Integrated Virtual Preparation and Commissioning: supporting formal methods during automation systems development

M. Dahl * **K. Bengtsson** * **P. Bergagård** * **M. Fabian** * **P. Falkman** *

* *Chalmers University of Technology, Department of Signals and Systems, 412 96 Göteborg, Sweden.*
email: {martin.dahl, kristofer.bengtsson, patrik.bergagard, fabian, petter.falkman} @ chalmers.se

Abstract: Virtual commissioning – the development and validation of industrial control systems against a simulation model – is attracting interest in the automotive industry. The main motivation for its use is that control systems can start to be integrated and tested before the construction of the physical system. In addition to this, the ability to continuously test can lead to increased reliability and enables better coping with late changes. At the same time, using formal methods during production preparation and control system design promise similar benefits. Formal methods however, are not seeing the same surge in interest – they are rarely used in the automotive industry. In this paper a framework is proposed, *Integrated Virtual Preparation and Commissioning*, where virtual commissioning models are used as a base for preparation and control system implementation assisted by formal methods. The extensive use of simulation in virtual commissioning allows computation results from formal methods to be continuously validated by visual inspection and using existing analysis tools (e.g. collision detection methods). The framework is applied in a case study, where the combination of a simulation model and a formal model is used as an aid in generating operation sequences for validation during production preparation. The resulting formal model can be used to study the behavior of the production system before a control system has been implemented.

Keywords: virtual commissioning, production preparation, simulation, digital manufacturing

1. INTRODUCTION

Today's automotive industry is under constant pressure to deliver high quality products at lower cost and shorter time to market. Not only is having robust, efficient, and flexible production systems needed, but the development time of such systems must also be considered.

Using formal methods relating to production have been widely studied in academia (Campos et al., 2014). Being a very broad research area, potential benefits include: better reliability and reduced development time through verification (Ovatman et al., 2014), improved scheduling (Solnon et al., 2008), and increased ability to handle complexity by synthesis (Miremadi et al., 2012). While formal methods related to production can positively impact both the development time of a production system as well as efficiency and reliability after production start, they are not used to any significant extent in the automotive industry (Ljungkrantz et al., 2010). The main reasons for not using formal methods seem to be lack of tools and methodologies, coupled with hard to see benefits – to gain the most, new tools and workflows need to be created and traditional mindsets need to be changed.

Another area of study related to reducing development time of production systems is virtual commissioning (VC). In VC, a simulation model of the production system is used for validation during development of an automation system (Lee and Park, 2014). The idea is that the simulation model of the production system should have the same I/O setup as the intended physical system and contain logic simulating device behavior. This gives the ability to start control system integration and testing before finishing (or even starting) construction of the physical system. Additionally, flexibility can be improved as changes can be tested offline during production. While VC comes with its own set of problems, mostly related to the modeling effort required (Lee and Park, 2014), it is however, gaining interest and with that, software support is emerging.

The models used for VC contain both physical and logical behavior. Formal methods can gain by reusing information contained in these VC models, and by using formal methods, it is possible to generate parts of the VC models. Moreover, because VC relies extensively on *simulation* as a means to validate control systems, functions for performing validation of the results of the formal methods already

⋆ This work has been carried out at the Wingquist Laboratory VINN Excellence Centre within the Production Area of Advance at Chalmers. It has been supported by VIRTCOM-Virtual preparation and commissioning of production systems including PLC logics, reference number 2014-01408, Vinnova, FFI within Sustainable production technology.

exist in digital manufacturing tools used for VC today. By extending VC to include preparation and control system design, these VC models can, in addition to providing hardware-in-the-loop testing, also be used as a validation tool for preparation and control system design assisted with formal methods. This means that the VC models will be constructed gradually during the development of the production system.

