Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-10 (2025) 554–559

Efficient simulation tools for digital twinning in cyber-physical systems: new challenges and a proposed road map

Chiara Cimino * **Federico Terraneo** ** **Gianni Ferretti** ** **Alberto Leva** **

* *Dipartimento di Meccanica, Politecnico di Milano Italy (e-mail: chiara.cimino@polimi.it).*

** *DEIB, Politecnico di Milano, Italy (e-mail: {federico.terraneo,gianni.ferretti,alberto.leva}@polimi.it).*

Abstract Creating digital twins (DT) of cyber-physical systems (CPS) presents unique challenges that current modelling and simulation (M&S) tools are not fully equipped to address. While some aspects of this challenge have been well explored, such as multi-domain modelling, real-time data synchronisation, and predictive analytics using AI and machine learning, others have received less attention. Two important, yet in our opinion underexplored, areas are (i) the handling of large systems of differential and algebraic equations (DAE) coupled with event-based dynamics, as Finite State Machine (FSM), and (ii) the need for not only efficient simulation code but also for efficiency in the process of generating that code. We discuss the challenges related to these topics in a CPS-based context, and sketch out a roadmap for developing solutions that enhance the effectiveness and scalability of M&S tools.

Keywords: Digital twins, Cyber-physical systems, Dynamic modelling and simulation, Multi-domain modelling, Computationally efficient simulation.

1. INTRODUCTION

In the past decade, the concepts of Digital Twins (DT) and Cyber-Physical Systems(CPS), (Kritzinger et al., 2018) and (Monostori, 2015), have driven the development of new modelling, simulation and control approaches within the context of Industry 4.0. Originally, CPS was envisioned as an autonomous system capable of processing and communicating data to create a digital representation of real-world processes. Over time, however, the concept has evolved, expanding both in terms of functionalities and of communication capabilities between systems. Initially, CPS referred to a physical system with analytical capabilities and the ability to communicate with other connected systems, while considering various levels of detail of the physical system itself (e.g., equipment, machines, or production lines). When the reference system is a production system, such as a manufacturing plant, the concept evolves into the Cyber-Physical Production System (CPPS) Graessler et al. (2024).

The introduction of CPPS has brought new challenges, including (i) integrating CPS (sub-)components of a CPPS through data, models, and analyses, and (ii) developing predictive models that support what-if analyses related to this integration. This complexity also applies to the idea of System of Systems (SoS) (Graessler et al., 2024), which refers to a group of interconnected CPPSs. Each CPPS, i.e. within an SoS, must manage itself dynamically and communicate its production analyses and predictions to other CPPSs that belong i.e. to the same SoS. In addition to the concepts outlined, analyses should be conducted by a digital counterpart of the system, tailored to the appropriate level (CPS, CPPS, or SoS). Specifically, every physical system following the CPS architecture should have a corresponding digital counterpart, as defined in (Kritzinger et al., 2018). This counterpart may take different forms of a Digital Model (DM), a Digital Shadow (DS), or a Digital Twin (DT):

- a Digital Model (DM) consists of manually updated models that replicate the system;
- a Digital Shadow (DS) includes models that automatically update based on real-time data;
- a Digital Twin (DT) not only updates automatically but can also perform actions on the physical system.

Thus, the role of a digital counterpart varies depending on its type. A DM is used primarily for monitoring, a DS can perform what-if analyses, and a DT can also control the physical system. Eventually, when a DT uses what-if analyses to make decision, one can also talk of *symbiotic simulation* (Aydt et al., 2008).

Furthermore, multiple digital counterpart – of any type – can be created for the same CPS or CPPS throughout the production lifecycle. To manage multiple DTs, tools are needed to connect data, models and analyses at various levels of detail, both within the same CPS and across interconnected CPSs. Recent developments to facilitate the integration of data and models are the Digital Thread (Pang et al., 2021; Negri and Abdel-Aty, 2023) or the use of ontology in the creation of DTs, which employs semantic relationships to formalize the knowledge of complex systems, i.e., the Cognitive Digital Twin (CDT),

10.1016/j.ifacol.2025.09.095

which applies such formalization to improve system understanding and performance (Karabulut et al., 2023; Jinzhi et al., 2022). Nevertheless, the integration among simulations of such models at different levels of detail has still not been addressed enough to establish solid foundations on which CPS/CPPS/SoS can be developed, in line with their literature definitions.

