MANUFACTURING SIMULATION WITHIN A RAPID CONTROL PROTOTYPING APPROACH – A COMPARATIVE STUDY

Tobias Pingel, Sven Pawletta, Thorsten Pawletta, Gunnar Maletzki

Hochschule Wismar, University of Applied Sciences: Technology Business and Design,
Faculty of Engineering, Research Group Computational Engineering and Automation,
23952 Wismar, Philipp-Mueller-Strasse, PF 1210, Germany.

tobias.pingel@hs-wismar.de (Tobias Pingel)

Abstract

The development and realization of complex and automated flexible manufacturing systems is a multi-step and cost-intensive process. Concepts and solutions which provide a systematic development process from the early design phase until the operational control are referred to as Rapid Control Prototyping (RCP) approaches. Where RCP is common practice in developing feedback control systems this approach in context with complex discrete controls of manufacturing systems is still a matter of research.
This paper introduces a project dealing with RCP technology for manufacturing systems and predictive simulations for optimization in the operation phase. The main focus of this contribution is put on the first phase of the control development process. In a detailed comparative study two DES tools are used to model and analyze a coating plant as an example of a flexible manufacturing system. The results of the study are presented and discussed in-depth by addressing modeling effort, simulation accuracy and runtime. After summarizing the preceding sections an outlook on future steps of the complete research project is given.

Keywords: Manufacturing Simulation, Rapid Control Prototyping, Simulation Model Based Control.

Presenting Author's Biography

Tobias Pingel is a Research Associate at the Faculty of Engineering, Hochschule Wismar, University of Applied Sciences: Technology, Business and Design, Wismar, Germany. He obtained the degree (Honours) in Multimedia Engineering in 2007 and M.Eng. in Process Automation in 2010. He joined the Research Group Computational Engineering and Automation in 2006 and then started with PhD studies at the Wismar University in cooperation with the University of Rostock in 2008. His research interests are predictive simulation and optimization of control strategies as well as distributed and parallel computing techniques. His PhD study focuses on predictive simulation of complex controls in the context of RCP.

1 Introduction

The development and realization of complex and automated flexible manufacturing systems is a multi-step and cost-intensive process. Therefore, a systematic approach from the early design phase until the operational control is described by the so called V-Model as illustrated in Figure 1.

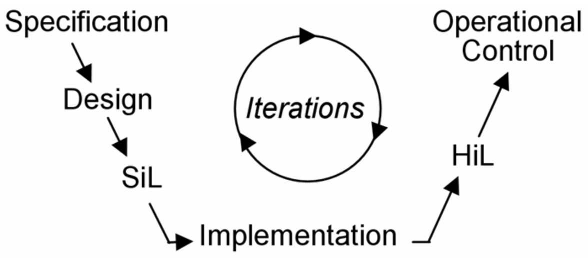

Fig. 1 V-Model for automation solutions following [1].

In this approach model- and simulation-based techniques play a central role nowadays. Hence, for requirements analysis and resource planning process models are already being developed in the early specification phase. While proceeding with subsequent design phases the level of detail of the used models is increased step-by-step. Demanding a strict separation of process and control algorithms during modeling test and analysis within a system simulation – followed by software-in-the-loop (SiL) simulations – can be performed at an early stage. After implementing the control algorithms on the target platform they can be tested within hardware-in-the-loop (HiL) simulations beginning with parts of the system until the operational control of the whole system successively. As a general rule, within the whole development process many backward iterations are necessary.

If it is possible to design the transitions and iterations between each of the phases of the V-Model more efficiently time effort and so the resulting development costs can be reduced significantly. Concepts and solutions which address these needs are referred to as Rapid Control Prototyping (RCP) approaches ([1]). The key idea of RCP is the usage of a coordinated tool chain including automatic code generation for the target platform.

Where RCP is common practice in developing feedback control systems this approach in context with complex discrete controls of manufacturing systems is still a matter of research. Therefor, in the past years various research activities has been initiated for providing a systematic model-based control design. One of these approaches is the Simulation Model Based Control Approach as presented in [2]. The idea of this approach is the successive extension of simulation models from the design phase as control programs for the operating phase within a coordinated software environment. The concept of simulation model based design of discrete controls is introduced in section 2.