1.1 Contribution

In this article, *Integrated Virtual Preparation and Commissioning* (*IVPC*) is introduced. IVPC is a framework for model-based preparation and control system implementation that supports hardware-in-the-loop testing. The main idea in IVPC is to be able to continuously validate intermediate work, which should be the result of using formal methods, during production preparation and later on control system implementation. Validation is performed using virtual models of the production system that are shared between the activities highlighted in Figure 1. This will have the following benefits:

- It becomes easier to change details relating to preparation late in development.
- Information reuse: information needed for formal methods is already present in the VC models.
- Parts of the VC models can be generated from the formal models.
- Hardware in the loop tests can be performed at any time during control system implementation.
- Using a familiar simulation environment as the base for using formal methods can make them more attractive.

Through a case study, one possible application of IVPC is demonstrated by applying these ideas to part of a body-in-white production system from a large automotive manufacturer. As part of this case study, a software product that bridges the simulation model built with the purpose of VC and a formal model for working with operation sequencing is developed, creating a user friendly way of working with operation sequencing.

1.2 Outline

This paper is organized as follows: in Section 2, a brief background is given. In Section 3, the IVPC framework is introduced and a few of the main ideas in it are then highlighted using a case study in Section 4. The paper ends with some concluding remarks in Section 5.

2. BACKGROUND

One possible decomposition of the development of a new body-in-white production system is outlined in Figure 1. Let us briefly review these activities for clarity. In *production planning*, the broad strokes of the production system are decided upon. This may include physical layout, resources (e.g. robots, tools) and the order of parts assembly. In *production preparation*, these details are refined – simulations and robot programs are created to make sure that cycle times are kept within acceptable bounds. In the *control system implementation* phase, one or more cycle(s)

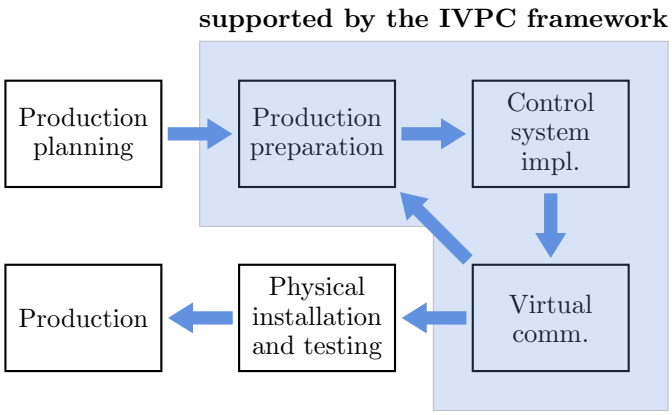

Fig. 1. The steps in the development of a new production system, activities supported by IVPC are highlighted.

is converted to logic contained in an automation control system. In the control system aspects such as interlocking and safety needs to be considered. During *Virtual Commissioning* the control system can be finalized by allowing testing and integration using a simulated plant. In case of undesired behavior details (e.g. control logic or robot programs) may need to change in earlier phases. The following *Physical installation* consist of installation of devices, electrical wiring and testing (i.e. dry runs). Finally, the system is ready for production start.

2.1 Formal methods in automation systems

Despite being a well researched area, formal methods have yet to gain any significant acceptance in the automotive industry. But being able to guarantee correct operation of production systems with steadily increasing complexity means that there is a lot to gain if they can be introduced into the daily engineering workflow. There are two main approaches in the context of guaranteeing correct operation of automation systems – verification, typically based on model checking (see Ovatman et al. (2014) for an overview), and synthesis of logic (see Miremadi et al. (2012) for an example).

Production systems featuring a lot of sequential behavior have traditionally been sequenced using Gantt charts (Wilson, 2003). In modern systems, where the demand for flexibility and the ability to optimize is greater, this way of sequencing can be too rigid, as the time of execution of all operations are set “in stone”. It also creates a mismatch in domain, as the implementation of the control system needs to be based on logic and the Gantt chart is based on time. This means that there is very little reuse possible and makes the Gantt chart more of a validation tool for the control system engineers.

By using synthesis of logic, a potentially better solution would be to specify as little as possible in the planning phase to make sure that no good solutions are discarded early. One way to achieve this is to only specify what conditions need to be fulfilled before each operation is allowed to start (Bengtsson, 2012). With conditions being based on logic rather than time, the formal model can then be used as the basis for control logic when designing a control system.

