Self-configuration of a Robotic Platform to support a self-organized Manufacturing Process

1st Luis A. Estrada Jimenez
Centre of Technology and Systems
UNINOVA
Lisbon, Portugal
lestrada@uninova.pt

2nd David Sanderson
Institute for Advanced Manufacturing
University of Nottingham
Nottingham, NG7 2GX, UK
David.Sanderson@nottingham.ac.uk

3rd Jack C. Chaplin
Institute for Advanced Manufacturing
University of Nottingham
Nottingham, NG7 2GX, UK
Jack.Chaplin@nottingham.ac.uk

4th Jose Barata
Centre of Technology and Systems
UNINOVA
Lisbon, Portugal
jab@uninova.pt

***Abstract*—Self-configuration in manufacturing is a key trend to generate adaptable production systems. Different product requirements need different machine settings and continuous software update. Existing approaches usually assume that manufacturing resources (e.g. robotic platforms) have already a predefined set of available capabilities or that there is a centralized manager able to provide configuration updates. Centralized approaches are not always a suitable solution. Dynamic changes in production require continuous maintainability of the central server, constraining the shop-floor agility. New approaches should consider emergent self-configuration i.e. carried out at run time and decoupled from a centralized unit. In this work we present a framework for self-configuration of robotic platforms, where those are not explicitly preconfigured; instead, control parameters are transferred wirelessly from raw material to resources during production (once they arrive to the resource), following an intelligent product-driven manufacturing approach. A set of templates are proposed to generalize the sequential behaviour of manufacturing operations. Thus, manufacturing resources can read, use these parameters and store them for future operations. This framework is illustrated with an assembly operation using an educational robotic platform.**

***Index Terms*—Self-configuration, Self-organization, Smart manufacturing, Robotic platform, Flexible shop-floor**

I. INTRODUCTION

Automation is referred as the process (i.e. manufacturing) that occurs ”without the direct participation of a human worker” [1]. With the advent of the fourth industrial revolution, this concept has been enhanced with the emergence of technologies such as cloud computing or collaborative robotics with the aim of satisfying dynamic market demands and high product customization [2], [3]. As a driver of automation, self-organization can be defined as *a mechanism that enables a system to change its organization without explicit external command* [4]. In manufacturing this concept can be associated to the logical and sequential organization of tasks, physical organization of modules, transportation of materials, resources or end effectors [5], [6]. Emerging automation concepts consider mobility not just of the raw materials, but also tools, robots and machines [7], generating complexity in the management of operations, transportation and configuration.

Products with different requirements need different resource configurations. This configuration update is usually carried out manually which is not very effective considering current market dynamism. This has been partially solved by the introduction of flexible manufacturing systems or Supervisory Control and Data Acquisition (SCADA) systems. However, those usually operate under a predefined working envelope defined by the part family, reducing its capacity of adaptation. Thus, the need of having the ability of robotic platforms to self-configure considering new production specifications.

Several works have proposed self-configuration of manufacturing. In [8], a framework is proposed using agent technologies and a cloud pipeline looking for optimal configuration parameters. In [9] the self-configuration of a plug and produce system is proposed based on a service oriented workflow manager. Other works usually focus on the process configuration using agent technologies, ontological models [10]–[14] or web services [15]. In [16], a function-behaviour structure methodology is presented. It is complemented by an ontological model to support the adaptability and configuration of the production system.

Overall, most approaches assume that resources have already a predefined set of skills, or that there is a centralized platform in charge of orchestrating services or of providing new capabilities to them.

Increasing market dynamism and unforeseen events suggest that it is not always feasible to define in advanced behaviours that explicitly state how a manufacturing shop-floor should react or what capabilities resources should have. For example, during the Covid 19 pandemic and due to the shortage of ventilators, many automakers decided to enter the business of manufacturing critical medical equipment [17], changing their core business goal. Continuous maintainability and high level of dependence on the knowledge of a centralized unit can be considered a disadvantages, in case it fails the whole manufacturing process can be affected. Thus, the importance of resource self-configuration decoupled from a centralized system and that can be executed at run time (emergent) without extra updates and just driven by the requirements of the product (intelligent produce-driven manufacturing [18]).