Following this approach, the design of control programs is not a separate development step after manufacturing simulation for topology planning and resource requirement analysis. Moreover, corresponding to the V-Model the manufacturing simulation can be seen as the first phase of the control design. With classic manufacturing simulation tools like GPSS such procedure was impossible so far. Neither planning models could be imported into tools for control design nor tools for manufacturing simulation provided a detailed design of discrete controls.

For performing discrete simulations within the Matlab environment in the context of RCP Stateflow was the only commercially available tool before 2006. Its applicability has been proven in [3] and also compared with Flexsim and Dymola in [4]. However, the supported modeling paradigm – state charts – is not the common technique used for material flow analysis. Classical simulation tools for this domain are typically transaction- or entity-oriented. In the meanwhile, Matlab provides a toolbox for entity-based discrete event simulation called SimEvents. In section 3 both tools are presented and it is shown how manufacturing simulation can be performed principally. For comparison reasons both tools are used to model and analyze a flexible manufacturing system as the benchmark problem. The results are discussed in-depth with focus on modeling effort, simulation accuracy and runtime.

Afterwards in section 4, an outlook on future research dealing with applicability of SimEvents in the context of RCP is given. Additionally, the conjunction of SimEvents with Stateflow as the modeling tool base for the subsequent phases of the V-Model including predictive simulations within the operational phase for optimization purposes will be addressed.

2 Simulation Model Based Design of Discrete Controls

The development of discrete controls using a model-based design approach is still a matter of research. In this section first an overview of the state-of-the-art as well as a future trend on design of discrete controls is given. Afterwards the Simulation Model-Based Control Approach following [5] is presented.

2.1 Design of Discrete Controls – State of the Art

Following [1] the process of designing discrete controls can be distinguished into heuristic and model-based approaches. The heuristic approach can be reduced to three major steps of development: (i) implementation, (ii) test and (iii) comparison. First, the desired behavior of *the controlled process* is e.g. just described verbally, then transferred into a control model and finally tested using simulation or at the real process immediately. A concluding comparison of designed and desired process behavior is used for evaluation. Due to the lack of separation of control and process a model-based approach is not applicable ([1]). Compared to continuous process descriptions in feedback control systems formally describing discrete control problems is much more complex and need to be designed modular-hierarchical in parts. For this reason, they are still just applicable in small-sized discrete control applications.

To increase the efficiency of control design simulation

tools are used more and more often. This allows the use of model-based approaches for the design process. Though, a fundamental requirement is the strict separation of control and process model. Where the process model reproduces the behavior of *the uncontrolled process* the control model implies the desired control behavior based on the specification – in combination the controlled process is modeled. Based on simulating the controlled process model the desired process behavior can be checked up on the specification. For validation purposes, design tools which are multidisciplinary, intuitive and easy to understand are most preferable ([1]).
Using model-based approaches following [1] three development steps can be discriminated: (i) analysis, (ii) verification and (iii) syntheses. After modeling the uncontrolled process and specification of the desired control the analysis is used for examining the entire system. Then, the verification is used to check if the designed control corresponds to the specification as well as to test on accessibility of desired and undesired regions within discrete or hybrid state space. The concluding syntheses generates a concrete control corresponding to the process and control specification.
The key idea of RCP is to supply a coordinated tool chain in a homogeneous software environment. However, in the area of model-based control design there is just rudimentary software support. Whereas the areas of test and validation by simulation as well as control syntheses using e.g. Simulink/Stateflow are quite well supported, there is still a lack of appropriate software tools for the development steps analyses and verification.

2.2 Simulation Model-Based Control Approach

Fundamental requirements for a discrete control design in terms of the Simulation Model-Based Control Approach following [5] are:

- a strict separation of the uncontrolled process model and the control model as well as
- a consistent component-oriented modeling of the process.

