Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 103 (2021) 158–163

www.elsevier.com/locate/procedia

9th CIRP Global Web Conference – Sustainable, resilient, and agile manufacturing and service operations : Lessons from COVID-19

Towards planning and control in cognitive factories - A generic model including learning effects and knowledge transfer across system entities

Marco Wurster*[a], Yannick Exner[a], Jan-Philipp Kaiser[a], Nicole Stricker[a], Gisela Lanza[a,]

[a] *Karlsruhe Institute of Technology, Institute of Production Science, Kaiserstraße 12, 76131 Karlsruhe, Germany*

* Corresponding author. Tel.: +49 1523 950 2599 ; *E-mail address:* marco.wurster@kit.edu

Abstract

Cognitive abilities allow robots to learn and reason from their environment. The gained knowledge can then be incorporated into the robot's actions which in turn affect the environment. Therefore, a cognitive robot is no longer a static system that performs actions based on a pre-defined set of rules but a complex entity that dynamically adjusts over time. With this, challenges arise for production systems that need to observe and ideally anticipate the cognitive robot's behavior. Often, digital twins are employed to test and optimize production control systems. This paper presents a generic approach to characterize, model and simulate learning processes and formalized knowledge in hybrid production systems assuming different station types with learning effects. Thereby, quantitative and qualitative learning processes are mapped including knowledge sharing and transfer across entities. A modular and parameterizable design enables the adjustment to different use cases. Eventually, the model is instantiated as a digital twin of a real production system for product disassembly employing cognitive-autonomous robots among human operators and rigidly automated machines. The model shows great potential to be integrated into test beds for planning and control systems of cognitive factories.

Keywords: Learning Effects; Cognitive Robots; Digital Twin; Hybrid Production Systems; Disassembly

1. Introduction

With the trend towards individualization and ever shorter product life cycles, the number of variants in production increases rapidly [1]. In remanufacturing, these challenge is complemented by uncertain product conditions, quantities and yield during end-of-life product disassembly. [2]. In such demanding conditions, conventional automated production systems reach their limits since adapting them to changing requirements is still time-consuming and costly. Due to the special requirements of remanufacturing, automation in industrial disassembly systems could not prevail yet [3]. In fact, these systems are merely based on manual labor due to the flexibility and the ability of a human operator to learn and anticipate new situations [4].

With the emergence of robotics and cognitive abilities in robotics, automated production systems might no longer be static systems but much more dynamic systems that can adapt over time. These systems (partially) consist of cognitive robots that can learn and perform new tasks autonomously in a similar way human operators do. Therefore such systems can seamlessly adapt to new requirements which makes them able to deal with new products and small lot sizes at competitive costs [5]. For their productive operation, however, the dynamics of the systems must be handled and controlled appropriately. Suitable production planning and control (PPC) approaches must be developed to fully utilize the new-gained potential.

In this work, the problem is approached by the development of a digital twin as a proven means of testing and optimizing production systems even before they are launched. Focusing on production systems that are constantly confronted with new variants and unknown operations, the degree of gained knowledge by human operators and robots has a big impact on the systems' performance [6]. Considering learning effects in the planning stage is therefore considered crucial to increase accuracy in PPC [7]. Because of that, a generic model is proposed that accounts for diverse learning effects and knowledge representation. Two types of learning effects are distinguished: (1) learning curve-based processes allowing knowledge transfer in the production system, and (2) learning processes in which new, possibly more efficient, disassembly sequences are learned and adopted. When instantiated, the model can cover various

10.1016/j.procir.2021.10.025

use scenarios of production systems. Furthermore, a production control logic is deployed to be confronted with the dynamic system. Both, the digital twin as a discrete-event simulation model as well as the control logic are implemented in Python using the SimPy Framework. They built on and extend an existing shopfloor model by Kuhnle [8].

2. Related Work

While not referring to knowledge, Wright first described organizational learning curves as a connection between the decline in unit labor cost for airplanes and their cumulative output [9]. For a given level of average labor costs, the resulting cost is the multiplication of the given labor cost with a certain factor, the so-called learning rate. The unit labor cost y for the x-th unit is determined via $y = a * x^{-b}$, where a is the average unit labor cost of the first produced unit and b is a parameter for objective reduction as the cumulative output x increases [9] [10]. Variations of the learning curves exist [11] [12].