We aim to close this gap by introducing a framework capable of self-configuring robotic platforms. This proposal starts with the assumption that a self-organizing process is being carried out, i.e. transportation of raw material towards a specific robotic platform. This transportation can be executed by Automated Guided Vehicles (AGVs). This is motivated as an answer for the self-management of highly flexible shopfloors i.e. the matrix production, a concept developed by KUKA where the logistics and production components are decoupled [19]. In the matrix production, a pool of AGVs are responsible for the handing of end effectors and raw material to the production cells (logistics self-organization). This enables very versatile production and the capacity to convert itself on the fly according to new production requirements [19].

The scope of this paper is centered in the development of a framework for self-configuration of robotic platforms under an intelligent product-driven manufacturing context i.e. the software integration aspects. The product has the knowledge of what operations, parameters and configuration it needs (in the case of the matrix production, the AGV with the product and raw material). Also, the product can transfer these parameters to the robotic platforms. For a simplified understanding we will focus on assembly operations. See Fig. 1 where this idea is showcased.

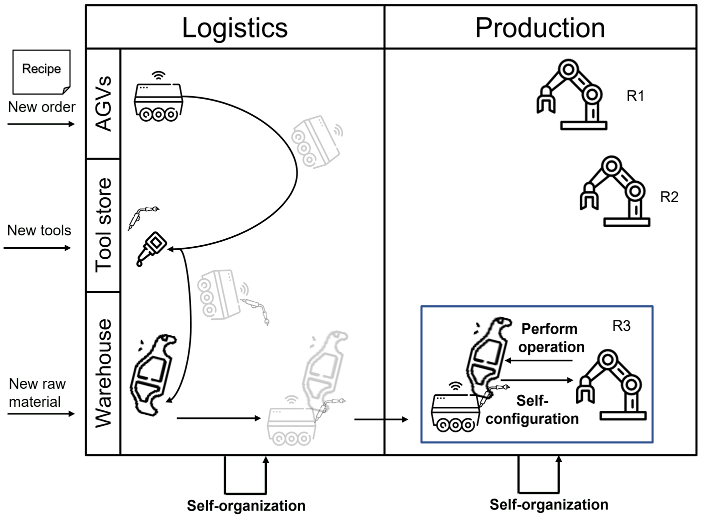


Fig. 1. Sketch of the matrix production concept. Self-configuration to support logistical and production self-organization

The rest of this paper is structured as follows. In Section II we describe the components of the framework, a product process flow representation and a sequential diagram of the self-configuration logic. Section III presents the application of this concept in the self-configuration of an assembly operation. Finally, Section IV describe some conclusions and future works of the approach.

II. A FRAMEWORK FOR SELF-CONFIGURATION OF A ROBOTIC PLATFORM

A. Framework components

The framework includes three main components: (1) the logistic unit (AGV) with raw material, end effectors and product recipe (representation of the product process flow, discussed in detail in part B of this section), (2) Robotic intelligent unit and (3) a robotic platform. Fig. 2 presents an overview of the framework. Below some details of the components of the self-configuration framework are presented:

(1)Logistics unit (AGV) with raw material, tools and recipe: The logistics unit will take the raw material in a carrier. The assembly operation (process flow representation) is stored as instructions in an Internet of things (IoT) device e.g. a Radio-frequency identification (RFID) card placed on the carrier. It has information of tasks to be performed or already performed in the raw material. The logistic unit can take also necessary tooling (e.g. fixtures, grippers) with the materials, or on separate ones.

(2)Robotic intelligent unit: It is composed of six sub-components. They allow its integration and bidirectional communication by receiving necessary parameters from the carrier and establishing a connection with the robot controller. The sub-components are:

- IoT device: Device that acts as a gateway. It reads information of tasks and parameters from the carrier and transfer them to the platform integration of the Robotic intelligent unit.
- Basic templates for assembly operations: Composed of a sequence of instructions of the robot i.e. tri-dimensional movement of the end effector and gripping. These predefined templates can be continuously re-used for generic assembly operations e.g. picking and placing, screwing, welding, etc. Different set of templates can be generated according to the model and brand of the robotic platform. Templates can be also created by design depending on the process and product, increasing its scalability.
- Physical restrictions: Describe the physical restrictions of the robotic platform e.g. maximum range of the end effector or maximum speed of movement.
- Robotic framework: A software library that allows the connection and communication of the platform integration with the robotic platform e.g. python script, Robotic operating system (ROS), etc. Different robotic frameworks can be implemented according to the model and brand of the robotic platform.
- Data base of learnt skills: While developing the process self-configuration, specific tasks and parameters will be stored. Those can be reused in future applications.
- Platform integration: Integrates the acquired parameters, templates and physical restrictions to execute required tasks of the product.