This paper focuses on this latter research area and on the simulation role of DT(s), considering their nested and hierarchical structure within a CPS/CPPS/SoS, and addresses a particular yet fundamental point: the need for simulation tools that can (i) operate online keeping the pace of the physical system and (ii) integrate continuous-time and event-based process descriptions.

2. RELATED WORK

The manufacturing environment contains both time- and event-driven dynamics. The former type includes phenomena like, for example, the behavior of a mechanical component governed by the laws of physics, while the latter type involves discrete occurrences, such as hitting an end stop or computing a new control signal when necessary. Continuous time-based dynamics is typically modeled using Differential Algebraic Equations (DAE) systems (Rosen and Pattipati, 2023), while event-based dynamics is generally represented using Finite State Machines (FSM), automata, or Petri nets (Wainer and Mosterman, 2018).

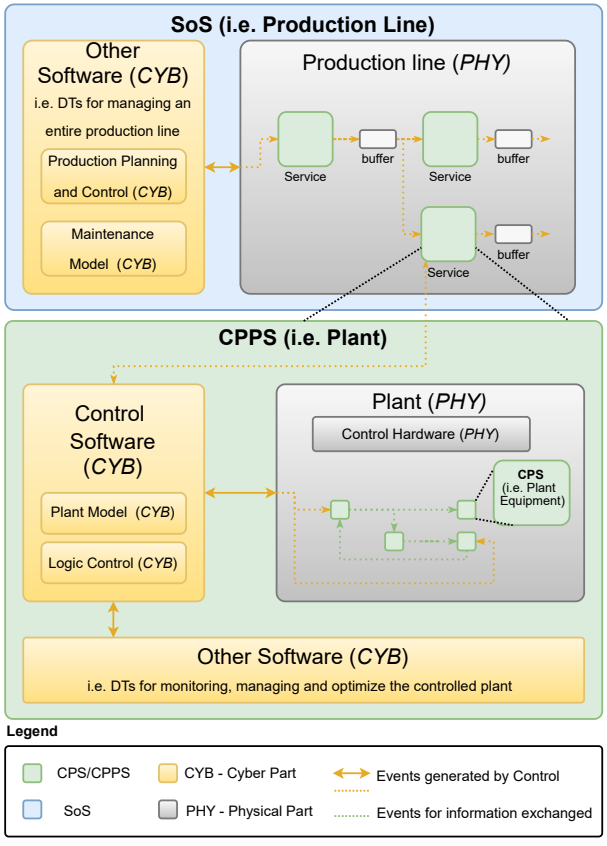

Figure 1. Illustrative example of CPS,CPPs,SoS-DT relationships to evidence the nested composition of CYBer and PHYsical elements, DTs included.

As discussed in Section 1, a production system, such as a plant – see Figure 1 – can be referred to as a CPPS, i.e., a system made up of a set of CPSs (i.e., tools and equipment with their cyber components that should communicate within the CPPS). At this level, DAE systems are commonly used for designing and reconfiguring the plant's dynamic behaviour. They help determine the correct system parameters for control purposes as does the Plant model in Figure 1, which can also be used for modulating control and predicting the plant's status. Additionally, DAE models are used to simulate failures in what-if *scenarii*. Similarly, at the equipment level, if plant machinery needs to be controlled together, iterative simulations at the CPS level may be necessary.

Discrete-event models are widely used for various purposes and at different levels of system detail. As shown in Figure 1, these models can be applied at the control level to regulate the control logic of a CPS, such as a plant or equipment control logic. At the CPS level, they can also be used for monitoring, maintenance, and management by referencing the system's operational status, which in turn relates to the physical system signals, including control signals. Moreover, event-based dynamics are prevalent in production planning and control strategies at the System of Systems (SoS) level, as seen in Figure 1. In this case, Discrete Event System (DEVS) specification (Zeigler et al., 2000) —- based on FSMs and automata -— is used in many commercial applications (for a list refer to Zahid et al. (2024)) to simulate a production system line, address operation and scheduling challenges, and evaluate the production flow over time.

As can be seen, different models are employed at varying levels of detail based on the scope of their application. Therefore, when simulating systems throughout their lifecycle –particularly through Digital Twins (DTs) –and considering the hierarchical system levels, the following research questions arise.

(1) When do CPS/CPPS or even SoS systems require simulation?
(2) What level of simulation depth is needed for each hierarchical level?