2.2 Virtual Commissioning

As described by Auinger et al. (1999), there exist four basic configurations in which development, testing and integration of an automation system can occur. Traditionally the physical production system has been tested and integrated using the real control system. Starting with a control system that is prepared to some degree, the system is integrated into the production system and tested. This activity may take weeks. But the real control system can also be coupled with a simulation model of the production system, in what is usually called a hardware-in-the-loop (HIL) setup. This setup is what VC commonly refers to (Lee and Park, 2014) and it is also how the term is used in this work. The inverse, reality-in-the-loop commissioning, would be the physical production system controlled by a simulated controller. This can have benefits for example when debugging a control system. The last combination of the two is the simulation model together with the simulated control system – this is called offline programming or constructive commissioning (Lee and Park, 2014). When designing a new control system, this is the natural place to start.

The main motivation to use VC as defined in the previous paragraph is to reduce testing and integration time during development. This is achieved by being able to test and integrate the control system before the physical production system is completely installed. The hope is that using a simulation model of the production system, undesirable behavior can be detected well ahead of physical installation. In fact, conducting a VC enables tests that would be prohibitively expensive or even impossible to run on a physical system. In addition to this, having the simulation model makes it possible to test changes to the production system while it is running and being able to incorporate last minute changes without worrying about their impact on the system. (Hoffmann et al., 2010; Park and Chang, 2012; Lee and Park, 2014)

Because VC, as defined in this work, uses the real control system, the simulation models needs to be specified at the level of sensors and actuators (Lee and Park, 2014). This is now possible in simulation software from vendors in digital manufacturing tools, usually by allowing the user to define signals connected to the simulation and exposing them to a control system via an interface (e.g. OPC, see Schwarz and Borcsok (2013)). Even though software and vendor support for VC is growing, there are a number of issues that hinder its broader adoption.

For companies that have not yet transitioned to extensive use of simulation, VC requires expertise that may not be readily available in house. Oppelt and Urbas (2014) cites lack of simulation knowhow as a major hurdle in the introduction of VC. This especially hinders adoption in smaller companies, which may also lack access to the required software tools. In addition, creating the simulation models requires an extensive modeling effort. While research is being conducted focusing on generating both the physical/kinematic parts of the models (e.g. Chang et al. (2011); Barth and Fay (2013)) and the logical parts (e.g. Park et al. (2010, 2013)), this is not general enough to be applied in all cases. A lot of modeling still need to be done manually.

There are issues related to timing during simulation, affecting the fidelity of the simulations (and thus the quality of the validation provided). This is well described by Carlsson et al. (2012), who also propose a solution to this problem.

In industries where many custom devices are used, the modeling effort to correctly describe the logic of these devices can be enormous. Creating a virtual representation of a complex device can be almost equal in effort to creating the software running on the device. While there exist a vision in industry that all devices should be delivered with a corresponding virtual model, it does not appear that this will be the case any time soon.

2.3 Integrated Virtual Commissioning

Oppelt and Urbas (2014) write that according to the The Association of German Engineers, current guidelines state that VC should be the last step in the automation engineering phase (which roughly corresponds to control system implementation in this article), basically to reduce the integration time of the control system. Instead of conducting a VC only as the last step, Oppelt and Urbas (2014) suggest extending VC to cover the entire automation engineering phase. They call this concept *Integrated Virtual Commissioning*. Their argument is that VC can provide continuous value during the automation engineering phase by enabling continuous testing during the development – "The virtual plant is growing together with the automation software and thus enables simulation supported automation engineering." (Oppelt and Urbas, 2014).

A conclusion on a similar theme was reached already by Drath et al. (2008), "Once the virtual commissioning is embedded into the engineering workflow, the corresponding virtual commissioning models will not only be created in the described test phase but already in the offer and process engineering phase". This early creation of the VC models is also one of the ideas in the *Integrated Virtual Preparation and Commissioning* framework.

3. INTEGRATED VIRTUAL PREPARATION AND COMMISSIONING

In essence, the building blocks for more efficient development of production systems are already there. While the theory around formal methods have been ready for a long time, there has not been a lot of work done on software and methodologies that suit the daily engineering work.

