of the KUKA robots "KUKACatalogKRC1". After the robot is imported the "Robot Simulation" App is used to teach the robot. This App gives the worker two different options. You can either deliver validated programs to the shopfloor for execution by creating production robot programs offline or upload already existing programs from the robot controller in order to validate or edit them. This paper will focus on the former. The Robot Programming app uses the visual basics format to import and export robot native language programs. This enables the worker to calibrate the workcell in order to match the real workcell.

4.2. Teaching the Robot

As the DELMIA app offers a wide variety of different programming options this section deals with how this process is done. A basic understanding of the 3D Experience Platform and programming with DELMIA is an important requirement for understanding the teaching progress. First of all, a simulation state has to be defined. A simulation state contains the positions and simulation properties of a resource at the point where the state is created. A simulation state defines a temporary state of the resource. The initial state of a simulation is the Nominal State and is the beginning of the simulation. After the Nominal State is defined further simulation states can be added to the simulation. Once the simulation states are defined, a motion controller has to be created. A motion controller is used when more than one resource is programmed within a workstation. The motion controller ensures that the resources operate as per planned sequence. In the next step, the necessary tools are attached to the robots and the motion profiles of the robots are defined. The motion profiles define the speed at which a robot performs its defined tasks. In the next step Tag points and tasks are added to the robot simulation. In DELMIA a tag point is an X, Y, Z location in space which determines the destination of the robot arm. A Task is a step by step movement that determines the order of motion characteristics for the robot motion. Tags have to be defined and then added to the task. If a task is programmed without Tags the robot will simply move to the destination without performing any action. Once tasks and tags are defined the simulation is ready to be run [16].

5. Application in the learning factory

With the above described technologies, an application scenario in the LLF is realised. "Clara" responds to a service request of an assembly station and drives there autonomously. Once the robot reaches its destination the operator starts the execution of the previously planned, simulated and programmed process steps. The easy teaching application allows the operator to quickly adapt Clara's behaviour. After the task is completed the robot drives back to its park position.

6. Conclusion and Prospects

This paper presented an easy teaching and simulation proof of concept for a mobile collaborative robot at the ESB Logistics Learning Factory. This allowed non programming experts to implement and test adaptive, flexible and integrated production scenarios at the LLF. Process simulation combined with the easy teaching method enables shop floor workers to fully utilize the potential of the existing robots without the need of being experts in programming. Decentralized decision-making benefits increase speed, quality, and flexibility. Even though the proposed easy teaching method is easier, the operator still spends a lot of time manually configuring and troubleshooting the program.

Based on the proof of concept, a generic system for manufacturer-independent semi-automated easy teaching based on an interdisciplinary research proposal at European level will be proposed in the next step. This research proposal aims for reducing barriers in deployment and operation of collaborative robots in hybrid working environments, develop artificial intelligence assisted teaching work flows and establish new application areas for collaborative robots.

References

[1] Martin Hägele, Walter Schaaf, and Evert Helms. Robot assistants at manual workplaces: Effective co-operation and safety aspects. In *Proceedings of the 33rd ISR (international symposium on robotics)*, volume 7, 2002.

[2] Eberhard Abele, Joachim Metternich, Michael Tisch, George Chryssolouris, Wilfried Sihn, Hoda ElMaraghy, Vera Hummel, and Fabian Ranz. Learning factories for research, education, and training. *Procedia CiRp*, 32:1–6, 2015.

[3] Sebastian G Brunner, Franz Steinmetz, Rico Belder, and Andreas Dömel. Rafcon: A graphical tool for engineering complex, robotic tasks. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3283–3290. IEEE, 2016.

[4] Monica N Nicolescu and Maja J Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM, 2003.

[5] Seung Han Kim and Jae Wook Jeon. Programming lego mindstorms nxt with visual programming. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 2468–2472. IEEE, 2007.

[6] James P Diprose, Bruce A MacDonald, and John G Hosking. Ruru: A spatial and interactive visual programming language for novice robot programming. In *Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on*, pages 25–32. IEEE, 2011.

[7] Zbigniew M Bzymek, Manuel Nunez, Mu Li, and Sean Powers. Simulation of a machining sequence using delmia/quest software. *Computer-Aided Design and Applications*, 5(1-4):401–411, 2008.

[8] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[9] Sami Haddadin, Alin Albu-Schäffer, and Gerd Hirzinger. Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research*, 28(11-12):1507–1527, 2009.

[10] Robert J Anderson and Mark W Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988.

[11] Jan Schuhmacher and Vera Hummel. Decentralized control of logistic processes in cyber-physical production systems at the example of esb logistics learning factory. *Procedia CIRP*, 54:19–24, 2016.

[12] Geoffrey Biggs and Bruce MacDonald. A survey of robot programming systems. In *Proceedings of the Australasian conference on robotics and automation*, pages 1–3, 2003.

[13] José Saenz, Norbert Elkmann, Olivier Gibaru, and Pedro Neto. Survey of methods for design of collaborative robotics applications-why safety is a barrier to more widespread robotics uptake. In *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering*, pages 95–101. ACM, 2018.

[14] Robotiq. Collaborative robots buyer's guide, 2018. *https://blog.robotiq.com/collaborative-robot-ebook*. (accessed 15 November 2018)

[15] Tomas Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71(7):821–841, 1983.

[16] 3d design & engineering software - dassault systèmes. *https://www.3ds.com*. (accessed 15 November 2018)