COMPARISON OF MATERIAL FLOW MODELS AND ACCELERATION OF THE MACROSCOPIC FLOW MODEL FOR VIRTUAL COMMISSIONING

Annika Kienzlen and Alexander Verl
Institute for Control Engineering of Machine Tools and Manufacturing Units
University of Stuttgart
Seidenstr. 36, 70174 Stuttgart, Germany
email: annika.kienzlen@isw.uni-stuttgart.de

KEYWORDS
material flow, virtual commissioning, real-time

ABSTRACT

Material flow simulations are of increasing importance for the virtual commissioning of control systems especially regarding plants in production. Material flow models for piece goods in literature can be mostly divided into four categories which differ in their suitability to the requirements of simulations for virtual commissioning. This paper presents an overview of the material flow models regarding model description and usage. It will be shown that the macroscopic flow model has potential for virtual commissioning. Improvements to achieve real-time capability through parallelization of the macroscopic flow model are introduced.

INTRODUCTION

Goods are the reason and the goal of every production. Consequently, it should be self-evident to take them into account for machine development and factory planning. However, there are different points of view.
On the one hand, material flow is the connecting element between all plant components. Scheduling of production steps depends on the processing times of the machines involved and on the transport time in between. Throughput times and stock levels are essential factors for the cost of production.
On the other hand, material flow as precise movement of goods and parts is also important for every single machine. The movement determines the possible geometries and the exact timing. Analysing the material flow is necessary to determine whether the control system is working as planned. The precise movement needs to be considered not only in the machines but also in between. This paper focuses on this side of the material flow.
The more goods pass a system, the higher is their importance but also the more complex are the calculations. Particularly in systems with a high throughput rate and fast moving goods, their precise movement is very important (Lacour (2012)). This is the case e.g. in bottling plants (Kadachi and Günthner (2001), Voigt (2004), Bernhard and Kahe (2008)). Though material flow simulation is already applied in many domains, there is a promising approach which could be improved for the application in virtual commissioning.

MATERIAL FLOW MODELS

Simulation models and tools have to be chosen according to the application's goals and requirements. Quite often, the particularities are not evident and one has to know in advance which differences exist (Roessler et al. (2015)). Material flow models can be sorted in four categories which are summarized in table 1 and will be explained below: event-discrete, kinematic, physics-based and macroscopic flow model (Scheifele et al. (2016)). The choice has significant influence on the modeling effort needed, the factors taken into account and the results available. For virtual commissioning (VC), the models have to be improved to fulfill all requirements.

Requirements Of Virtual Commissioning

In control engineering, testing of the control system on a virtual machine is referred to as virtual commissioning (Wünsch (2008)). VC is part of the development process of machines and plants and takes place before the test at the real prototypes. VC requires a virtual representation of the planned machine, a simulation model. The model consists of a representation of machines and environment (geometry), the possibilities of movement and the signals to communicate with the control system. The simulation has to (Röck (2007), Wünsch (2008), Hoher (2017))

- show whether the movements and the processing is possible as planned,
- prove the control system is working and is capable of controlling the process as planned,
- indicate whether the control system is able to detect or prevent errors in the process,
- have deterministic, stable calculations of high accuracy,
- be time-deterministic.

Table 1: Comparison of categories of material models in the light of virtual commissioning

	Considered parameters			Calculation time		Layout	Abstraction	Control connection
	Velocities	Geometries of elements	Forces, torques, masses	proportional to number	Real-time capable			
event-discrete	-	-	-	partly	+	1D– 2D	highest	-
kinematic	+	+	-	+	+	3D	middle	+
physics-based	+	+	+	+	-	3D	lowest	+
flow model	+	-	-	-	-*	2D	middle	-*

- not considered/not possible/No + considered/possible/Yes * current research

In production systems, the model is the base for a Digital Twin (Lechler et al. (2019)) which is a digital representation of and connected to a physical object (Fuller et al. (2020)). According to older definitions, the Digital Twin is identical to the simulation (Shafto et al. (2012)).
The most demanding variant of VC is a hardware-in-the-loop (hil) simulation where the whole control system including hardware and code is being tested on a simulation (Pritschow and Röck (2004)). The difference between real machine and simulation should not be noticeable for the control system. Therefore, the simulation has to provide the same inputs and outputs as the real machine in the same rate as the control system. The standard communication cycle time for a programmable logic controller is 1 ms which is referred to as real-time (Röck (2007)). This requirement can be particularly hard to meet (Scheifele et al. (2019)). For a hil simulation, the connection between simulation and control system should be the same than in the real scenario which usually is via field bus.
For the VC of plants and especially for single machines, the material flow is an important aspect. In this context, material flow is the exact movement of piece goods based on their physical properties and the interaction with their environment. While the machines are influenced directly by the control system, the influence on the piece goods is taking place only indirectly through the behaviour of the machines.