Another observation is that VC is actually happening, albeit slowly. This may be due to the fact that, even though it requires a lot of engineering hours, the purpose of it is straightforward and it does not differ much from the current workflow. This is especially true when VC modeling is added as extensions to existing simulation software.

Due to increased modeling costs when using VC, and costs associated with its introduction, it is vital that the models created provide as much additional value as possible. One idea on how to achieve this is to not only use these models for VC, but also *as a base for preparation and control system implementation assisted by formal methods.*

Hence, we introduce the framework *Integrated Virtual Preparation and Commissioning*, *IVPC*. The idea in IVPC is to have the same virtual models of the production system shared between the preparation, control system implementation and VC phases (the highlighted area in Figure 1) and *use these models for validation* throughout development.

Of course, in the early stages, neither I/O:s nor robot programs have been defined. This does not prevent development using IVPC. In fact, there may be benefits to organically grow the VC model during the process. Some parts of the VC model can be generated (and by extension, re-generated when changes occur) and should not be created manually. One example of this is robot programs that simply move products between value adding operations, which can be generated using software that solves the path generation (e.g. Fraunhofer IPS [1] or Siemens Kineo [2]).

Let us review some potential benefits of using VC models as the basis for this framework:

- The same models can be used during both preparation and VC. This minimizes the amount of manual conversion work needed before implementing and then performing VC. But more importantly, it enables going back and changing details late in development.
- Information needed for the formal methods is already present in the VC models, or can be generated based on them. Ideally, a user should only have to enter some key details when working with the formal methods.
- Hardware in the loop tests can be performed at any point during control system implementation, for any components that have been modeled at the signal level. This will enable continuous testing of the control system, as described by Oppelt and Urbas (2014).
- Even when work on the VC models is ongoing (i.e. during preparation), work can be assisted by formal methods. An example of this is provided in Section 4.
- Having a familiar simulation environment (that hosts the VC models) as the base for using formal methods should increase their attractiveness – especially if the computation results can be analyzed in said simulation environment.

4. IVPC CASE STUDY - OPERATION SEQUENCING

To better illustrate what *Integrated Virtual Preparation and Commissioning* can mean in practice, let us study an example of using a VC model together with formal methods to help an engineer more efficiently work with sequencing of operations. The aim is to create a model containing the logic relevant to operation ordering which will help the engineer to simulate all possible combinations of sequences that are valid. This enables early validation of logic behavior.

The scope of this case study is limited to the preparation phase. This means that no information about I/O:s that is present in the VC model is used and no control system is implemented. Instead the VC model is used as a simulation and robot programming tool and the operations are started by generating sequences that can be executed within the simulation software. In later stages, operations will instead start by setting signals using the VC capabilities in the digital manufacturing tool.

4.1 The studied cell

The production system studied is a robot cell for welding support beams which is part of a production line at Volvo Car Corporation.

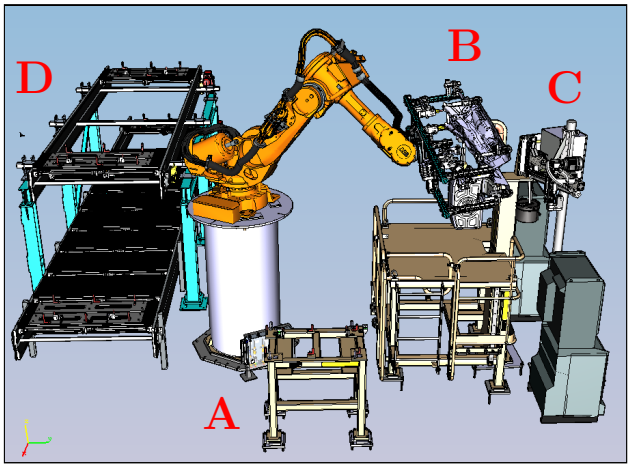

Fig. 2. VC model of the robot cell used as an example in the case study.