To address these questions, we can suppose that simulation is generally necessary when (i) the monitored parameters deviate significantly from expected forecasts (for example, simulating the system under nominal conditions), or (ii) different *scenarii* need to be explored. Regarding the second question, one must consider the correct hierarchical level, as the purpose of simulation varies depending on where it is conducted. In the following, this paper elaborates the use of simulation at the CPPS and SoS levels, under the hypothesis that a CPPS can be hierarchically composed by CPSs that has similar functionality but at lower levels.

The CPPS level

At the CPPS level, a plant can be described by a set of interconnected models (green blocks and green lines in Figure 1). These models may include a combination of DAE models to describe the dynamic behavior of the process and FSMs to represent various operating states

of the system (e.g., idle, working, failure, energy-saving). This modeling strategy is used not only to detail the possible states of a CPPS but also to inform decisions, such as maintenance strategies (Magnanini and Tolio, 2020). Therefore, when a simulation is required, the DAE models describing the CPPS must be consistent with one another, including any fault models. The system's machine state from other analyses must also be created and updated in sync with the DAE models, as depicted by the bidirectional yellow arrow at the plant level in Figure 1. At the CPPS level, it is crucial to account for the dynamics of the real system and the effects these dynamics have on machine states. If the system is in an unexpected state, the CPPS must track this and incorporate it into subsequent simulations and predictions.

The SoS level

At the SoS level, when a simulation is required—for example, when the service time of a CPPS deviates from the expected—SoS-level simulations should be conducted using models specific to the SoS. Additionally, these simulations should incorporate analyses from the CPPS level if necessary to investigate whether something is malfunctioning at the specific plant level. In theory, simulations should allow for the integration of CPPS- and CPS-level analyses at the SoS level when required.

3. THE CHALLENGES TO FACE

As shown by the brief literature review above, manufacturing systems are inherently CPSs, and the DTs required for their engineering heavily rely on dynamic M&S. It is now time to explore the entailed challenges in some detail — especially those addressed by the presented research.

3.1 Variable structure and heterogeneous connectivity

Manufacturing systems are made of components that can be physical (e.g., machines) or cyber (e.g., control blocks). Due to the increasing need for flexibility, these components are often rearranged into new configurations. This subjects them to different boundary conditions. For instance, a component might have some quantities prescribed from outside in a certain configuration, but not in another. To enable efficient model-based engineering, models must be modular, and the components that make them up must be organised in well structured libraries. Given the *scenario* just sketched, models of physical components need to be developed independently of how they are interconnected with other components — in other words, using *a-causal* connectors. This requirement rules out modelling paradigms where models are connected through inputs and outputs (*causal* connectors). In contrast, however, input/output connections are entirely natural when dealing with cyber components, such as the control blocks mentioned above.

As such, any tool designed to address the digital twinning of CPSs must support both causal and a-causal component connectivity, and allow for the integration of the two.

3.2 Role of first principles

Models for physical components can be broadly classified into two types: first-principle (FP) models and data-based (DB) models. The former are based on the fundamental laws of physics that govern the behaviour of the modelled object, while the latter rely on identifying the said behaviour from recorded data. As no single approach can be universal, FP and DB models must coexist. However, in model-based engineering, FP models play a particularly important role for at least two reasons. The first, more obvious reason is that when designing something new, no data is available. The second, more subtle reason is that if one wants to identify a model of a given component from data *gathered in a certain condition* and then use that model when the component operates under *different* conditions, it is crucial to ensure that the said data is in no sense condition-specific — quite difficult to guarantee.

As such, any tool designed to address the digital twinning of CPSs must effectively support FP modelling, which – as discussed further in the following – primarily involves systems of differential and algebraic (DAE) equations.

3.3 DAE/DEVS coexistence

In the CPS driven context, cyber components are typically designed using various formal methods, which are not discussed in detail here. The most common of these are continuous- and discrete-time dynamic systems, as well as automata. Regardless of the adopted formalism, these components are implemented as algorithms executed by multiple processors, either periodically or in response to specific events. The natural counterpart of such components is given by DEVS models, as these can represent both automata, typical for example of logic controls at any level, and queuing networks, typical of manufacturing system models. Notably, the IEC 61131 standard International Electrotechnical Commission (2013) defines programming languages that align with these formalisms, while the IEC 61499 one introduces the concept of Function Block (FB) as the atomic control element. An FB consists of algorithms to implement discrete-time dynamic systems, along with an Execution Control Chart (ECC) that specifies the behavior of an automaton to coordinate the execution of the said algorithms.