Event-discrete Material Flow Model

The event-discrete material flow model (EDMF) focuses on time-related parameters. Therefore, it is mostly used for the scheduling of transportation. The observation level of the EDMF is on the material exchange between machines or whole factories. Considered parameters are quantities of goods and time in form of duration and points in time. Therefore, the different production steps and their time have to be known or estimated. The time steps are not constant but depending on the events and much larger than the control rate. Since it is a logic based model, physical behaviour is excluded. The EDMF does not consider precise movements of the piece goods. The calculations are mainly based on stochastics considering average throughput times, failure rates and demand forecasts.
There are some other models on the level of observation of the EDMF like agent based and system dynamics (Roessler et al. (2015)). Because the level of detail of all of these models is not suitable for VC and they are less common, the models will not be discussed any further.

Kinematic Material Flow Model

A kinematic model provides position and velocity for each piece good. The calculations are based on trajectories of machines and piece goods (Scheifele et al. (2016)). The main application is to test whether the piece goods can be handled as planned and whether all the material paths are calculated correctly. Consequently, all the geometries of the machines, of the environment and of the products or products' parts have to be known. As soon as geometries are considered, collisions can be taken into account. The calculation of collisions can get computationally complex if there are many bodies (Hoher (2017)). However, the bodies only have to be kept separate and there are not any momenta as a result of the collisions. Since physical influences are ignored, complexity of the calculations is not seen as problem in research. The calculations are stable and time-deterministic, even within the control rate of 1 ms.

Physics-based Material Flow Model

Physics-based material flow models (PMF) have the lowest abstraction level of all four types taking into account most physical aspects. Usually, PMF are rigid body simulations without temperature differences. The model is based on Newton's law of motion (Göttlich et al. (2014)) where the double-derived position $x \in \mathbb{R}^3$ of each piece good i at time t is the sum of its forces $f_{n,i}$ proportional

Table 2: Usage of categories of material flow models

	Field	Requirement	Software	Thesis	Paper
event-discrete	production and process planning, logistics	time steps determined	SimulateFirst, pL-SIM, Sim 3D, FlexSim, AutoMod, Plant Simulation, Arena, Experior	Rybicka (2017), Kudlich (2000), Voigt (2004)	Wu and Wysk (1989), Huber and Dangelmaier (2009), Seidel et al. (2012), Glatt et al. (2018)
kinematic	hil, virtual commissioning	geometries designed, trajectories calculated	ISG-virtuos, WinMod	Hoher (2017)	Hoher and Verl (2012)
physics-based	test of control system	geometries and materials determined	Algoryx, Machineering, FEEsim, VisualComponents, Sim3D, Experior	Bender (2007), Spitzweg (2009), Lacour (2012)	Hoher et al. (2012), Göttlich et al. (2014), Neher and Lechler (2015), Ostergaard and Danjou (2017), Richter et al. (2018), Glatt et al. (2019)
flow model	conveyor belts, optimization	circular/ quadratic base, no 3D movements	less studied than the other models → not ready for industrial usage	Pfirsching (2018)	Hoher et al. (2012), Göttlich et al. (2014; 2015; 2018), Prims et al. (2019), Rossi et al. (2019)

to its mass m_i with $x_i(0) = x_{i,0}$ and $i = 1, ..., N_n$:

$$m_i \ddot{x}_i(t) = \sum_{n=1}^{N_f} f_{n,i}(t) \tag{1}$$

For angular movement with Euler's equation of motion, the moments have to be taken into account as well as the forces which do not apply in the center of mass. The derived angular velocity $\omega \in \mathbb{R}^3$ of each piece good i at time t in dependence of the piece good's moment of inertia θ_i consists of the cross product of the forces f and of the displacement between the point of application and the center of mass $r \in \mathbb{R}^3$ on one hand, and the sum of moments $\tau_{m,i}$ on the other hand with $\omega_i(0) = \omega_{i,0}$ and $i = 1, ..., N_n$:

$$\theta_i \dot{\omega}_i(t) = \sum_{n=1}^{N_f} f_{n,i}(t) \times r_{n,i}(t) + \sum_{n=1}^{M_f} \tau_{m,i}(t) \tag{2}$$