In the last decades, the number of publications on learning curves in production and operations management increased exponentially. While numerous modeling approaches, empirical studies and practical applications can be found, the vast majority focuses on human operators and manual work [7]. In fact, there is only one approach by Li et al. known to the authors, that focuses on dealing with robotic learning effects from a quantitative perspective [6]. The authors propose a line balancing solution considering learning effects of learning robots using backward induction rules.

In general, the idea of production systems that combine autonomous learning subsystems and human operators is widely discussed. Zaeh et al. are the first to describe the "cognitive factory" as automated technical systems with cognitive abilities in production [13]. Bannat et al. outline the demand for highly responsive and adaptive production systems, motivate the paradigm of cognitive factories and address new challenges on PPC [5]. Zaeh et al. propose an adaptive production control approach for cognitive factories that utilizes artificial cognitive capabilities [14]. From a more technical perspective, Vongbunyong et al. present a concrete approach to the utilization of cognitive robotics and learning by demonstration for end-of-life product disassembly [15] [16] [17]. In their demonstrator, robots learn to disassemble LCD screens by reasoning and emulating the behavior of human operators. No prior knowledge is assumed. The authors prove the general concept of robotic learning, learning by demonstration from human operators and knowledge transfer.

It can be concluded that existing work either investigates learning effects by human operators, follows a conceptional cognitive factory approach or focuses on enabling learning and knowledge transfer from a process, but not a planning perspective. Learning effects of both, human operators and cognitive robots, as well as the exchange of knowledge and possible dependencies are not combined or investigated quantitatively. In the future, this will be an important prerequisite for accurate system planning and control and therefore pressure the need for extended learning effect integration in production planning systems.

This paper presents an approach to model various learning effects in production systems in a quantitative way. Thereby, conventional learning curves of human operators are complemented in a model with learning effects of cognitive robots and knowledge sharing across system entities. First, learning processes and respective system entities are defined, categorized and mapped generically (Section 3). Afterwards, a cognitive disassembly system is instantiated and the approach is tested using a plain production control logic (Section 4). Eventually, the paper is summarized and concluded (Section 5).

3. Learning effect modelling approach

Learning is defined as a change in knowledge $\Delta know_{s,o}$ of operation o at station s. A station is a self-contained entity, like a manual workplace or a robot-based system that can perform certain operations according to their capabilities and knowledge. In this paper, knowledge is assumed as an abstract dimension and a simple representation of product and process-specific know-how of each station focusing on the operational effect from a quantitative planning perspective. While, from a technical perspective, knowledge representation differs severely across entity types, ranging from implicit experience of operators to different formalization approaches of robot skills, the knowledge value is assumed generally applicable for each station of each type. It is defined as an arbitrary value $know_{s,o} \in [0, 1]\ \forall s \in S, o \in O$ in this model, where S is the set of production stations and O the set of operations. For multi-product cases, operations are either defined exclusively for each product or the representation is enhanced by another dimension, e.g. index $p \in P$ for the set of product types leading to $know_{s,o,p}$. Strikingly, a value of $know_{s,o} = 0$ indicates complete absence of product or process know-how at s and the disability to perform o respectively. This is the case if, e.g. an autonomous technical entity has never performed and could not observe another entity perform o or similar operations before. Vice versa, $know_{s,o} = 1$ indicates perfect knowledge. This means, e.g. an operator or a learning robot figured out the optimal process and is able to perform an operation in nominal duration. During the learning phase, towards $know_{s,o} = 1$, operations are performed with partial knowledge, e.g. at $know_{s,o} = 0.5$. In this case, product and process knowledge is incomplete, favorable sub-processes or there sequence are still unknown, increasing the execution time. Knowledge-related operational failures can be caused, meaning operations can fail due to a skill deficit.

Besides, knowledge thresholds define the minimum knowledge $know_{s,o,min}$ required of a station s to be able to perform o. On that note $know_{s,o} \geq know_{s,o,min}$ is a necessary but not a sufficient condition for s to perform o successfully.

Two possible ways for a station to gain knowledge are considered:

- **Intra-station learning** refers to the knowledge $know_{s,o}$ that is gained within a station s by executing operation o.
- **Inter-station learning** describes the knowledge sharing capability as knowledge can be transferred from one station s' to another s (e.g. a human operator observed by a vision system, whose behavior and skills can be reproduced by a robot via imitation learning. Techniques, examples and further reading on how knowledge and skills can be transfered from human operators to autonomous robots are given by Billard et al. [18]).