The cell, which can be seen in Figure 2, consist of an industrial robot with a gripper, a weld station, a tip-dressing station and two conveyors for transporting the part away after welding. The cell takes two types of input parts (let's call them Product A and Product B), each with different weld programs. Parts are placed by an operator on the table by **A**, welded at **B** and then placed on the top or bottom conveyor at **D** in Figure 2. The tip-dressing device is located at **C** in Figure 2, and is designed to fold in and tip-dress in front of the stationary weld after every 4:th weld.

4.2 Sequence Planner

The model containing the logical behavior of the operations in the cell is hosted in *Sequence Planner* (SP) (Bengtsson, 2012), a software developed at Chalmers University of Technology. As its name suggest, it is specifically designed for working with sequences of operations.

In SP, operation sequences and their relations are visualized using the Sequences of Operations (SOP) language, the definition of which can be found in Lennartson et al. (2010).

During preparation, when the VC model is changing often (e.g. when new devices are added, or robot programs are added or updated), it needs to be straightforward to keep the formal model up to date. A two-way communication channel between SP and the software hosting the VC model is set up to allow for seamless integration between the two.

[1] Fraunhofer IPS, http://www.fcc.chalmers.se/software/ips/, 2015
[2] Siemens Kineo, http://www.plm.automation.siemens.com/, 2015

4.3 Formal operation model

The modeling language used to define the operation sequencing is inspired by the work in (Bergagård et al., 2015). The production system is modeled using operations and variables that describe positions of devices and parts. Operations have conditions that determine when they are allowed to execute. These conditions include the variables, making the operations allowed to start depending on the global state of the system.

Conditions on the operations to fulfill global specifications are preferably synthesized. Calculation of this is based on Supervisory Control Theory (SCT) (Ramadge and Wonham, 1987). SCT is a model-based framework for control of discrete event systems based on automata. One of the key ideas in SCT is that modeling should be kept simple by focusing on local behavior. Then specifications are added to forbid (global) behavior that is not desired. A control function is synthesized from the model and the specifications, guaranteeing that the specifications will always be fulfilled.

4.4 Creating the formal model

Information from the software hosting the VC model can be extracted using the two-way communication channel. If operations for gripping and welding have already been implemented (as robot programs) in the VC model, the conditions relating to device positions as well as where the parts are located can be generated from it. Based on this, transport operations (in this case moving the robot between locations) can be generated (see Magnusson et al. (2011), where a formal model for transport operations is generated from a simulation model).

As work progresses, the model becomes more detailed. For example, the weld tool needs to have tip dressing applied after every 4:th weld to satisfy quality requirements, which means that an operation for tip dressing needs to be added. This leads to potential interlocking conflicts as the welding operations and the tip dressing operation share the same physical space. This can be handled with the addition of a specification that states that the tip dresser cannot be active when any of the two product types is being welded. But to leverage the VC model, specifications such as this should instead be generated by identifying potential collisions using simulation and updated automatically whenever robot programs or geometries change (an example of this can be found in Shoaei et al. (2010)). This means fewer things to (manually) keep in sync when the VC model changes.

An example of a specification that cannot as easily be generated can be seen in Figure 3. In this safety specification, it is explicitly stated that whenever an operator is present in the cell, the operation gripProductA is not allowed to execute. Note that the entering and leaving of an operator are modeled by operations, to easily be able to study scenarios where an operator is involved (see Section 4.5). The implication of this specification depends on the other operations. If gripProductA is the only operation that satisfies the conditions needed to move on to welding, the execution would stop whenever needing to grip Product A and the operator is present. But there could exist another

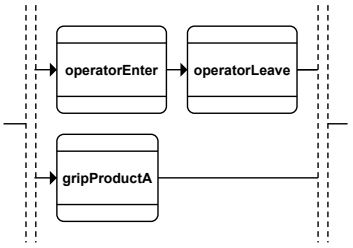

Fig. 3. An example of a manually created specification in the SOP language. The parallel branches are mutually exclusive.

version of gripProductA (e.g. gripProductA_safe, with a different robot program) that is not forbidden to run by the specification in Figure 3.

