Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-2 (2022) 7–12

On consistency of physical and DEVS models in control-targeted DTs: an industrial case study

Chiara Cimino[1], Cristian Granata[2], Gianni Ferretti[3] and Alberto Leva[3]

***Abstract*— The DT-based design and control of modern production assets requires simulation models with different viewpoints, purposes, and nature. Most notably, component-level design and control require detailed physical models, while plant-level automation and production management mostly need DEVS-type ones. Inconsistencies among the said models can have significant impacts on decision processes related e.g. to (re-)configuration, diagnostics and maintenance—in one word, on the ultimate outcome of an asset. However, given their heterogeneous nature, making physical and DEVS models consistent requires new methods and tools, in fact starting with the definition itself of what "consistent" has to mean. We here present a methodology to state and address such consistency problems, and support our statements with a case study taken from the cosmetics industry.**

I. Introduction

Modern industrial assets increasingly require lifelong support from "Digital Twins" (DTs for short) to assist their design, operation, maintenance and management at large. There are strong and articulated motivations for this necessity, as well as a huge literature and many applications, e.g. to [11], [5], [12], [17].

The said necessity is strengthened by the contemporary industrial *scenario*. In the past, a production asset traversed quite well defined phases – design, commissioning, operations with some re-configuration, end of life – on a time scale in the range of years or more. Such phases were in practice cascaded, hence it made sense e.g. for a design team to "pass over" the asset to operations people. It was of course required to define interfaces for team-to-team information transfer, but *internally* – this is a key point – each team could organise and manage their knowledge in almost complete independence.

Nowadays, the situation is different. The need for frequent re-configurations, that once was confined to high-mix & low-volume productions, is rapidly spreading into a market that requires customised products practically as fast as standard ones. The field of cosmetics, that funded this research, is an apparent example, and less peculiar than one might imagine. The above cascade of phases is far less representative of reality than it was, and to top, the time scales of asset management, maintenance and day-by-ay operation, which once were very different, are approaching to one another.

Each DT created – see [5] – is usually employed for specific decisions on an asset, starting from commissioning ([16], [15]) to embrace its entire life cycle. A DT can also have different natures (it could come from field data, models, or both), usage (online, offline or a combination of the two) and different interpretations for each professional [3]. For example, a control engineer, a production planner and a data scientist can mean for "DT" respectively a simulation model made of differential equations, a Discrete EVent System (DEVS) and a decision aid built on machine learning.

To not take consistent decisions trhough an asset life cycle – so that nobody ever takes a decision with his DT assuming true any other decision that somebody else has modified on its DT – it is of major importance to have an integrated approach to build consistent DTs, regardless of their nature, usage and purposes. The matter, discussed in [14], [18], [4], [13]m gives the subject of "models in control" new facets, both methodological and technological [19]. A proposal of DTs integration is the topic of the long-term research to which this paper belongs [3], with the ultimate goal of devising methods and tools to manage a knowledge base made of both models and data sets, accessed by professionals with different cultures. In this paper, we investigate how to ensure consistency – in the sense just suggested – among detailed physical models and DEVS ones, so as to integrate the automation and control (A&C) decision strategies with the operations and management (O&M) ones. We also exemplify our findings on an industrial case, to prove their practical viability.

II. Background and problem statement

As we said, a knowledge base made of DTs contains both models (in our context, entities that can be simulated) and data. Apparently, any idea of "consistency" in such a base has to be expressed as *relationships* among the entities in the base (models and data). However, while the concept of "relationship" among data is well known since the dawn of the relational model [6], [7], [1], that of "model-data" or "model-model" relationship is far less established, if not in fact to establish at all.

The matter is discussed – and a solution outline provided – in [3]. We here report just an illustrative example, tailored to the particular problem addressed in this paper. Consider a hierarchical control system in a manufacturing line, with machine-level controls (mostly of modulating type like e.g. speed or pressure ones) and line-level ones (mostly logic like e.g. sequencing ones) on top of them. Machine-level controls rely on DTs based on dynamic models made e.g. of transfer functions, the parameters of which represent the physics of the machine (masses, friction coefficients, an so on) and the tuning of the controllers. Line-level controls rely

[1]C. Cimino is a research assistant at the Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Italy, chiara.cimino@polimi.it

[2]C. Granata is with Omnicos Group srl, Bagnolo Cremasco, Italy, c.granata@omnicos.it

[2]G. Ferretti and A. Leva are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy, {gianni.ferretti,alberto.leva}@polimi.it

10.1016/j.ifacol.2022.04.161

on DEVS-based DTs where the details of machine operations are concealed, but the parameters of which depend on how machines are controlled. For example, a line-level control functionality could rely on a machine-level one to guarantee (via tool motion and force control) that some machining time lie in a certain range despite part-to-part variability in say the material hardness.

Apparently, for the overall system to work properly, any modification in the machine-level control must reflect into its DT, and this has to trigger the update of all line-level DTs that depend on that machine-level one, as in the absence of that, decisions at the line level may be erratic. Setting up a mechanism to manage a set of DTs in such a way requires in the first place to define what a DT-DT (model-data and/or model-model) relationship is, which was initially done in [3] by identifying a set of such relationships.