In each time step dt, it has to be verified if there is any collision between any piece good and its environment or another piece good. In comparison to the machine's movement, the piece goods' movement is not controlled directly by the control system but is a result of their environment. The geometry of the individual piece goods have to be taken into account. Consequently, the computational effort is increasing with increasing number of piece goods even if bounding boxes instead of complex geometries are used for the collision calculation. The collision detection and resolution is often the bottleneck in the simulation of material flows (Hoher et al. (2013)). Consequently, the most information is provided but there is a risk of not meeting the requirement of time-determinism.

Macroscopic Flow Model

The latest idea is the examination of piece goods analogous to fluids which leads to good results (Göttlich et al. (2014), Rossi et al. (2019)). The piece goods are not considered separately but as a density distribution $\rho(x,t) : \mathbb{R}^2 \times \mathbb{R}$ depending on place x and time t. Consequently, the movement of the piece goods results from the static environment and the dynamic interaction between the piece goods. The static velocity v^{stat} depends on the position x of the piece good. It includes restrictions like static elements in the environment but also predetermined velocities for example those of a conveyor belt. The dynamic velocity as indirect collision consideration gets active as soon as a limit density ρ_{max} is exceeded. It consists of a Heaviside function H and the term of force I. The relationship can be described as a non-local hyperbolic partial differential equation with conservation of mass and therefore conservation of density with:

$$\partial_t \rho + \nabla \rho(v^{dyn}(\rho) + v^{stat}(x)) = 0 \tag{3}$$

$$v^{dyn}(\rho) = H(\rho - \rho_{max}) I(\rho) \tag{4}$$

$$\rho(x, 0) = \rho_0(x), x \in \mathbb{R}^2 \tag{5}$$

The equation has been numerically solved with the Roe method with dimensional splitting which is time-deterministic (Göttlich et al. (2014)). Decisive for the

computational effort is the discretization in time and space. To represent the machines three-dimensional the MFM has to be combined with another model.

Comparison

The models differ widely especially in their application, which is outlined in table 2 (non-exhaustive and restricted to control engineering). EDMF will be used at first during development, often for supply chain management. The application in control engineering is very limited. The kinematic models can be applied after the machines or plants are designed. They allow a first test of the interaction between mechanic construction and control code. To use a PMF, the development process has to be nearly finished. Details of the machines, the piece goods and the control system have to be decided. The MFM can be used in a similar phase but the application is restricted: the piece goods only move in a plane, there is no overlapping or tilting. Moreover, the piece goods need a circular or quadratic base area (Prims et al. (2019)). Otherwise their collision behaviour is different, they can get wedged easily, which is not reflected in the model. It is possible as well to combine (parts of) models. For instance a kinematic model can be combined with some simple physical principles as gravity (Hoher and Verl (2012), Gulan et al. (2014)) or some physical properties can be taken into account for an EDMF (Westbrink and Schwung (2018)).
According to table 3, the EDMF and the kinematic model provide less information for the VC of control systems. The PMF achieves the best results in terms of accuracy. The MFM still achieves more results than the kinematic model, because the trajectories have to be given for this one and will not be adjusted.

Table 3: Evaluation of categories of material models in the light of VC

	Advantages	Disadvantages
event-discrete	Computing efficient, overview	not detailed, no precise movement, no control connection
kinematic	descriptive representation	not detailed, no collision reactions, trajectories needed
physics-based	highest accuracy, clear representation	calculation-intensive
flow model	independent of material number, high accuracy	for few goods relative calculation-intensive

The computational effort for the PMF increases with increasing number of piece goods which can lead to computational effort and duration needed for the calculations too high for the real-time requirements of a control system. The MFM on the other hand does not calculate collisions directly, it does not even consider single piece goods. Consequently, the computational effort is independent of the number of piece goods (cf. figure 1). If it was possible to reduce the calculation time, the MFM could be a valid alternative or complement to the PMF. Since there has not been much research on the MFM, there still is lots of potential. In the following, details of the MFM will be presented and possibilities of acceleration proposed.