From another perspective, two types of learning can be differentiated: **qualitative learning** which refers to the gained capability to perform an operation (enabling) and **quantitative learning** (intra-station and inter-station) which improves task times, process and object knowledge or the process execution quality. In some cases, they can become indistinct, as e.g. quantitative learning can lead to the qualitative process of enabling. However, the proposed taxonomy, as displayed in Figure 1, attempts to categorize and provide a general overview of how learning types are distinguished and examined in the following.

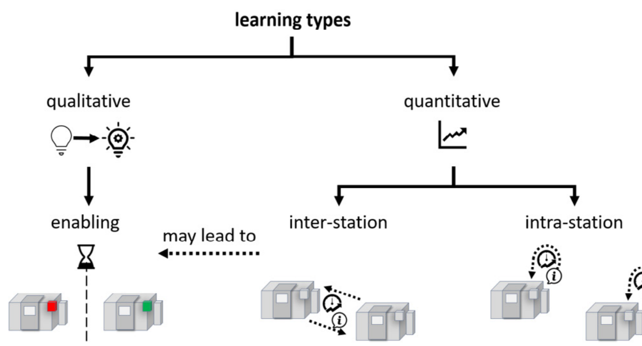

Fig. 1: Taxonomy of learning types

3.1. General learning procedure

Generally, it is assumed that executing operations time after time increases knowledge and decreases task times. Both are calculated similarly using an adapted approach of a Wright learning curve. However, since impacting failure rates and station capabilities, the focus in the following will be on the knowledge model.

Operation knowledge is captured by quantitative learning and modeled via the cumulative amount of operation executions per station s and per operation o based on an adapted version of the Wright learning curve.

The actual knowledge $know_{s,o}(x_{s,o})$ of o at s after a certain amount of trials $x_{s,o}$ of o at s, which is bound by the maximum of 100%, is calculated via:

$$know_{s,o}(x_{s,o}) = min(1, a^{know}_{s,o} * x_{s,o}^{-log_2(b_{know})}) \tag{1}$$

with a variable knowledge factor $a^{know}_{s,o}$ indicating the knowledge level at first execution of o at s and a given learning rate b_{know}. While $x_{s,o}$ is increased by executing o at s (**intra-station learning**), $a^{know}_{s',o}$ can only be manipulated by executing o at $s' \in S \setminus s$ (**inter-station learning**). Given the arbitrary nature of the knowledge definition ($know_{s,o} \in [0, 1]\ \forall s \in S, o \in O$), setting the initial value of knowledge is determined based on the Wright learning curve and an externally provided estimate of the cumulative amount of executions $n_{s,o}$ of operation o until a station s reaches 100% knowledge by intra-station learning. Given the learning rate b_{know}, the initial knowledge $a_{s,o}$ is therefore derived via:

$$a^{know}_{s,o,init} := n_{s,o}^{log_2(b_{know})} \tag{2}$$

3.2. Incorporation of inter-station learning

In this approach, the impact of inter-station learning is accounted by a variable $a^{know}_{s,o}$ that can be manipulated by gaining knowledge at stations of different types, without interrupting the conventional learning process according to Wright and the possibility for discounted knowledge gains when intra-station learning has already progressed.
Conversely $a^{know}_{s,o} = a^{know}_{s,o,init} = const.$ can be assumed when neglecting inter-station learning. The current state of knowledge will only depend on $x_{s,o}$. However, if inter-station learning is assumed on the other hand and station s receives knowledge from a station s', $a^{know}_{s,o}$ in equation 1 is updated according to:

$$a^{know}_{s,o,new} = a^{know}_{s,o} + \Delta a^{know}_{s',s} \tag{3}$$

with $\Delta a^{know}_{s',s}$ as the increase of knowledge for a (hypothetical) $know_{s,o}(x_{s,o} = 1)$. $\Delta a^{know}_{s',s}$ is thereby a modular term that depends on the learning interdependency of the entity pair.
In this approach we assume $\Delta a^{know}_{s',s}$ to be determined by a function of the current knowledge $know^{curr}_{s,o}$. More particular, it is assumed that the amount of knowledge that can be gained from station s' approaches 0 as $know^{curr}_{s,o}$ approaches its maximum. Furthermore, it is assumed that the knowledge gain will not follow a pre-determined curve but is stochastically distributed. Therefore, a random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ is assumed for $\Delta a^{know}_{s',s}$ and bound between $[0, 1]$. The expected value μ and standard deviation σ^2 as functions of the knowledge $know^{curr}_{s,o}$ at the current level of operation trials $x^{curr}_{s,o}$ are calculated as follows:

$$\mu\ (know^{curr}_{s,o}) = \sigma^2\ (know^{curr}_{s,o}) = \eta_{1,s',s} * e^{-\eta_{2,s',s} * know^{curr}_{s,o}} \tag{4}$$

with a knowledge transmission effectiveness of $\eta_{1,s',s}$ and a knowledge loss modelling factor $\eta_{2,s',s}$. How $\Delta a^{know}_{s',s}$ relates to $a^{know}_{s,o,curr}$ is emphasized in Figure 2:

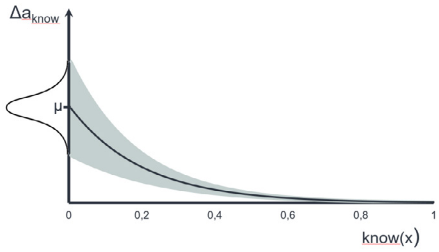

Fig. 2: Knowledge improvement effectiveness Δa_{know} through inter-station learning as a stochastic corridor

Both $\eta_{1,s',s}$ and $\eta_{2,s',s}$ are externally provided parameters with $\eta_{1,s',s} \in [0,1]$ and $\eta_{2,s',s} > 0$. While $\eta_{1,s',s}$ indicates how effective knowledge is collected at a station by a another receiving entity, $\eta_{2,s',s}$ describes how much this effect diminishes with prior knowledge. This approach is assumed as a simplified model of an observed manual station (s') - cognitive robot (s) pair of learning entities. Besides the proposed approach for $\Delta a^{know}_{s',s}$, other functions can be deployed as well to map various other types of inter-station learning effects. On a side note, in the particular case of two or more entities of the same type with a common knowledge base, such as two identical robots, the mechanisms of intra-station learning apply.
Another important concept that combines inter-station with qualitative learning is the concept of enabling. System entities that collect knowledge $know_{s,o}$ via inter-station learning but require a certain amount of knowledge $know^{activ}_{s,o} \in [0,1]$ are enabled as soon as a certain knowledge threshold $know^{activ}_{s,o}$ is reached, meaning $know_{s,o} \geq know^{activ}_{s,o}$. Some entities might conduct intra-station learning after they have been enabled via inter-station knowledge transfer. Different entity types might be enabled in a binary fashion and won't improve further, neither through intra- nor inter-station learning, after qualification. In a real-world application, this might be an automated station that needs to be installed by an automation technician to be able to execute an operation. This setup procedure will happen once a certain level of knowledge is present in the system.

3.3. Evolving precedence relation knowledge

As a second capsuled concept, cascading qualitative learning is incorporated, i.e. introducing new operation sequences at a certain point in time. The Petri-net-based approach (according to [19]) determines the workflow and precedence criteria for every product but allows for logical OR transitions where various routing options are valid. In real life, these options might be introduced to the demonstration-based cognitive system by trial-and-error or operation executions of human operators that follow different workflows. Whenever such an observation is taken by the cognitive system, it gets integrated into the knowledge base (i.e. the Petri net precedence graph) of the system. The system representation of the Petri net is therefore dynamic as it adjusts over time.

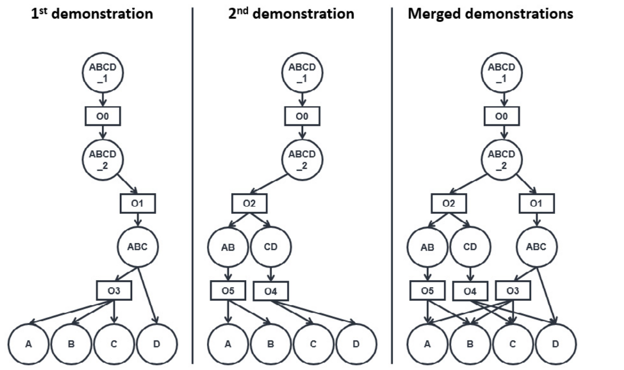

Fig. 3: Application of Petri nets to map dynamic precedence knowledge: derivation of a master graph via fusion of two single demonstrations

Figure 3 shows three different Petri-net-based disassembly precedence graphs for a product consisting of four components (A, B, C, D). On the left side, there is an initial workflow that might be known to the system from the beginning. At some point in time, a second workflow, which is displayed in the middle, might be introduced. Both are then merged to form a master precedence graph by adding the corresponding places and transitions via logical OR as displayed on the right side of the figure.