(3)Robotic platform: Executes the assembly tasks according to control operations provided by the Robotic intelligent unit.

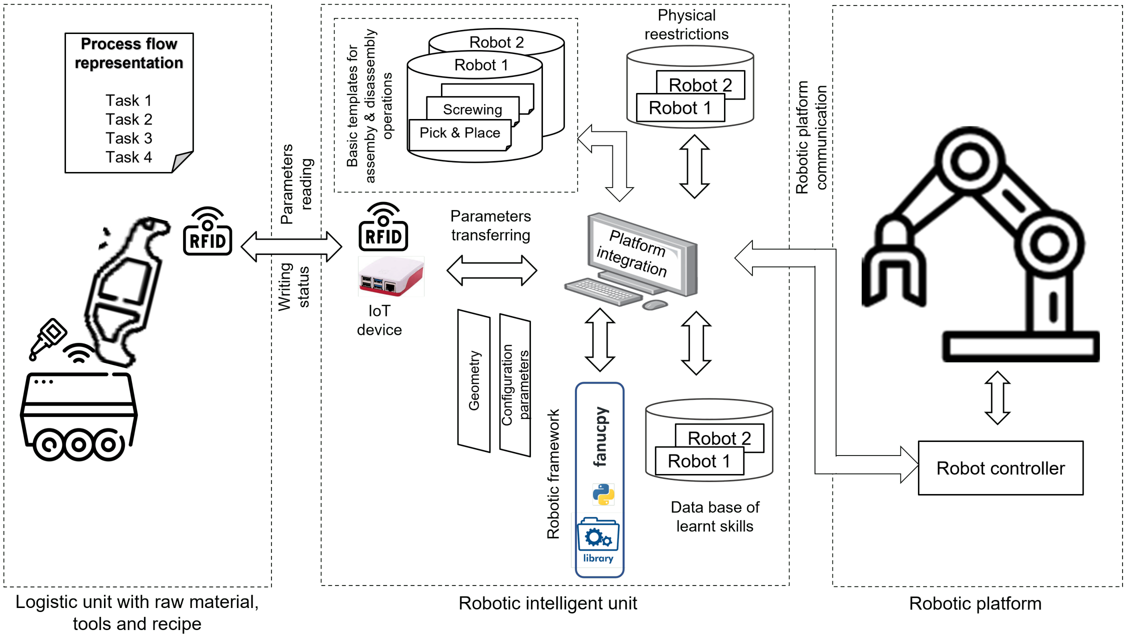


Fig. 2. Framework for manufacturing self-configuration of a robotic platform

B. Process flow representation

The process flow represents a sequence of steps necessary to carry out an assembly operation.

It contains a recipe of tasks that have to be executed in the raw material. This knowledge modelling is not a simple task. Inspired by the work of [20] we proposed a simplified representation of this process considering following elements:

- *Process flow:* It has all required sequential information to assembly a product. It is composed of at least one task.
- *Task:* This represents a fundamental assembly operation e.g. picking and placing, screwing, welding, etc. Individual tasks consist of finer activities called process steps.
- *Process step:* It is ”an elementary operation of a process flow which cannot be further decomposed into sub steps” [20]. Process steps can therefore be wrapped as a functional configuration of a robotic platform and can be represented as a set of attributes. Examples of process steps are: end effector moving, gripping, etc.

A sketch of a product process flow and its components is presented in Fig. 3(a) , Fig. 3(b) presents the composition of a process step as well as two examples for moving and gripping. Fig. 3(c) presents a sequential representation of a task as a composition of process steps i.e. picking and placing and screwing.