Figure 1: Calculation time dependent on the number of piece goods

CONCEPT OF ACCELERATION FOR THE MACROSCOPIC FLOW MODEL

The density distribution of the MFM according to Göttlich et al. (2014) consists of $N_x \times N_y$ cells with surface $dx \times dy$. In each of these cells, the density $\rho(t)$ has to be calculated in each time step dt. Therefore, the dynamic density ρ^{dyn} resulting from the current dynamic velocity v^{dyn} is calculated at first, and the static density ρ^{stat} resulting from the static velocity v^{stat} afterwards. In case of the static density, the adjacent cells have to be taken into account. As far as the dynamic velocities are concerned, the parameter ϵ is weighting the repulsive forces I and the smoothing kernel η is weighting the density conditions within a convolution as shown in equation 6.

$$I(\rho) = -\epsilon \frac{\nabla(\eta * \rho)}{\sqrt{1 + ||\nabla(\eta * \rho)||_2^2}} \tag{6}$$

To reduce the calculation time, several approaches can be considered:

- The computational effort could be reduced by increasing the cell size. To achieve optimal results, the cell size represented by dx and dy has to be chosen in proportion to the time step dt (Rossi et al. (2019)). Since the time step is determined by the communication cycle time of the control system, the density cell size cannot be adapted without downgrading the results.
- The convolution has a particularly high computational effort. The smoothing kernel η is weighting the

whole density distribution to determine the effects at place x. If the considered area is big, the effects of the distant cells are rather small. Hence, consideration should be given to the possibility of restricting the force term to a smaller area.

- If the whole density distribution is taken into account for the convolution, there is another possibility to reduce its calculation time. Since the calculation time depends for each cell only on the density distribution of the previous time step, it should be possible to schedule their calculation in parallel. It could be beneficial to transfer the parallel calculation on a graphics card (GPU) (Panchatcharam et al. (2013)). This seems to be the most promising approach which will be presented in the following. The PMF could not reach time-determinism even with parallelization.
- Depending on the field of application and the goal of the simulation, the information of the material flow are not needed in the control system's rate. In this case, the material flow simulations can run parallely with a slower simulation rate. To do so a co-simulation environment with a multi-rate approach is needed (Scheifele and Verl (2016), Scheifele et al. (2019)).
- There is another method based on a co-simulation environment: co-simulation with prediction of coupling signals. If the material flow is mainly continuous, the material flow simulation can run in parallel with a slower simulation rate and the signals of the material flow model which are relevant for the control system are predicted for the faster steps of the control system (Günther (2017), Kienzlen et al. (2020)). The last two solutions are also applicable for the PMF.

IMPLEMENTATION

Compute unified device architecture (CUDA) is NVIDIA's GPU programming model (NVIDIA (2018)) which can be implemented in straight forward C, C++ or Phython. In CUDA, one function ("kernel") is called from the CPU ("host") and invokes multiple calculations ("threads") on the GPU ("device")(Zou et al. (2009)). For better manageability, the threads are structured in three dimensional blocks which in turn are structured in three dimensional grids. The threads of one block run in parallel, the blocks can run in parallel or sequentially depending on the implementation and the hardware (Luebke (2008)). These simple parallelizations are the advantage of calculations on the GPU which come with the disadvantage of having to copy data between CPU and GPU.
The implementation of the MFM has two main parts as can be seen in figure 2: initialization and simulation step. The initialization of the model is done once at the beginning of the simulation. It consists of the initialization of the initial density distribution ρ_0, of the static velocity field and of the smoothing kernel η. This is scheduled on the central processing unit (CPU). Afterwards, a loop starts which is executed for each simulation step. Each step, the dynamic density field ρ^{dyn} is calculated whereby a division is applied in the two spatial directions x and y. This step mainly consists of the convolution $D_j = \eta * \rho_i$. This part leads to the highest computational effort why it is scheduled on the GPU. The static density ρ^{stat} is less demanding, therefore it can be scheduled on the CPU. The four density fields (dynamic in x, dynamic in y, static in x and static in y) get combined to one overall density distribution. Finally, the overall density distribution has to be given to another part of the simulation. This usually takes place on the CPU, therefore, at this point at the latest, the information has to be copied to the CPU. Alternatively to the output, there could be a post-processing step to calculate relevant signals for the control system or information for another simulation.