As such, any tool designed to address the digital twinning of CPSs must effectively support the coexistence of DAE and DEVS components in the same model.

3.4 Scalable detail and efficiency

Simulating DAE and DEVS models jointly presents two interconnected challenges, which, in the authors' opinion, currently represent the primary obstacles to achieving an integrated approach for the digital twinning of CPSs. To illustrate these challenges, consider a modulating controller such as a PID. The simplest representation of this controller is given by differential equations, which can be integrated at simulation time using variable-step solvers for fast execution. However, such a solution method does not precisely capture the behaviour of the real controller,

which is implemented as an algorithm, as mentioned earlier. A more accurate representation would be a discrete-time dynamic system, where state and output are updated (periodically) during simulation. An even more detailed representation could involve a full *replica* of the controller code, which would be invoked by the simulation tool (technical details are beyond the scope here) when necessary. A first problem in such a context is that finer-detail representations have more parameters and variables than coarser-detail ones, which complicates the task of maintaining consistency within the model base (most notably, among differently detailed representations of the same object). But there is a second, more significant, issue: detailed representations require interrupting the integration of the DAE system whenever an event occurs, and this can severely impact simulation speed (Cimino et al., 2024)

As such, any tool designed for the digital twinning of CPSs must jointly support equation- and algorithm-based modelling. It must also enable the creation of standardised model interfaces to facilitate component interchangeability and model-to-model consistency checks. Finally, and most importantly, it must manage simulation events efficiently, which requires to decide when an event is "significant" (for example, most of the periodic calls to the code of a modulating controller would likely generate so little an output variation that if the simulation just skips them, holding the previous output, the effect is negligible).

3.5 Dimensionality

All the issues mentioned so far are exacerbated by the size of the models involved, which at plant-wide level can easily contain hundreds of thousands – sometimes millions – of equations and variables. This makes scalability a crucial property for any solution algorithm and tool. Furthermore, in the engineering process, models are often modified between simulation runs. Assuming that the simulation tool is a compiler (not an interpreter) for efficiency reasons, the scalability issue applies not only to the generated code but also to the process of generating that code.

As such, any tool for the digital twinning of CPSs must not only provide fast simulation but also enable rapid compilation, even when handling large-scale models.

4. THE PROPOSED STRATEGY AND ROADMAP

We have seen that as for the integration of models and simulations, various methodologies are being explored. Among these, an important role is covered by the Object-Oriented Modeling (OOM) methodology for DAE-based systems (Cimino et al., 2021), as well as by semantic relationships to ensure consistency across models (Jinzhi et al., 2022). We have also seen that when handling a CPS/CPPS at all the mentioned levels, SoS included if necessary, the possibility of continuously updating the models is essential, as knowledge on the system evolves throughout its lifecycle. Currently, the mainstream practice is to privilege DAE models for component and system engineering,i.e. in the design phase, while tasks nearer to the job shop floor, such as capacity planning, online scheduling and so forth, are better handled in a DEVS setting, to which DAEs can just provide off-line information — and when the two kinds of models need to run together, the common view is that the only possibility is resorting to framework-specific tools coupled via co-simulation. Hence, ultimately, the only reason for which such a setting is not (yet) under developement is that current M&S tools cannot (yet) provide DAE/DEVS integration with the efficiency required for industrial applications. The advantages of that integration would be numerous and open several new perspectives, at present not explored just because the available technology offers inadequate support i.e., could provide a way to respond to the challenges identified herein referred to the use of M&S tools in manufacturing.

To exemplify in the available space, consider three hypothetical *scenarii*. First, it is common to create several DEVS models for different analyses (e.g., different fault settings) while one or a few DAEs, suitably representing the said settings by means of parameters with a clear phisical interpretation, would suffice. Second, and somehow analogous, describing machine (FSM and DEVS) states often incurs a combinatorial explosion, while having a DAE represent the underlying physics would prune the model base. Third – as we show in the example later on – sometimes DEVS models require probability distributions that depend on physical phenomena (controls included) in such a way to make the inclusion of a DAE the best way to represent them.

The proposed strategy is to adopt Object-Oriented Modelling (OOM, and in particular the Modelica language) extensively as it offers native support for multi-physics and mixed equation-algorithm modelling, as well as openness to co-simulation (which sometimes could still be convenient in the end) via the Functional Mockup Interface standard. Putting this idea to work requires an articulated roadmap, described in the following.