C. Sequential diagram

The process flow connection and communication describe the sequence of steps required to transfer necessary parameters to the robotic platform. Here the main elements are described as software blocks similar to an event driven approach. Once a new product arrives, the parameter transference starts and afterwards the assembly operations are executed. These interactions are displayed in Fig. 4. Main steps of the sequential diagram are summarized below:

- *Requesting connection:* The sequence starts when the logistics unit with the raw material establish a connection with the controller of the robotic platform. This digital connection triggers a compatibility checking test to make sure the robotic platform is able to perform the assembly operation(s), otherwise the connection will not be established. After a successful connection, the process flow is extracted and immediately sent to the Robotic intelligent unit.
- *Querying and storing skills/parameters:* After receiving the process flow, if the set of tasks and attributes are new (i.e. because of a new product/product variant), those will be automatically stored in the Robotic intelligent unit to be used in future operations.
- *Querying templates for assembly operations:* Specific tasks and parameters (skills) will be matched according to a set of templates available in the Robotic intelligent unit. Those will provide the sequential configuration flow necessary to execute the task.
- *Requesting control function from robotic framework:* Templates with specific parameters will use the robotic framework as a gateway for communication with the robotic platform.
- *Executing operation:* After matching parameters and templates with the robotic framework, those will be executed in the robotic platform.
- *Updating product status and closing connection:* When all tasks that the robotic platform is able to do are executed, the status of the product will be updated (in

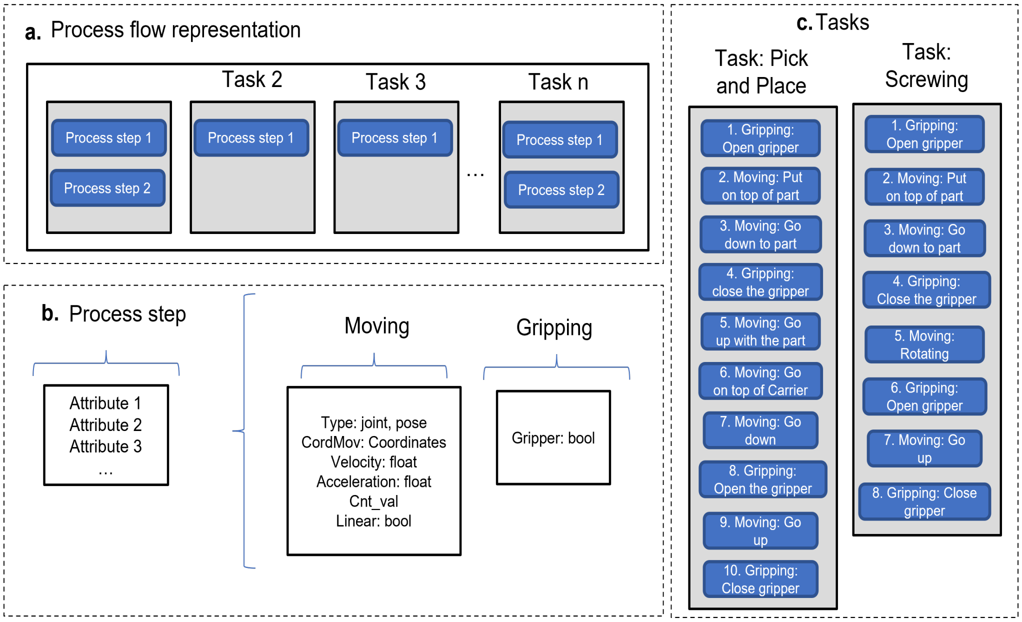


Fig. 3. Representation of: (a) product process flow, (b) process step, and (c) tasks

terms of completed tasks). After that the connection will be closed.

III. USE CASE: SELF-CONFIGURATION OF AN ASSEMBLY OPERATION

The use case implemented in this work includes the software development for the self-configuration of a specific robotic platform. To illustrate this concept, a simple product was chosen. i.e. a hinge. Hinges are used to join two parts together while creating a revolut joint between them. Four main components compose this example: pin, inferior leaf (part C), superior leaf (part B), and a screw (part A). Objective of this assembly operation is to pick and place the inferior leaf, the superior leaf and the screw on the pin. After the screw is placed, a screwing operations is carried out. See Fig. 5 for a 3D representation of the parts and assembly operation.

A. Software implementation

The software implementation consisted in the development and integration of four python scripts and a JSON file, the last represents the assembly process flow. JSON was chosen for simplicity in the data manipulation. Other formats to consider that may facilitate the interoperability and standardization of the process flow are: AutomationML (AML) or Business to Manufacturing Mark up Language (B2MML). The Robotic platform used in this work is the the FANUC Educational cell (R-30iB Mate Plus Controller).