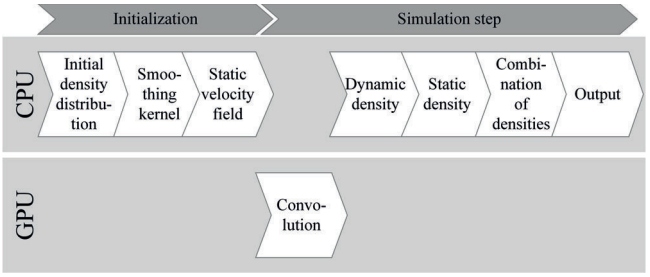

Figure 2: Implementation of the convolution of a simulation step with CUDA

When implementing with CUDA, there are different sorts of memory (cf. figure 3): host memory on the CPU and device memory on the GPU. To use data on the GPU, it has to be copied to its memory. On the GPU, there is global memory which can be accessed from all of the threads. This is the standard sort of memory which need not be declared explicitly. When using global memory, one has to make sure that different threads will not access the same data. The opposite is local memory, which can be accessed only from one thread. In between, there is shared memory which can be accessed within one block. Its amount is restricted to an overall limit depending on the graphics card's prop-

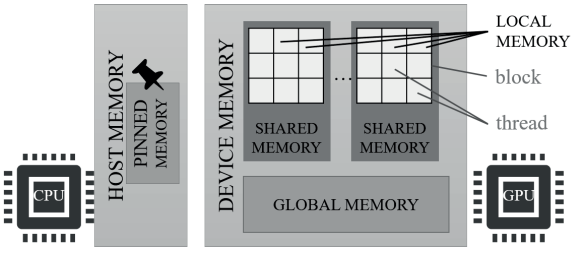

Figure 3: Different sorts of memory within CUDA

erties. Computational effort is needed to copy data from the global to the shared memory and back but on the shared memory, threads from different blocks will not interfere. The last sort of memory is important for the data exchange between CPU and GPU: Pinned memory has to be defined on the CPU. This memory space is specified for data which has to be exchanged between CPU and GPU frequently.
To improve the memory usage of the MFM, there can be taken different steps:

1. The convolution D_j can be scheduled on the GPU.
2. The density distribution of the previous step ρ_{t-1} can be copied to the shared memory. This way, the different blocks do not have to access the global memory as often and moreover, the different blocks will not interfere to access the same data.
3. A local variable for each cell of the convolution D_j can be defined so all the interim results can be calculated locally. The interim results have to be copied to the global memory afterwards but with the calculation locally the global memory has to be accessed less frequently.
4. Interim results can be calculated in advance and be reused later on. This idea is also valid for the calculation on the CPU. For the calculation on the CPU, this means calculating the interim results only once and reusing them throughout the loops. For the calculation in CUDA, this means calculating the interim results in all threads but only once per thread.
5. An alternative to scheduling only the convolution on the GPU is scheduling the whole simulation step excluding the output on the GPU as depicted in figure 4. This way the parallelization can be applied on bigger parts. The disadvantage is that more information has to be copied on the GPU including the static velocity field in two spatial directions.