4. Instantiation of the model and testing

4.1. System entities

Given the generic design of the approach, various station types with individual properties can be defined to model the specific realities of a production system and allow for scalability. However, in the following, the approach will be instantiated on a specific set of distinctive station types which collectively form a coherent hybrid production system. Figure 4 gives a graphical overview of their setup and connectivity.

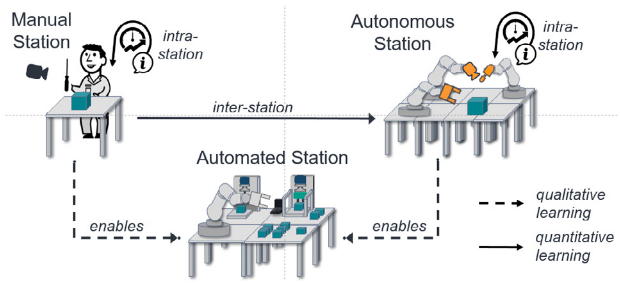

Fig. 4: Idealization of learning processes in the assumed production system with three station types

1. **Manual stations** are stations where a human operator follows or determines a workflow. Learning by demonstration is conducted through autonomous observation of the operator's actions via an RGB-D camera that captures used tools and the operator's movements to identify operations and determine a workflow.

2. **Autonomous stations** consist of cognitive robots that are capable to learn by themselves and from others. Knowledge sharing from manual stations is facilitated via the camera system mounted to the manual station (inter-station learning). Furthermore, these stations follow an (exploratory) trial-and-error approach to gain knowledge or identify new workflows on their own.
3. **Automated stations** are stations that are not capable of learning on their own. In their initial state, the knowledge level for these stations is 0%. They need to be set up for each specific operation and workflow, which is possible with the exceedance of an externally defined threshold level $know^{activ,autom}_{auton,o}$ at the autonomous station. This enables the specific operation at all automated stations. The knowledge level is set to 100%. Accordingly, the modelled knowledge in the digital twin is of binary nature ($know_{s,o} \in 0, 1\ \forall s \in S^{non-learning}, o \in O$).

4.2. Disassembly use case

The presented approach has been tested on an instance of a remanufacturing production system that aims at mastering core (i.e. product that gets disassembled in remanufacturing) disassembly with varying product conditions by utilizing autonomous robots in an agile and scalable matrix production system. The station types hosted by the system equal the setup as defined in the previous section. Additionally, a measuring station complements the setup. This station - without any learning abilities - derives information on the condition of the cores and provides it as input for the production control. Four stations, one of each type, are assumed in the following test case. A discrete event simulation is deployed as a digital twin of the system and used to test the approach in the following.
While inter-station learning and knowledge sharing are introduced in the model as peer-to-peer interactions, the digital twin hosts a knowledge server entity, which stores the level of available knowledge. In a real production system, such an entity can be utilized to model the proceedings in the digital twin and to derive necessary actions, movements and tools in the real-world production system.

4.3. Production control scope

Production control treats the returned products as shop orders and sends them through various stations until they are fully disassembled. Possible next actions are provided to the control system by the digital twin which in turn selects one of them for execution. Each action consists of a (s, o)-tuple for a given shop order in the system.To validate the proposed model, a simple random logic is deployed to show the learning processes in an unbiased way.

4.4. Test cases and parameters

The presented approach has been validated in various simulation runs. A simulation run includes the disassembly of 2,000 orders.

Only one product type is assumed as a simple assembly consisting of four parts (A, B, C, D). This type of product, as introduced in Figure 3, can be disassembled in two different ways. Therefore, testing is split into two different cases. One in which the disassembly precedence knowledge is static and complete (case 1) and another case where the precedence knowledge is incomplete in the beginning and evolves (1x) during run-time (case 2).

The system entities are parameterized using $b_{know} = 0.8$ and $n_{s,o} = 100 \forall s, o$.