In this example we assume that the recipe has been already extracted by the IoT device (Fig. 2) and therefore it is already available in the Robotic intelligent unit. Software scripts implemented in python are:

- *Tasks, process steps and parameters extraction:* It is used to extract key parameters from the recipe (JSON file)
- *Assembly templates:* Set of functions that represent the pre-built sequence of steps.
- *FANUC robotic framework:* Script that allows the connection and communication with the robotic framework. In this example the fanucpy: Python package for FANUC industrial robots was used [21], [22]. This driver has been tested in KAREL and FANUC teach pendant languages. By establishing a connection with the server of the robot controller, control and monitoring variables can be easily accessed. It has already predefined functions that facilitate the moving, opening and closing of the gripper of the robotic platform (FANUC educational cell).
- *Platform integration:* Script that integrates the assembly parameters, templates and communication framework to execute the movement of the robot.

B. Results and Discussion

Two recipes (JSON files) were implemented to test the scripts developed. One of them describing the assembly process of the hinge and the other one its disassembly. The last one can be very practical in terms of re-usability and recycling of products (they have the knowledge of how to dissemble themselves).

Fig. 7a shows a hinge assembly process and Fig. 7b shows its disassembly after the parameters self-configuration using the FANUC educational cell.

Compared to other research works where there is a predefined set of skills or capabilities (usually manually predefined), or a centralized server that orchestrates different parameters, in this concept the self-configuration occurs in run-time. This can drastically reduce configuration effort and time when new products are launched. As far as the manufacturing resource has physical capabilities to perform an operation and the set

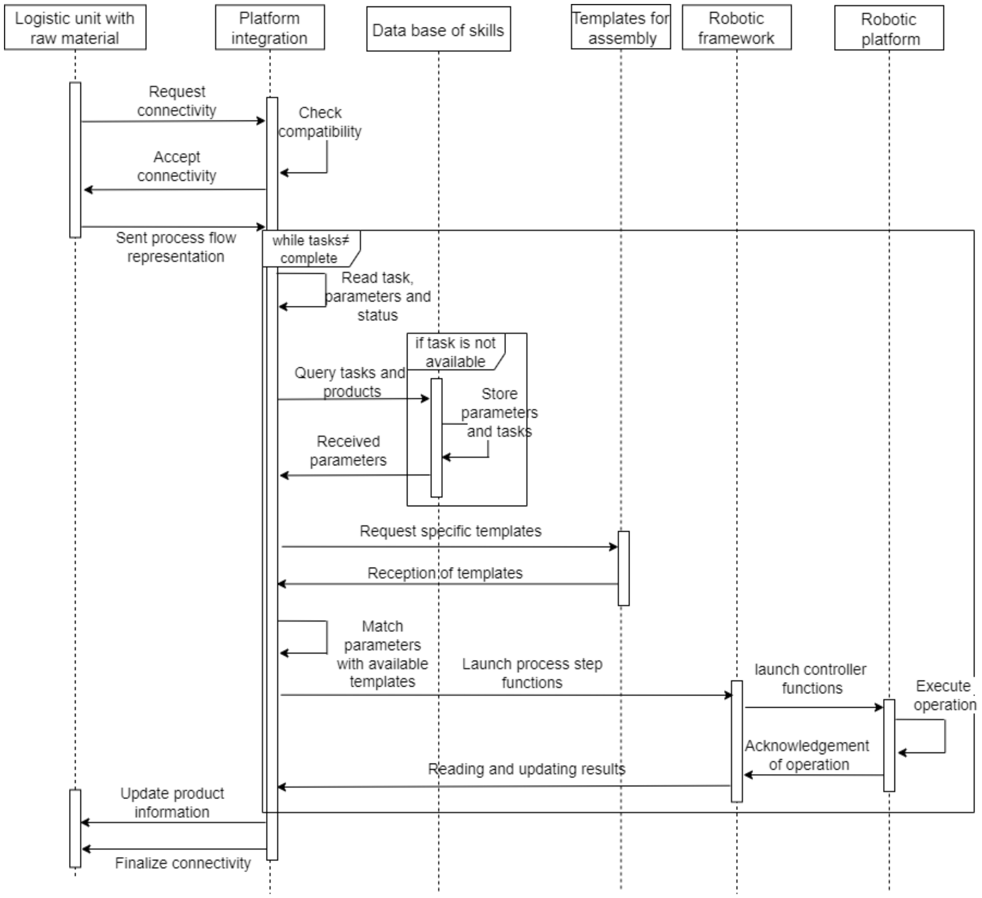
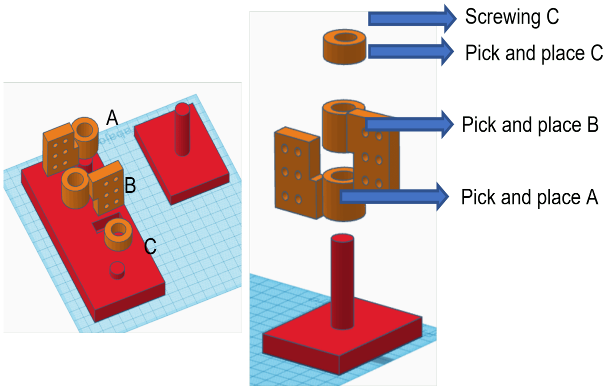