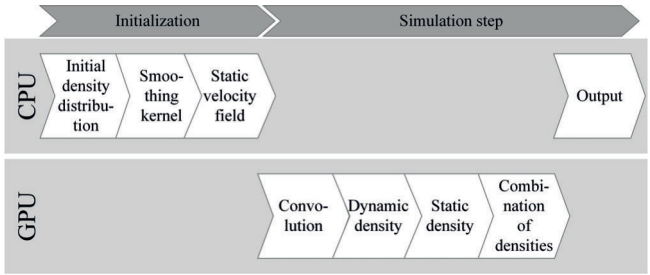

Figure 4: Implementation of the whole simulation step in CUDA

To get an estimation of the computing times, a simple example was implemented analogously to (Hoher et al. 2012, Göttlich et al. 2014): Circular piece goods are transported on a conveyor belt against a singularizer and are thus deflected. The discretization was chosen with 160 x 220 cells. All of the measurements depend on this factor. With the motivation that in future as little special hardware as possible should be required for the simulation, a standard PC was used (Intel® Core™ i7 with a possible processor speed of 2.8 GHz, 4 cores, 8 logical processors, 8.0 GB of RAM). For parallelization in CUDA a graphics card from NVIDIA is required, GeForce 940MX was used here (total amount of global memory 2048 MB, 128 CUDA Cores/MP, Max. dimension size of a thread block (1024, 1024, 64)). The double parallelized implementation is more than ten times faster than the pure implementation in C++ in this scenario. The usage of shared memory reduces the calculation time by a further 20 %, the implementation with local memory by further 15 % and interim results by further 18 % as shown in figure 5. The computation of the whole simulation step on the GPU does not change much. This is because lots of additional information have to be copied on the GPU. If it would be copied on the GPU initially, the results would be much better.

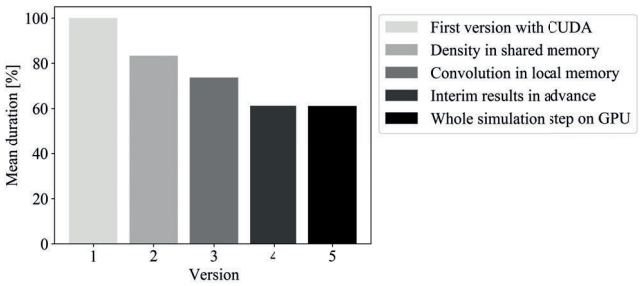

Figure 5: Calculation time of a simulation step with CUDA

CONCLUSION

The simulation of material flow is an important part for the VC of plants. There are four different categories of material flow models, three of which can be used for this purpose and two - the physics-based models and the macroscopic flow model - obtain results with satisfying accuracy. The computational effort of the physics-based models depends on the number of piece goods why the calculation time exceeds the real-time of the control system for a great number of piece goods. The macroscopic flow model seems to be a valid alternative in this case. The initial calculation time is too high so far but the parallelization on the GPU supports the objective of achieving a simulation step in real-time of control. There are already some improvements but more work is needed to fully achieve this goal.

FUTURE WORK

In order to use the macroscopic flow model effectively for the virtual commissioning of control systems, a post-

processing step is necessary to generate signals. This could be, for example, the calculation of collisions in the material flow, the calculation of the position of individual piece goods in certain areas or the calculation of the outflow. After the design and implementation of these calculations, the macroscopic flow model can be integrated in a hil simulation. As a next step, it would be interesting to combine the physic-based model and the macroscopic flow model and to switch between the models depending on the relevant information.

ACKNOWLEDGEMENT

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the project 327964174 (OptiPlant) in cooperation with the University of Mannheim. The authors would like to thank for the financial support.

REFERENCES

Bender J., 2007. *Impulsbasierte Dynamiksimulation von Mehrkörpersystemen in der virtuellen Realität.* Thesis, Universität Fridericiana zu Karlsruhe. doi: 10.5445/KSP/1000006079.

Bernhard J. and Kahe T., 2008. *Simulation und Modellmanagement in der Getränkeindustrie.* In M. Rabe (Ed.), *Advances in simulation for production and logistics applications*, Fraunhofer IRB Verlag, Stuttgart. ISBN 3816777988, 199–208.

Fuller A.; Fan Z.; Day C.; and Barlow C., 2020. *Digital Twin: Enabling Technology, Challenges and Open Research.* doi:10.1109/ACCESS.2020.2998358.

Glatt M.; Kasakow G.; and Aurich J.C., 2018. *Combining physical simulation and discrete-event material flow simulation.* vol. 72, 420–425. doi:10.1016/j.procir.2018.03.054.

Glatt M.; Kull D.; Ravani B.; and Aurich J.C., 2019. *Validation of a physics engine for the simulation of material flows in cyber-physical production systems.* vol. 81, 494–499. doi:10.1016/j.procir.2019.03.125.

Göttlich S.; Hoher S.; Schindler P.; Schleper V.; and Verl A., 2014. *Modeling, simulation and validation of material flow on conveyor belts. Applied Mathematical Modelling*, 38, no. 13. doi:10.1016/j.apm.2013.11.039.