Inter-station learning occurs between the manual and autonomous station with $\eta_{1,man,auton} = 0.1$ and $\eta_{2,man,auton} = 25$. The knowledge threshold of an autonomous station is assumed at $know^{activ,auton}_{auton,o} = 0.45$ and the threshold for the automated station at the autonomous station at $know^{activ,autom}_{auton,o} = 1.0$.

4.5. Results

Figure 5 shows the knowledge level per station during the beginning of the simulation in which learning is still occurring for an exemplary operation o_1. As the manual station is enabled from the very beginning, it follows only its own learning curve as it conducts solely intra-station learning. The autonomous station learns by demonstration from the manual station and is initially bound to inter-station learning as it does not reach the enabling threshold of $know^{activ,auton}_{auton,o}$. After about 270 finished orders, the autonomous station is enabled and can start to learn intra-stationary. Due to the initially low value of $x_{auton,o}$, this leads to strong learning progress as a doubling in the cumulative amount of operation trials at the autonomous station occurs more frequently. In a figurative sense, this would mean that the autonomous station can learn faster by trying things out for its own, rather than just by transferring knowledge from other stations, which seems intuitively logical. Once the autonomous station reaches a knowledge level of 100%, the automated station gets enabled which is indicated by a change in its knowledge level from $know_{autom,o} = 0$ to $know_{autom,o} = 1$.

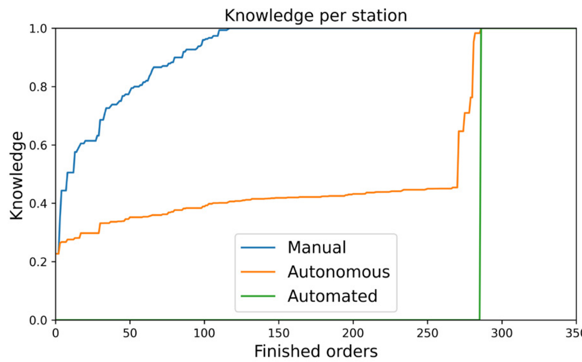

Fig. 5: Knowledge development for an exemplary operation o_1 station by station (case 1)

Figure 6 displays the average knowledge across all offered operations per station. Thereby the knowledge development of

case 1 and case 2 are compared. First of all, one can observe in both cases that enabling of the five operations is conducted at different points in time, as every combination of operation o and station s follows its own learning curve. A step-shaped curve can be spotted for the knowledge development of the autonomous station and, especially, for the automated station. For the latter, the enabling steps of the automated station indicate the moment in which the autonomous station reaches 100% knowledge for another operation o.
By introducing new operation sequences at a later point in simulation time, learning might stall until new sequences are introduced. However, this does not affect the operations which are available to the system as they can reach 100% individually. In the evolving case, it can be observed that the second sequence was introduced after approximately 370 finished orders.

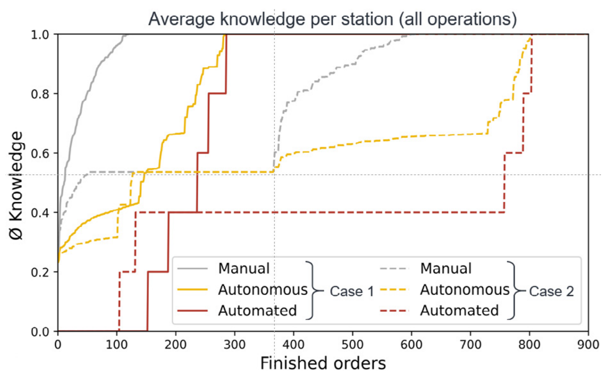

Fig. 6: Development of the average knowledge over all operations o station-by-station with perfect (case 1) and with evolving (case 2) precedence relation knowledge

5. Conclusion and outlook

In the paper at hand, learning effects and knowledge sharing are incorporated into a shopfloor model for production systems with learning abilities. Existing learning curve models for human operators are adapted and extended to fit technical entities that operate autonomously and can exchange knowledge with human operators in hybrid factories. The generic model can be used as a guideline and be deployed for various use cases, e.g. to predict system behaviors using discrete-event simulation models to pre-test planning and control approaches virtually before they go live.
However, before the modeling approach can be exploited, assumptions should be validated. This includes empirical testing and adjustment of formulae and parameters for learning effects and knowledge sharing using real data. Thereafter, the approach can be utilized as a helpful extension for simulation models to develop suitable planning and control systems for cognitive factories.