Fig. 4. Sequential diagram proposed for the self-configuration framework



Parts in carrier

Assembly process

Fig. 5. Sequential steps for the assembly of a hinge

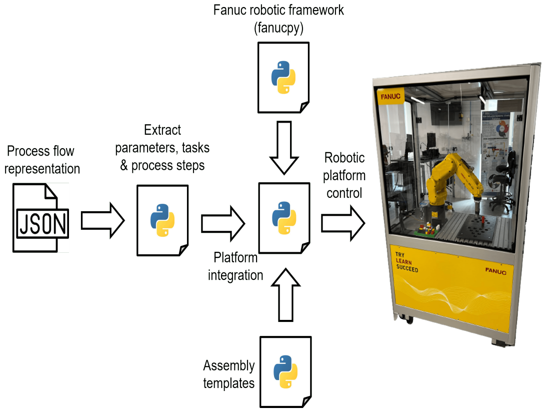


Fig. 6. Software implementation of the self-configuration framework

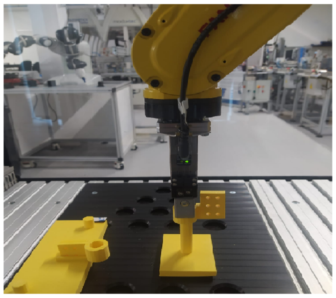
of templates available, the production possibilities (in terms of product variability) can be drastically enhanced.

IV. CONCLUSIONS AND FUTURE WORKS

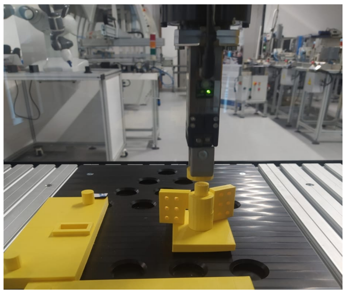
Current paper presents a framework for manufacturing self-configuration to support the self-organization of manufacturing systems (specially intra logistics self-organization).

Main idea behind this approach is that manufacturing resources (i.e. robotic platforms), do not need to be programmed in advance to perform a specific task. They just need generic templates of assembly operations.

While the assembly process is carried out, new parameters are transferred to the robotic platform. Therefore, there is no need of digital connection to a centralized server or need of having an operator continuously re-configuring the parameters



a. Hinge assembly



b. Hinge disassembly

Fig. 7. Implementation of the self-configuration framework using the FANUC educational cell in the (a) assembly and (b) disassembly of a hinge

when new products or product variants are launched.

There is a continuous self-configuration, and robotic platforms can permanently learn and store new tasks.

The main contributions of this work are: the development of the framework and its logical sequential representation, modelling of the process flow, concepts of templates for assembly operations, integration and development of the software as python scripts, and a show case using a robotic platform for assembly and disassembly of a hinge.

Future works will consider additional development of the framework i.e. rules that describe physical restrictions of the robot (range of parameters that the resource accepts e.g. velocity, acceleration, maximum gripper opening), testing and comparison with other robotic frameworks e.g. ROS, the integration with other complementary technologies e.g. vision systems, and the test with other robotic platforms. A knowledge model will be constructed that enabled the intelligent selection of resources. Also, a more comprehensive case study showing also reconfiguration aspects of AGVs will be included as this work focuses more on software integration aspects. Finally, mechanisms to encrypt sensitive assembly instructions should be investigated as a way to protect trade secrets when manufacturing new products.