Göttlich S.; Klar A.; and Tiwari S., 2015. *Complex material flow problems: A multi-scale model hierarchy and particle methods. Journal of Engineering Mathematics*, 92, 15–29. doi:10.1007/s10665-014-9767-5.

Göttlich S.; Pfirsching, Marion ; and Hristov, Jordan, 2018. *A micro-macro hybrid model with application for material and pedestrian flow. Cogent Mathematics & Statistics*, 5. doi:10.1080/25742558.2018.1476049.

Gulan S.; Odefey U.; Bär T.; Beesten H.; Hämmerle H.; Kärcher B.; and Riedl M., 2014. *Physikbasierte Simulation im Anlagenbau.* In *Simulation technischer Systeme - Grundlagen und Methoden in Modellbildung und Simulation.* 57–62.

Günther F.C., 2017. *Beitrag zur Co-Simulation in der Gesamtsystementwicklung des Kraftfahrzeugs.* Thesis, Technische Universität München.

Hoher S., 2017. *Ein gekoppeltes Materialflussmodell zur durchgängigen Entwicklungsunterstützung von Materialflusssteuerungen: Stuttgarter Beiträge zur Produktionsforschung.* Thesis, Universität Stuttgart. doi: 10.18419/OPUS-9309.

Hoher S.; Neher P.; and Verl A., 2013. *Collision: Impossible - Echtzeitfähige 3D-Kollisionkontrolle bei mehrkanaliger Bearbeitung.* In G. Frey; W. Schumacher; and A. Verl (Eds.), *Elektrische Automatisierung - Systeme und Komponenten.* Berlin, 1–10.

Hoher S.; Schindler P.; Göttlich S.; Schleper V.; and Röck S., 2012. *System Dynamic Models and Real-time Simulation of Complex Material Flow Systems.* In H.A. ElMaraghy (Ed.), *Enabling Manufacturing Competitiveness and Economic Sustainability*, Springer Berlin Heidelberg. 316–321. doi: 10.1007/978-3-642-23860-4_52.

Hoher S. and Verl A., 2012. *Multi-Simulator-Materialflusssimulation für die virtuelle Inbetriebnahme. G Frey, W Schumacher, A Verl, SPSIPC-DRIVES*, 387–396.

Huber D. and Dangelmaier W., 2009. *Controlled simplification of material flow simulation models.* In *Proceedings of the 2009 Winter Simulation Conference.* IEEE, 839–850. doi:10.1109/WSC.2009.5429707.

Kadachi M. and Günthner W., 2001. *Simulationsgestützte Planung und Nutzung von Getränke-Abfüllanlagen.* Tech. rep., Lehrstuhl für Fördertechnik Materialfluss Logistik, Garching.

Kienzlen A.; Scheifele C.; and Verl A., 2020. *Predicting coupling signals in a material flow real-time co-simulation with a Kalman filter.* In Teti R.; D'Addona D. M. (Ed.), *13th Conference on Intelligent Computation in Manufacturing Engineering. Procedia CIRP*, vol. 88, 9–14. doi:10.1016/j.procir.2020.05.002.

Kudlich T., 2000. *Optimierung von Materialflußsystemen mit Hilfe der Ablaufsimulation.* Thesis, Technische Universität München.

Lacour F.F., 2012. *Modellbildung für die physikbasierte virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen.* Thesis, Technische Universität München.

Lechler T.; Fischer E.; Metzner M.; Mayr A.; and Franke J., 2019. *Virtual Commissioning - Scientific review and exploratory use cases in advanced production systems*. *Procedia CIRP*, 81, 1125–1130. doi:10.1016/j.procir.2019.03.278.

Luebke D., 2008. *CUDA: Scalable parallel programming for high-performance scientific computing*. In *5th International Symposium on Biomedical Imaging*. IEEE, 836–838. doi:10.1109/ISBI.2008.4541126.

Neher P. and Lechler A., 2015. *Using game physics engines for hardware-in-the-loop material flow simulations*. In *International Conference on Advanced Intelligent Mechatronics*. IEEE, 1002–1007. doi:10.1109/AIM.2015.7222670.

NVIDIA (2018), 2018. *CUDA C Programming Guide*. NVIDIA Corporation, v9.2 ed.