The simulation model during design phase as illustrated in figure 2 is – according to the demands – strictly separated into process model and control model. Furthermore, the single process elements such as buffer and server are modeled separately. Whereas the components of the process model (high-level) describe the states and time behavior of the real process elements, the control model implies the control strategies and is made up modular-hierarchically – depending on its complexity. Here, high-level components pass control-relevant state values to the control model which returns appropriate control values according to the particular control strategy.
The required extensions of the simulation model for SiL or operational control respectively are highlighted in gray. For connecting the real process with the corresponding high-level component an appropriate low-level component is required. Furthermore, variables used in the design phase such as service times are replaced by real sensor values. The sensor signals cause a state update within the corresponding high-level components which causes the control model to react with appropriate control values in return. Thereupon real actuator values are generated by the corresponding low-level components. This model is being executed in a simulation environment, thus, a component for real-time synchronization need to be added to guarantee accurate handling of process signals.

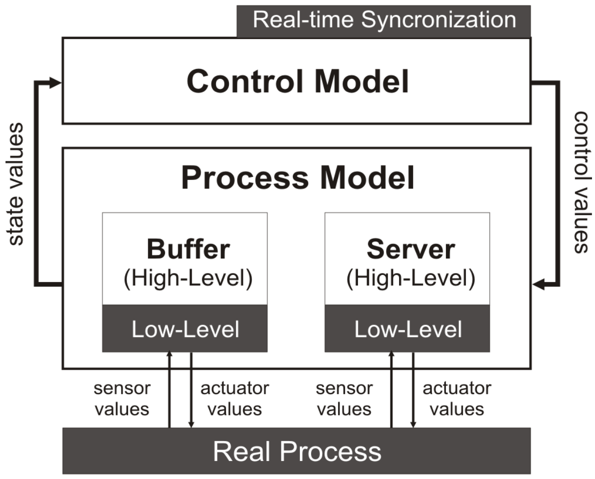

Fig. 2 Simulation Model-Based Control Approach following [5].

Since the state information is stored within the high-level components it can be accessed by the control model during operating phase. Hence, this approach allows the computation of additional state information which is not provided by corresponding sensor signals. Furthermore, a functionality for monitoring is also provided: sensor signals can be compared with computed state values and checked up on errors.
The presented model structure can be used for extensive simulation studies. Different control strategies can be tested using the simulated process and verified with the given specification. By the extension of the simulation model used in design phase to a control program for the operating phase – implying that the design platform equals the target hardware – the subsequent control syntheses can be understood as an *implicit* control syntheses and is comparable to SiL simulations in the context of RCP. The applicability of simulation model-based control design on the base of an implicit control syntheses has been proven in [5].

3 Manufacturing Simulation using Discrete Event Systems (DES)

Descriptions for Discrete Event Systems following [6] can be categorized as (i) event-based, (ii) state-based and (iii) others. In this section, first principle manufacturing simulation using two DES tools of the Matlab product family are presented: Stateflow as a state-based DES tool and SimEvents as an event-based representative. Afterwards both modeling tools are benchmarked using an example flexible manufacturing system. The subsequent discussion focuses on: modeling effort, simulation accuracy and runtime.

3.1 State-based Modeling using Stateflow

Stateflow ([7]) extends Simulink by a graphical tool for modeling and simulation of event-driven reactive systems based on finite state machines. Applicability of using Stateflow for manufacturing simulation has been proven in [3]. Here, blocks such as e.g. buffers, servers and transportation lines need to be self-implemented and connected together for modeling the material flow.

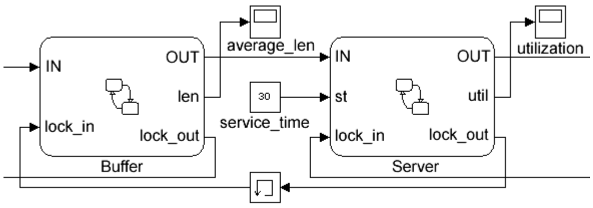

Fig. 3 Section of a material flow system using Stateflow (buffer and server).

Figure 3 illustrates the connection of a buffer and a server component. Each component is implemented as a separate state chart using Stateflow. A buffer can be modeled using three parallel states – for material input, output and statistics – as shown in figure 4. A detailed description is given in [3].