Acknowledgements

The project AgiProbot is funded by the Carl Zeiss Foundation.

References

[1] Y. Koren, The global manufacturing revolution: Product-process-business integration and reconfigurable systems, 2010.
[2] J. Kurilova-Palisaitiene, E. Sundin, Challenges and opportunities of lean remanufacturing, International Journal of Automation Technology 8 (5) (2014) 644–652.
[3] H. Poschmann, H. Brüggemann, D. Goldmann, Disassembly 4.0: A review on using robotics in disassembly tasks as a way of automation, Chemie Ingenieur Technik 92 (4) (2020) 341–359.
[4] H. Bley, G. Reinhart, G. Seliger, M. Bernardi, T. Korne, Appropriate human involvement in assembly and disassembly 53 (2) (2004) 487–509.
[5] A. Bannat, T. Bautze, M. Beetz, J. Blume, K. Diepold, C. Ertelt, F. Geiger, T. Gmeiner, T. Gyger, A. Knoll, C. Lau, C. Lenz, M. Ostgathe, G. Reinhart, W. Roesel, T. Ruehr, A. Schuboe, K. Shea, I. Stork genannt Wersborg, S. Stork, W. Tekouo, F. Wallhoff, M. Wiesbeck, M. F. Zaeh, Artificial cognition in production systems, IEEE Transactions on Automation Science and Engineering 8 (1) (2011) 148–174.
[6] Y. Li, T. O. Boucher, Assembly line balancing problem with task learning and dynamic task reassignment, The International Journal of Advanced Manufacturing Technology 88 (9-12) (2017) 3089–3097.
[7] C. H. Glock, E. H. Grosse, M. Y. Jaber, T. L. Smunt, Applications of learning curves in production and operations management: A systematic literature review, Computers & Industrial Engineering 131 (2019) 422–441.
[8] A. Kuhnle, Simrlfab: Simulation and reinforcement learning framework for production planning and control of complex job shop manufacturing systems (2020).
URL https://github.com/AndreasKuhnle/SimRLFab
[9] T. P. Wright, Factors affecting the cost of airplanes, Journal of the Aeronautical Sciences 3 (4) (1936) 122–128.
[10] L. Argote, D. Epple, Learning curves in manufacturing, Science (New York, N.Y.) 247 (4945) (1990) 920–924.
[11] E. H. Grosse, C. H. Glock, S. Müller, Production economics and the learning curve: A meta-analysis, International Journal of Production Economics 170 (2015) 401–412.
[12] I. Tirkel, Yield learning curve models in semiconductor manufacturing, IEEE Transactions on Semiconductor Manufacturing 26 (4) (2013) 564–571.
[13] M. F. Zaeh, M. Beetz, K. Shea, G. Reinhart, K. Bender, C. Lau, M. Ostgathe, W. Vogl, M. Wiesbeck, M. Engelhard, C. Ertelt, T. Rühr, M. Friedrich, S. Herle, The cognitive factory, in: H. A. ElMaraghy (Ed.), Changeable and Reconfigurable Manufacturing Systems, Springer Series in Advanced Manufacturing, Springer London, London, 2009, pp. 355–371.
[14] M. F. Zaeh, M. Ostgathe, F. Geiger, G. Reinhart, Adaptive job control in the cognitive factory, in: H. A. ElMaraghy (Ed.), Enabling Manufacturing Competitiveness and Economic Sustainability, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 10–17.
[15] S. Vongbunyong, S. Kara, M. Pagnucco, A framework for using cognitive robotics in disassembly automation, in: D. A. Dornfeld, B. S. Linke (Eds.), Leveraging technology for a sustainable world, Springer, Berlin, 2012, pp. 173–178.
[16] S. Vongbunyong, S. Kara, M. Pagnucco, Application of cognitive robotics in disassembly of products, CIRP Annals 62 (1) (2013) 31–34.
[17] S. Vongbunyong, S. Kara, M. Pagnucco, Learning and revision in cognitive robotics disassembly automation, Robotics and Computer-Integrated Manufacturing 34 (2015) 79–94.
[18] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Science+Business Media, Berlin, Heidelberg, 2008, pp. 1371–1394.
[19] Y. Tang, M. Zhou, E. Zussman, R. Caudill, Disassembly modeling, planning and application: a review, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), IEEE, 2000, pp. 2197–2202.