Ostergaard N.H. and Danjou S., 2017. *On numerical simulation of the dynamics of bottles in conveyor systems*. *Journal of Applied Packaging Research*, 9, no. 3.

Panchatcharam M.; Sundar S.; Vertrivel V.; Klar A.; and Tiwari S., 2013. *GPU Computing for meshfree particle method*. *International Journal of Numerical Analysis and Modeling*, , no. 4, 394–412.

Pfirsching M., 2018. *A multi-scale model for material flow problems based on a non-local conservation law: simulation and optimization*. Thesis, Universität Mannheim.

Prims D.; Kötz J.; Göttlich S.; and Katterfeld A., 2019. *Validation of Flow Models as New Simulation Approach for Parcel Handling in Bulk Mode*. *Logistics Journal*. doi:10.2195/lj_Rev_prims_en_201906_01.

Pritschow G. and Röck S., 2004. *"Hardware in the Loop" Simulation of Machine Tools*. vol. 53. ISSN 00078506, 295–298. doi:10.1016/S0007-8506(07)60701-X.

Richter C.; Katterfeld A.; and Pusch M., 2018. *Multilevel DEM-Simulation des Bewegungsverhaltens von Getränkeflaschen im Pulk*. *Logistics Journal*, , no. 11. doi:10.2195/lj_NotRev_richter_de_201811_01.

Röck S., 2007. *Echtzeitsimulation von Produktionsanlagen mit realen Steuerungselementen*. Thesis, Universität Stuttgart.

Roessler M.P.; Riemer J.; and Mueller M., 2015. *Decision support for choosing an appropriate simulation method for dynamic material flow analysis*. *Journal of Industrial and Intelligent Information*, 3, no. 4.

Rossi E.; Kötz J.; Goatin P.; and Göttlich S., 2019. *Well–posedness of a non-local model for material flow on conveyor belts*. *ESAIM: Mathematical Modelling and Numerical Analysis*. ISSN 0764-583X.

Rybicka J., 2017. *Modelling Flexible Manufacturing Systems through Discrete Event Simulation*. PhD, Cranfield University.

Scheifele C.; Lechler A.; and Verl A., 2016. *Materialflussmodelle für die HiL-Simulation*. *wt Werkstattstechnik online*, 106, no. 3, 119–124.

Scheifele C. and Verl A., 2016. *Hardware-in-the-Loop Simulation for Machines Based on a Multi-Rate Approach*. *9th EUROSIM Congress on Modelling and Simulation*.

Scheifele C.; Verl A.; and Riedel O., 2019. *Real-time co-simulation for the virtual commissioning of production systems*. *Procedia CIRP*, 79, 397–402. ISSN 22128271. doi:10.1016/j.procir.2019.02.104.

Seidel S.; Donath U.; and Haufe J., 2012. *Towards an integrated simulation and virtual commissioning environment for controls of material handling systems*. In *Winter Simulation Conference (WSC)*. IEEE. doi:10.1109/WSC.2012.6465081.

Shafto M.; Conroy M.; Glaessgen E.; Kemp C.; LeMoigne J.; and Wang L., 2012. *Modeling, Simulation, Information Technology and Processing Roadmap: Technology Area 11*.

Spitzweg M., 2009. *Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen*. Thesis, Technische Universität München.

Voigt T., 2004. *Neue Methoden für den Einsatz der Informationstechnologie bei Getränkeabfüllanlagen*. Thesis, Technische Universität München.

Westbrink F. and Schwung A., 2018. *Virtual Commissioning Approach based on the Discrete Element Method*. In *16th International Conference on Industrial Informatics*. IEEE, 424–429. doi:10.1109/INDIN.2018.8471954.

Wu S.Y.D. and Wysk R.A., 1989. *An application of discrete-event simulation to online control and scheduling in flexible manufacturing*. *International Journal of Production Research*, 27, no. 9, 1603–1623. doi:10.1080/00207548908942642.

Wünsch G., 2008. *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. Thesis, Technische Universität München.

Zou C.; Xia C.; and Zhao G., 2009. *Numerical Parallel Processing Based on GPU with CUDA Architecture*. In *International Conference on Wireless Networks and Information Systems*. IEEE, 93–96. doi:10.1109/WNIS.2009.46.