4.5 Studying different scenarios

Talking to simulation engineers have revealed a need to easily validate different scenarios related to sequencing using the VC model. In industry, ad hoc solutions such as having a robot program read sequences from a file have been observed. After synthesizing the control function the formal model contains all possible *valid* sequences encoded in the conditions on the operations. Even though any possible sequence can be generated, studying all possible outcomes may not be feasible or desirable. By allowing a user to enter a rough outline of what is to be inspected, it is possible to generate sequences (given that such a sequence is allowed to execute by the formal model) that contain some key elements.

For example, to validate the resulting formal model after adding the specification in Figure 3, an engineer wants to study a straight sequence containing two instances of Product A, with the operator being present during the second instance. As expected, when the operator is present (indicated by **1** in Figure 4) the safe version of the grip operation will be executed (indicated by **2** in Figure 4). Using the two-way communication channel, this sequence can be executed in the software hosting the VC model, allowing for validation through visual inspection and built-in collision detection methods.

These types of tests would traditionally have been conducted during virtual commissioning, as a test of the control system, but the use of a formal model for the logic relating to when operations can execute makes it possible to conduct them already in the preparation phase.

5. CONCLUSION AND FUTURE WORK

This paper is the first step in realizing a framework for applying formal methods to the different phases of designing part of a new production system. The case study in Section 4 aims to demonstrate that with a formal model integrated in the workflow, validation of sequencing logic can start earlier compared to when performing a traditional virtual commissioning. The next step is driving the simulation using the VC capabilities in the simulation software, which means that a control system needs to be created from the formal model of the operations.

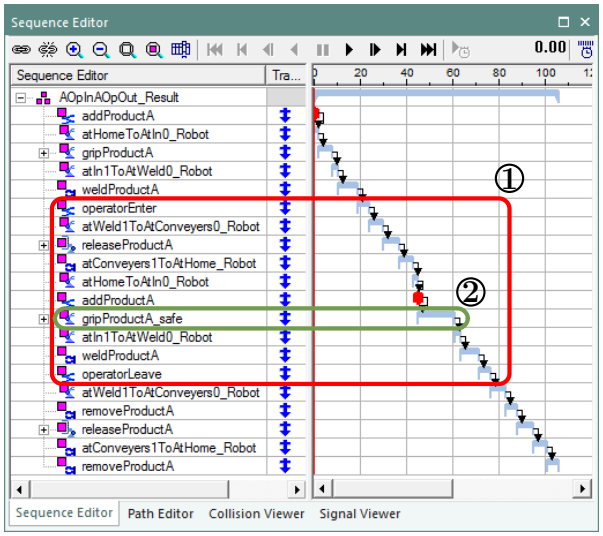

Fig. 4. A sequence sent to the simulation software for validation. The effect of a specification that ensures a safe grip operation (**2**) when an operator is present (**1**) is highlighted.

Note that no studies have been conducted regarding whether the method developed in the case study will result in any significant time savings during development.

ACKNOWLEDGEMENTS

The authors would like to thank the industrial partners of the VIRTCOM project for the discussions during meetings, and Volvo Car Corporation in particular for sharing the VC model used in this work.

REFERENCES

Auinger, F., Vorderwinkler, M., and Buchtela, G. (1999). Interface driven domain-independent modeling architecture for "soft-commissioning" and "reality in the loop". In *Proc. 31st Conf. Winter Simul. Simulation—a Bridg. to Futur. 1*, 798–805. ACM.

Barth, M. and Fay, A. (2013). Automated generation of simulation models for control code tests. *Control Eng. Pract.*, 21(2), 218–230.

Bengtsson, K. (2012). *Flexible design of operation behavior using modeling and visualization*. Doktorsavhandlingar vid Chalmers tekniska högskola. Ny serie, no:. Institutionen för signaler och system, Automation, Chalmers tekniska högskola,.

Bergagård, P., Falkman, P., and Fabian, M. (2015). Modeling and automatic calculation of restart states for an industrial windscreen mounting station. *IFAC-PapersOnLine*, 48(3), 1030–1036.