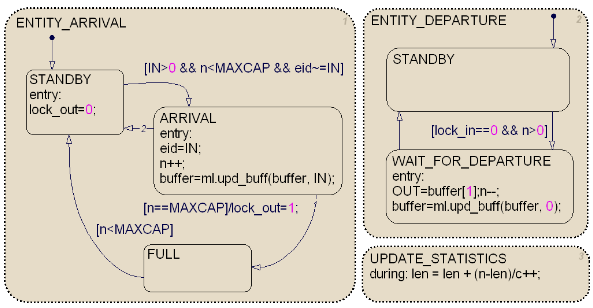

Fig. 4 Stateflow-Chart of a buffer.

For the communication between the material flow components three signals can be discriminated. On the one hand the material flow itself, which is implemented as a signal transporting the unique ID of the corresponding workpiece using `IN`- and `OUT`-ports. And, on the other hand, a backward-directed synchronization signal (`lock_in` / `lock_out`) for realization of blocking workpieces in the case a buffer is full or a server busy. Furthermore, every component is equipped with statistics functionality such as average length of elements for buffers or average utilization for servers. Additionally, the service times for servers can be parametrized using the input `st`. Other parametrization is adjustable within the state chart directly. Following this scheme the entire material flow system can be designed.

3.2 Event-based Modeling using SimEvents

SimEvents ([8]) extends Simulink by a blockset for discrete-event[1] simulation and can be used for planning simulation of manufacturing systems. SimEvents provides blocks for e.g. entity generation, servers, buffers, attribute manipulators and routing components. Here, the material flow is presented by a flow of entities where each entity represents a corresponding workpiece. The blocks just need to be connected via `IN`- and `OUT`-ports. Furthermore, each component is equipped with outputs for statistics as well as parametrization inputs.

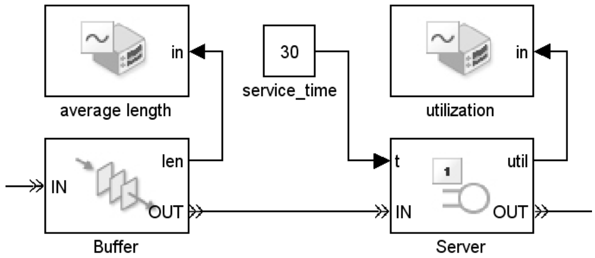

Fig. 5 Section of a material flow system using SimEvents (buffer and server).

Figure 5 pictures the modeling of a buffer connected with a server. The statistic outputs are directly connected to scope blocks for evaluation purposes after the simulation run.

3.3 Benchmarking Stateflow and SimEvents

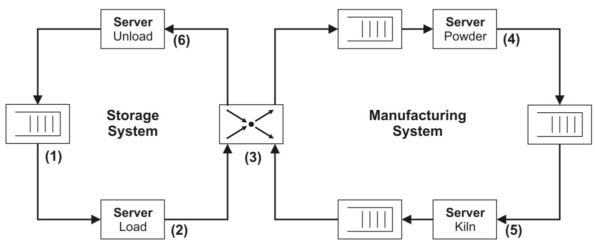

Fig. 6 Material flow system of coating plant.

As the benchmark problem a simple planning simulation of a production line for coating bicycle frames is chosen (see figure 6). Here, an amount of 15 frame carriers from the storage system (1) are loaded with raw frames of distinct bicycle types (2) and transported to a powder station (4) traversing a transport nodal point (3). Depending on the state of direction of the transport nodal point, the transport takes ten to 30 seconds and one powdering process lasts 5 minutes. Afterwards, the powdered frames are burned in a batch of three frames for 15 minutes in the kiln station(5). Then, the carriers are unloaded (6) and transported back to the storage system to be held for another cycle.
Within the planning simulation the coating plant needed to be dimensioned for a capacity of 150 coated frames per eight hours shift. First, a dimensioning of one kiln

[1] transaction-oriented; the world views are: event-oriented, transaction-oriented, activity-oriented and process-oriented ([6]).

station and one powder station is chosen. The planning simulation had been performed with Stateflow first and SimEvents later on. After the simulation runs both models yielded a potential manufacturing capacity of 90 coated frames per eight hours shift[2]. However, the potential manufacturing capacity did not meet the requirements. Thus, the resource capacities needed to be increased. Since the buffers were equipped with infinite capacity for the first planning phase the servers could be identified as bottlenecks quite easily.