Structured abstraction. It is necessary to abstract model interfaces and structures with the DAE/DEVS coexistence in mind, (Figure 2), because, when controls enter the arena, changing the detail level could mean modifying model *structures* — e.g., replacing a transfer function (DAE) with an algorithm (DEVS).

Efficient event handling. It is also necessary to manage events efficiently, (i) grouping those from periodic sources — for example, anything within a PLC cycle is sensed at the beginning of the next one, see Cimino et al. (2024), and (ii) allowing the integration of the continuous time system to "disregard" events that do not jeopardise the correctness of the solution if not handled precisely in time — a hardly explored matter.

Efficient simulation. Finally, besides writing efficient models, it is necessary to make their *compilation* efficient. This means designing and developing high-performance (Modelica) compiling tools to cope with models of the size one encounters when the DAE/DEVS integration challenge is taken. Examples are proposed in Casella (2015) and Agosta et al. (2023).

5. EXAMPLE

We provide an example about the necessity of resorting to joint DAE/DEVS modelling that are based on two different *structured abstraction*, as per their definition trhough

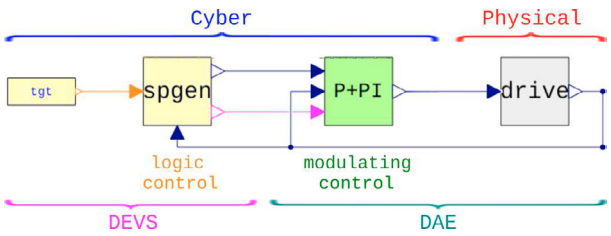

Figure 2. Modelica scheme for the example model with its Cyber-Physical and DEVS-DAE partitions.

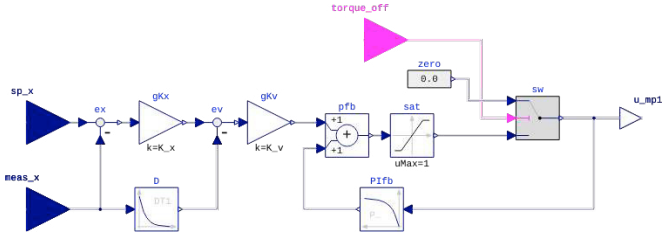

Figure 3. Modulating control in the DAE part of the example model.

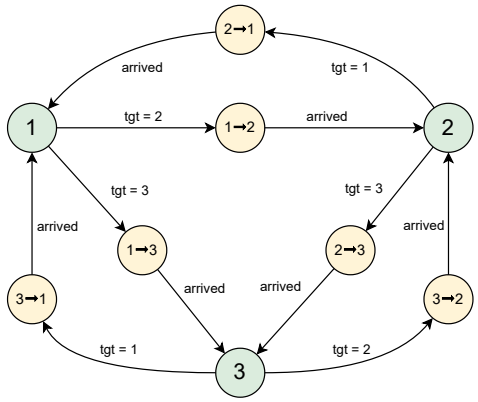

Figure 4. Automaton for the DEVS part (logic control) of the example model.

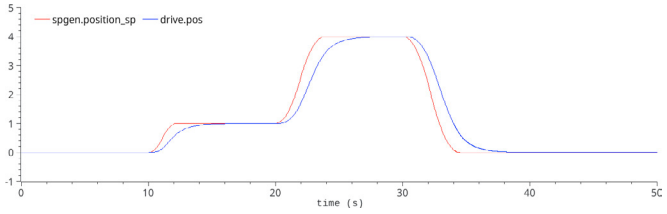

Figure 5. Example of the controlled system behaviour: position set point from the generator (`spgen` in Figure 2) and measured position from the drive.

the roadmap. We consider a linear positioner with three stations, that needs to move parts of variable mass and form factor, which results in a variability also of the observed friction. The positioner is operated by a drive endowed with position control, the DAE *structured abstraction* is realised with the widely used cascade structure with a velocity PI and a position P controller. Figure 2 shows the complete simulation model, while Figures 3 and 4 respectively report the modulating and logic control schemes. We assume that the system must perform a repetitive operation, moving among stations 1–3 (locates respectively at 0, 1m and 4m in the single positioner linear coordinated) as shown in the example behaviour in Figure 5; not completing the operation within 70s is considered a failure.