Campos, J., Seatzu, C., and Xie, X. (2014). *Formal methods in manufacturing*. CRC Press.

Carlsson, H., Svensson, B., Danielsson, F., and Lennartson, B. (2012). Methods for reliable simulation-based PLC code verification. *IEEE Trans. Ind. Informatics*, 8(2), 267–278.

Chang, M., Ko, M., and Park, S.C. (2011). Fixture modelling for an automotive assembly line. *Int. J. Prod. Res.*, 49(15), 4593–4604.

Drath, R., Weber, P., and Mauser, N. (2008). An evolutionary approach for the industrial introduction of virtual commissioning. In *Emerg. Technol. Fact. Autom. 2008. ETFA 2008. IEEE Int. Conf.*, 5–8. IEEE.

Hoffmann, P., Maksoud, T., Schumann, R., and Premier, G. (2010). Virtual Commissioning of Manufacturing Systems a Review and New Approaches for Simplification. *Proc. 24th Eur. Conf. Model. Simul.*, 2(Cd).

Lee, C.G. and Park, S.C. (2014). Survey on the virtual commissioning of manufacturing systems. *J. Comput. Des. Eng.*, 1(3), 213–222.

Lennartson, B., Bengtsson, K., Yuan, C., Andersson, K., Fabian, M., Falkman, P., and Åkesson, K. (2010). Sequence planning for integrated product, process and automation design. *Autom. Sci. Eng. IEEE Trans.*, 7(4), 791–802.

Ljungkrantz, O., Åkesson, K., and Fabian, M. (2010). *Practice of industrial control logic programming using library components*. INTECH Open Access Publisher.

Magnusson, P., Sundström, N., Bengtsson, K., Lennartson, B., Falkman, P., and Fabian, M. (2011). Planning transport sequences for flexible manufacturing systems. In *Proc. 18th IFAC World Congr. 2011, Milano, 28 August-2 Sept. 2011*.

Miremadi, S., Lennartson, B., and Åkesson, K. (2012). A BDD-based approach for modeling plant and supervisor by extended finite automata. *Control Syst. Technol. IEEE Trans.*, 20(6), 1421–1435.

Oppelt, M. and Urbas, L. (2014). Integrated Virtual Commissioning an essential Activity in the Automation Engineering Process From virtual commissioning to simulation supported engineering. In *Ind. Electron. Soc. IECON 2014 - 40th Annu. Conf. IEEE*, 2564 – 2570.

Ovatman, T., Aral, A., Polat, D., and Ünver, A.O. (2014). An overview of model checking practices on verification of PLC software. *Softw. Syst. Model.*, 1–24.

Park, H.T., Kwak, J.G., Wang, G.N., and Park, S.C. (2010). Plant model generation for PLC simulation. *Int. J. Prod. Res.*, 48(5), 1517–1529.

Park, S.C. and Chang, M. (2012). Hardware-in-the-loop simulation for a production system. *Int. J. Prod. Res.*, 50(8), 2321–2330.

Park, S.C., Ko, M., and Chang, M. (2013). A reverse engineering approach to generate a virtual plant model for PLC simulation. *Int. J. Adv. Manuf. Technol.*, 69(9-12), 2459–2469.

Ramadge, P.J. and Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1), 206–230.

Schwarz, M.H. and Borcsok, J. (2013). A survey on OPC and OPC-UA: About the standard, developments and investigations. In *Information, Commun. Autom. Technol. (ICAT), 2013 XXIV Int. Symp.*, 1–6. IEEE.

Shoaei, M.R., Lennartson, B., and Miremadi, S. (2010). Automatic generation of controllers for collision-free flexible manufacturing systems. In *Autom. Sci. Eng. (CASE), 2010 IEEE Conf.*, 368–373. IEEE.

Solnon, C., Cung, V.D., Nguyen, A., and Artigues, C. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *Eur. J. Oper. Res.*, 191(3), 912–927.

Wilson, J.M. (2003). Gantt charts: A centenary appreciation. *Eur. J. Oper. Res.*, 149(2), 430–437.