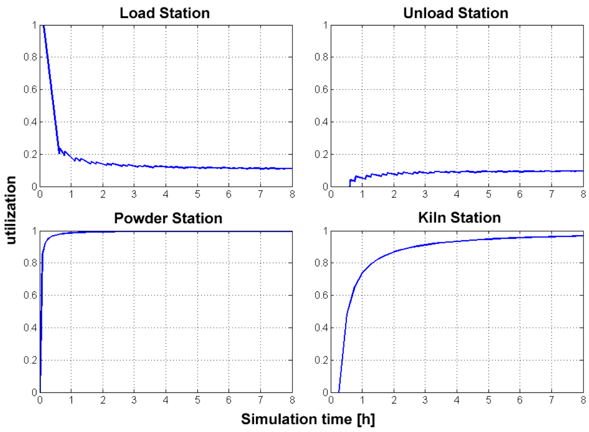

Fig. 7 Comparison of server utilization

In figure 7 the average utilization of all servers is illustrated. The graphs clearly show that the load and unload stations are underemployed, whereas the kiln and powder station are fully utilized. Hence, the capacities of these servers needed to be increased – a dimensioning of two powder and two kiln stations were chosen. This time, the Stateflow model yielded a potential manufacturing capacity of 180, whereas the simulation run with SimEvents resulted into 182 coated frames. The specification requirements were met.

3.4 Evaluation: Modeling Effort, Simulation Accuracy and Runtime

Premising a Stateflow library of fully implemented material flow components, both models hardly differ in *modeling effort*. Just the visual clear arrangement is limited due to additional synchronization lines within the Stateflow model.

First major differences can be observed in the *simulation accuracy*. Whereas both simulation runs of the first dimensioning step resulted in a potential manufacturing capacity of 90 coated frames the second dimensioning step differed from Stateflow with 180 to SimEvents with 182 coated frames. Since an analytical verification also resulted into 182 frames the SimEvents model is classified to be accurate.

Due to the state-based modeling with internal timing for realizing service times the execution of the Stateflow model has to be sampled equidistant in time. Time sampled execution of simulation models triggered by events on a continuous time base is generally inaccurate caused by the discrete time pattern. By reason of the choice of whole-numbered service times for the servers and a tenfold smaller time pattern a time accurate processing of the event-based material flow is theoretically possible – but practically not realizable with the Stateflow implementation as presented in section 3.1. On the one hand, a synchronization signal is used which needs to be hold for a minimum of one time step to avoid algebraic loops – cumulative delays are generated within this context. On the other hand, in comparison to an event-based processing just one workpiece can be moved per time. To maximally reduce these delays an appropriate minimum time pattern has to be chosen. In a subsequent simulation run using Stateflow the time pattern had been minimized to a tenth of the original size. Then, the simulation run also resulted into a potential manufacturing capacity of 182 coated frames.

Since the simulation with Stateflow is time sampled unnecessary state updates – at time points where no events occur – are being performed. In contrast, the processing of SimEvents models is based on an event list and computations are performed only when events occur. Hence, the simulation performance of both simulation tools should differ enormously. In table 1 the *runtimes*[3] measured for performing both simulation models are compared. Additionally, the runtimes for low and high time pattern precision for executing the Stateflow model is shown. As expected, there is a remarkable difference in runtime between SimEvents and Stateflow. Also, a linear dependency of time pattern precision and runtime using Stateflow is observable.

Tab. 1 Comparison of runtimes of SimEvents (SE) and Stateflow (SF); lp: low precision; hp: high precision.

SE	SF (lp)	SF (hp)
1.7 s	11.6 s	116.3 s

4 Summary and Outlook

In this contribution a comparative study of two DES tools for manufacturing simulation within a RCP approach has been presented.