We want to investigate how the DEVS *structured abstraction* rely on the probability distribution of the completion time t_{fin} for the considered operation, that is turn influenced by the settling time t_{set} used to tune the modulating position control, to be set in the DAE *structured abstraction*. The goal of this study is to determine whether or not the DAE *structured abstraction* in Figure 2 can be used to parametrise the probability distribution of interest of the DEVS *structured abstraction* with respect to t_{set}. If this is possible, then one could employ the "complex" DAE/DEVS positioner model only offline for the said parametrisation; online simulations of the entire system that contains the positioner could just use the obtained probability distribution, avoiding the DAE/DEVS model to the advantage of computation speed. To this end we perform 2000 simulations, with part mass and frictions extracted from the same distribution, for each value of interest for parameter t_{set}. We then measure the resulting completion times, and perform a Gaussian kernel probability density estimation.

The resulting estimated probability distributions of t_{fin} as a function of t_{set} are shown in Figure 6. As can be seen, these distribution exhibit moments (most notably, the average) that vary in a nontrivial manner with respect to t_{set}. But most important, differences are *qualitative*: in some cases we have an evident bimodality, in others a peak and a long thin tail, in others a quasi-flat aspect. Indeed, providing a reliable description of the modelled system by not considering its DAE/DEVS nature, or even relegating this nature to offline, component-level simulations only, is at least highly questionable.

Overall, referring to Figure 1, the example show a possible integration based on OOM models of the DAE/DEVS abstraction levels that correspond to having DTs with consistent simulations at the level of CPPS and CPS. When considering a more realistic case, a system might have many different FSM/DEVS models used for different purposes, i.e. monitoring, maintenance, fault detection (other what-if analyses), and all of them should be consistent among each other and with their corresponding DAE model(s). With such a possible complex chain of simulation(s), it is fundamental to rely on the roadmap proposed using efficient simulation.

6. CONCLUSIONS AND FUTURE WORK

We explored in more detail the whole CPS-based context and the challenges related to the application of such concept in order to reach the feasibility of an integrated approach. In particular, we focused on the DAE/DEVS *structured abstraction* integration, emphasising the critical need to efficiently handle both model types jointly — a point where current M&S tools, while powerful, fall short. We proposed a strategy based on OOM, highlighting the difficulties in handling events and in particular control algorithms, owing to their imperative nature that is often at odds with the declarative one of OOM. Based on that proposal we sketched out a possible roadmap, including

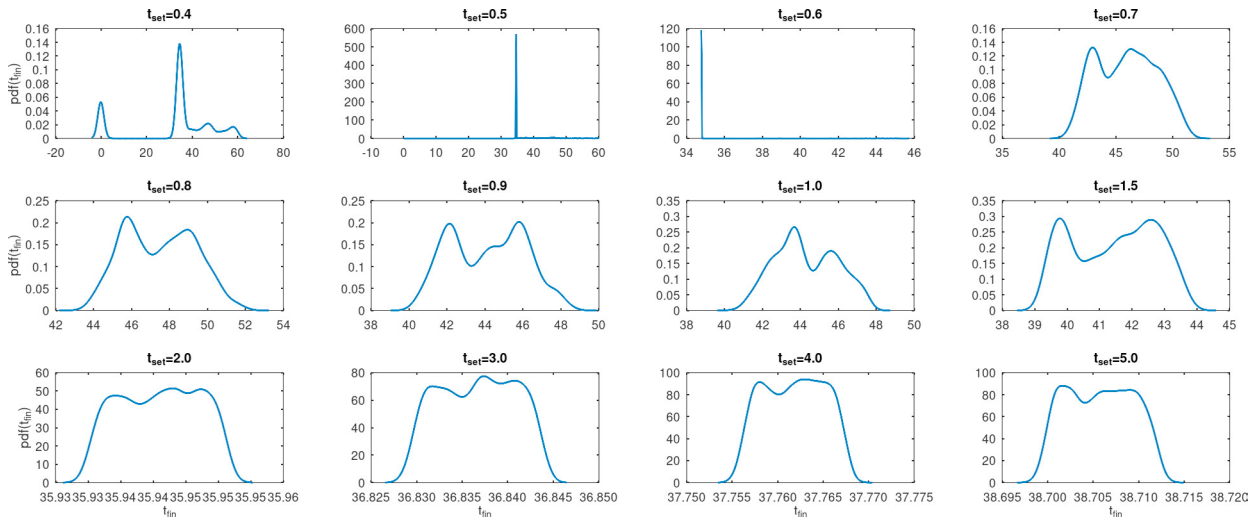

Figure 6. Estimated probability distributions for the completion time t_{fin} of the operation in Figure 5 in the face of varying mass and friction, for different values of the required position settling time t_{set}; the peak around zero in the first plot is an estimation artefact, $t_{fin} = 0$ means operation not completed within 70s.

advances in compilation tools to handle the heterogeneous and large-scale models that can be encountered.