ACKNOWLEDGMENT

This work is carried out under DiManD Innovative Training Network (ITN) project funded by the European Union through the Marie Sktodowska-Curie Innovative Training Networks (H2020-MSCA-ITN-2018) under grant agreement number no. 814078.

REFERENCES

[1] M. P. Groover, *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India, 2016.

[2] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *Cirp Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[3] L. A. Estrada-Jimenez, T. Pulikottil, N. N. Hien, A. Torayev, H. U. Rehman, F. Mo, S. N. Hojjati, and J. Barata, "Integration of cutting-edge interoperability approaches in cyber-physical production systems and industry 4.0," in *Design, Applications, and Maintenance of Cyber-Physical Systems*. IGI Global, 2021, pp. 144–172.

[4] G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos, "Self-organization in multi-agent systems," *The Knowledge engineering review*, vol. 20, no. 2, pp. 165–189, 2005.

[5] L. A. Estrada-Jimenez, S. Nikghadam-Hojjati, and J. Barata, "Characteristics of adaptable control of production systems and the role of self-organization towards smart manufacturing," in *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2021, pp. 39–50.

[6] L. A. Estrada-Jimenez, T. Pulikottil, R. S. Peres, S. Nikghadam-Hojjati, and J. Barata, "Complexity theory and self-organization in cyber-physical production systems," *Procedia CIRP*, vol. 104, pp. 1831–1836, 2021.

[7] R. Berger, "Rise of the machines – how robots and artificial intelligence are shaping the future of autonomous production," 2019.

[8] H. U. Rehman, J. C. Chaplin, L. Zarzycki, and S. Ratchev, "A framework for self-configuration in manufacturing production systems," in *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2021, pp. 71–79.

[9] S. Scheifele, J. Friedrich, A. Lechler, and A. Verl, "Flexible, self-configuring control system for a modular production system," *Procedia Technology*, vol. 15, pp. 398–405, 2014.

[10] H. Tang, D. Li, S. Wang, and Z. Dong, "Casoa: an architecture for agent-based manufacturing system in the context of industry 4.0," *IEEE Access*, vol. 6, pp. 12 746–12 754, 2017.

[11] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The ideas project: plug & produce at shop-floor level," *Assembly automation*, 2012.

[12] R. Frei and G. D. M. Serugendo, "Self-organizing assembly systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 885–897, 2011.

[13] G. Rzevski, P. Skobelev, A. Zhilyaev, O. Lakhin, I. Mayorov, and E. Simonova, "Ontology-driven multi-agent engine for real time adaptive scheduling," in *2018 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*. IEEE, 2018, pp. 14–22.

[14] A. D. Rocha, J. Tripa, D. Alemão, R. S. Peres, and J. Barata, "Agent-based plug and produce cyber-physical production system–test case," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 1545–1551.

[15] Y. Zhang, C. Qian, J. Lv, and Y. Liu, "Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 737–747, 2016.

[16] D. Sanderson, J. C. Chaplin, and S. Ratchev, "A function-behaviour-structure design methodology for adaptive production systems," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3731–3742, 2019.

[17] I. Lawrence Ulrich, "Automakers pivot to produce ventilators, respirators, and face masks," 2019.

[18] D. McFarlane, S. Sarma, J. L. Chirn, C. Wong, and K. Ashton, "The intelligent product in manufacturing control and management," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 49–54, 2002.

[19] KUKA, "Matrix production: an example for industrie 4.0," 2019.

[20] V. Hammerstingl and G. Reinhart, "Skills in assembly," 2018, version 1.1. [Online]. Available: https://mediatum.ub.tum.de/1428286

[21] A. Torayeff, "fanucpy: Python package for fanuc industrial robots," 2022, version 0.1.5. [Online]. Available: https://github.com/torayeff/fanucpy/

[22] A. Torayev, G. Martínez-Arellano, J. C. Chaplin, D. Sanderson, and S. Ratchev, "Towards modular and plug-and-produce manufacturing apps," 2022.