First the state of the art as well as a future trend on model-based control design of discrete controls is introduced. A promising approach is the Simulation Model Based Control Approach following [5]. Demanding a strict separation of process and control as well as a consistent component-orient modeling of the process, a successive extension of simulation models from the early design phase as control programs for the operating phase in a coordinated software environment is provided.

For the comparative study two DES tools – Stateflow and SimEvents – are presented and benchmarked performing a planning simulation of a flexible manufacturing system with focus on modeling effort, simulation accuracy and runtime. Where both tools hardly

[2]Since the simulation runs include the start-up phase of the coating plant, a measurement of the stationary process – right after the start-up phase – need to be performed if a 24h operating mode is object of consideration ([9]).

[3]Just the time needed for the simulation run; no time for code compilation considered by using Stateflow.

differ in modeling effort, simulation accuracy and runtime are the deciding criteria. Since the processing of Stateflow models (as implemented in 3.1) is sampled in time, the simulation accuracy – as maintained within SimEvents simulations using event lists with a continuous time base – can just be achieved by choosing appropriate minimum sample times. But, with small sample times the computational effort and thus the simulation runtime increases. Therefore, SimEvents is the preferred tool for planning simulation. In contrast to Stateflow the applicability in the context of RCP is still to be proven.

Following the Simulation Model-Based Control Approach the planning simulation can be understood as the first step of the entire design process. The simulation model based on SimEvents can be used as the model of the uncontrolled process and extended step-by-step. In the next step the control model need to be designed, which has – depending on its complexity – a modular-hierarchical layout. Here, Stateflow can be used for designing the discrete control model as exemplified in [3]. The combination of Stateflow and SimEvents is shown in [10]. After finishing this step the designed control strategies can be validated using the process model within an entire system simulation.

To use the simulation model as a control program for the operating phase the process model need to be extended by low-level components for interfacing the real process as well as a real-time synchronization to provide an accurate processing of sensor signals. Implying that the platform used for control design matches the target hardware, this approach enables an implicit control syntheses ready for acting as a control program for the operating phase.

If the applicability of SimEvents in the context of RCP can be proven the last preparatory studies are completed. The subsequent study focuses on controlling flexible manufacturing systems using predictive simulations under real-time conditions to optimally react on continuous changes of order situations and machine disturbances or breakdowns.

5 References

[1] D. Abel and A. Bollig. *Rapid Control Prototyping - Methoden und Anwendungen*. Springer-Verlag Berlin, 2006.

[2] M. Kremp, T. Pawletta, G. Colquhoun: *Optimisation of manufacturing control strategies using online simulation*, In: Proc. 5th EUROSIM Congress on Modelling and Simulation, Paris, September 6-10, 2004.

[3] G. Maletzki. *Modellierung, Simulation und Steuerung von Materialflusssytemen mit Zustandsautomaten*. Diplomarbeit, Hochschule Wismar, 2003.

[4] S. Schreiber, M. Barth, A. Fay: *Modellierungs- und Simulationsmethoden: Vergleich und Bewertung anhand des Benchmarks ARGESIM C2*, In: Proc. ASIM-Treffen 2010 der Fachgruppen STS und GMMS, Ulm, 4.-5. Maerz, 2010.

[5] G. Maletzki. *Simulationsmodellbasiertes Rapid Control Prototyping von komplexen Robotersteuerungen*. Dissertation, Universitaet Rostock, 2010 (in progress).

[6] C. Stenzel. *Modellierung diskret-ereignisorientierter Systeme*. Techn. Bericht, Hochschule Wismar, FG CEA, 2009.

[7] The Mathworks. *Stateflow® and Stateflow® Coder™ 7 Users Guide*. The Mathworks Inc., Natick, 2010.

[8] The Mathworks. *SimEvents 3® Users Guide*. The Mathworks Inc., Natick, 2010.

[9] J. Košturiak and M. Gregor. *Simulation von Produktionssystemen*. Springer-Verlag Wien New York, 1995.

[10] The Mathworks, Webinar: *Timing ist alles - ereignisdiskrete Simulation mit Stateflow und SimEvents*, 15. Juni, 2010.