While the path to effective integration presents challenges, the potential benefits in terms of simulation accuracy, efficiency, and system insight make it a worthwhile endeavor for advancing industrial modelling and simulation capabilities. Future work will thus focus on such developments, and on the evaluation of manufacturing *scenarii* where it is possible to create policies – based on the sketched roadmap – for the automatic (or at least assisted) handling of the different detail levels required in the CPS/CPPS/SoS context.

REFERENCES

Agosta, G., Casella, F., Cattaneo, D., Cherubin, S., Leva, A., Scuttari, M., and Terraneo, F. (2023). MARCO: an experimental high-performance compiler for large-scale Modelica models. In *Modelica Conferences*, 13–22.

Aydt, H., Turner, S.J., Cai, W., Low, M.Y.H., Lendermann, P., Gan, B.P., and Ayani, R. (2008). Preventive what-if analysis in symbiotic simulation. In *2008 Winter Simulation Conference*, 750–758. IEEE.

Casella, F. (2015). Simulation of large-scale models in modelica: State of the art and future perspectives. In *Linköping electronic conference proceedings*, 459–468.

Cimino, C., Ferretti, G., and Leva, A. (2021). Harmonising and integrating the digital twins multiverse: A paradigm and a toolset proposal. *Computers in Industry*, 132, 103501.

Cimino, C., Terraneo, F., Ferretti, G., and Leva, A. (2024). Scalable and efficient digital twins for model-based design of cyber-physical systems. *International Journal of Computer Integrated Manufacturing*, 37(10-11), 1232–1251.

Graessler, I., Wiechel, D., and Rarbach, S. (2024). Model-based impact analysis in dynamic system of systems. *Procedia CIRP*, 128, 585–590.

International Electrotechnical Commission (2013). IEC 61131-3 Programmable controllers – part 3: programming languages, edition 3.0.

Jinzhi, L., Zhaorui, Y., Xiaochen, Z., Jian, W., and Dimitris, K. (2022). Exploring the concept of cognitive digital twin from model-based systems engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 121(9), 5835–5854.

Karabulut, E., Pileggi, S.F., Groth, P., and Degeler, V. (2023). Ontologies in digital twins: A systematic literature review. *Future Generation Computer Systems*.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IfacPapersOnline*, 51(11), 1016–1022.

Magnanini, M.C. and Tolio, T. (2020). Switching-and hedging-point policy for preventive maintenance with degrading machines: application to a two-machine line. *Flexible Services and Manufacturing Journal*, 32, 241–271.

Monostori, L. (2015). Cyber-physical production systems: roots from manufacturing science and technology. *at-Automatisierungstechnik*, 63(10), 766–776.

Negri, E. and Abdel-Aty, T.A. (2023). Clarifying concepts of metaverse, digital twin, digital thread and aas for cps-based production systems. *IFAC-PapersOnLine*, 56(2), 6351–6357.

Pang, T.Y., Pelaez Restrepo, J.D., Cheng, C.T., Yasin, A., Lim, H., and Miletic, M. (2021). Developing a digital twin and digital thread framework for an 'industry 4.0'shipyard. *Applied Sciences*, 11(3), 1097.

Rosen, K.M. and Pattipati, K.R. (2023). Operating digital twins within an enterprise process. In *The Digital Twin*, 599–659. Springer.

Wainer, G.A. and Mosterman, P.J. (2018). *Discrete-event modeling and simulation: theory and applications*. CRC press.

Zahid, M., Bucaioni, A., and Flammini, F. (2024). Model-based trustworthiness evaluation of autonomous cyber-physical production systems: A systematic mapping study. *ACM Computing Surveys*, 56(6), 1–28.

Zeigler, B.P., Praehofer, H., and Kim, T.G. (2000). *Theory of modeling and simulation*. Academic press.