In this paper we refer to one such relationship, named "representation in other domain", that occurs when the same entity is modelled adopting different paradigms. For example, an inductor can be modelled in the time domain with a differential equation having tits inductance as the one parameter, or in the phasor domain with an algebraic equation, where the parameters are inductance and network frequency. In such a case, relating the two models is quite easy. But coming back to the case sketched above, a machining station can be modelled as a differential-algebraic system, with physical parameters, or a queue-type DEVS, where the parameters characterise the statistical distribution of the machining time. In the following we study this particular example of DT-DT consistency problem, based on a real-world case, and exploit it to make some general considerations.

III. THE ADDRESSED RELATIONSHIP

The specific kind of relationship that we consider, as said, is termaed "representation in other domain" and in a nutshell amounts to re-writing the same physical object model with different dynamic descriptions and interfaces. In particular, relating the A&C physical models with the O&M ones means to model the same machine as a queue/server block [2], following the Discrete EVent Systems (DEVS) modelling.

A very important peculiarity of establiching this relationship – in fact, the main reason why it is introduced – is its usefulness for turning *a priori* component-level uncertainty into *a posteriori* machine- or in general system-level uncertainty. Let us explain, for brevity, with an example. When using a queue and server model for a machine in order to analyse the production line that contains it, it is obviously required to characterise the variability of the output of that machine, be this in terms of service time for each processed part, of characteristics of the output parts, or whatever else. Such a characterisation may come from analysing measured data along the production history, and possibly – but for many kinds of analysis not necessarily – attempting to relate those data to some machine input, measured historically as well.

The problem is however when the plant is being built, hence there is no data, and to set up the management logic some information about the mentioned variability is required anyway. One could make assumptions based on experience, but this is not always available (especially if the plant is innovative). One could start with "cautious" settings (assuming to know what "cautious" is in the case at hand, but we are not discussing this) and then refine controls along the plant operation, but as we said, nowadays there might not be enough time for that before the next re-configuration occurs. In one word, the outcome of such cases can easily be suboptimal operation, if not undesired behaviours.

The solution we propose is to exploit the introduced relationship, and reason as follows, starting again with a simple example. Suppose to consider a machining operation, where the variability of interest is that of the machining time. First, a detailed model-based DT needs creating that explicitly contains the *physical* quantities that the knowledge of engineers identifies as responsible for the machining time variability *with their intrinsic variability* (for example, the part hardness). Since this intrinsic (*a priori*) variability can be characterised before the machine exists, a convenient simulation campaign can provide a reliable estimate of the *a posteriori* variability required by the "simple" model-based DT made using the DEVS modelling.

Formally, this means relating the simple to the complex models by expressing some parameters of the former (in the example, a specified number of moments of the machining time distribution) as functions of some parameters of the latter (for example the electro-mechanical and the control ones) plus data about the parts to machine (in the example, moments of the hardness distribution). The functional dependence just introduced shall be expressed in terms of relationship by means of a suitably specified set of simulations of the complex one in order to compute the simple model parameters, followed by the necessary elaboration of the output. This can clearly be done off-line with respect to the simulations that use the simple model.

No doubt that having such a relationship specified once and then enforced automatically, allows the analysts who deal with the asset-level models that contain the machine as a queue with server to be sure that no modification of e.g. a control parameter that somebody in charge of the machine might apply for local reasons, will make their conclusions not correspond to the real situation. No doubt therefore also that such a functionality is of value for the management of model-based DTs from different domains. The question is now whether or not the idea just outlined can be applied to a real-life situation. We provide a response with the case study in the following, after which we are making some further considerations suggested by the described experience.

IV. THE CASE STUDY

We now come to apply the ideas just sketched to an industrial case, namely lipstick solidification. This process takes place on a rotary table where silicone moulds receive the hot and fluid product, traverse a cooling tunnel in which chilled air flows in the opposite direction, then reach a position where they are extracted to collect the finished now solidified product; at this point each emptied stand on the table receives another mould, that goes through a pre- heating section before receiving another fluid injection and restarting the cycle.

The available knobs to act on the system are the table average rotation speed (the motion is actually discontinuous, as moulds have to stop at filling and emptying stations) together with the air flow rate and its conditions at the cooling tunnel inlet, in turn governed by a chiller coupled to the solidification machine.

The goal of the process is to have each lipstick follow a desired cooling curve over time, so as to obtain good consistence and finishing and be finally extracted from the mould smoothly, avoiding in particular defects like a non homogeneous aspect or even localised cracks. At the same time, the total energy consumption by the table motor, the mould pre-heating system (radiating lamps) and the chiller, should be kept as low as possible compatibly with a satisfactory operation.

A. Detailed physical model

The physical (first principle) model was written using the object-oriented Modelica language [9]. We now synthetically describe its main components.

1) Rotary table, mould and bulk: the compound of silicone mould and bulk is represented as a finite-volume model with cylindrical symmetry, discretised by horizontal slices. The part is thus modelled as a set of concentric cylinders, see Figure 1: the outer hollow one represent the metal support, the middle one the mould, and the inner one the bulk.

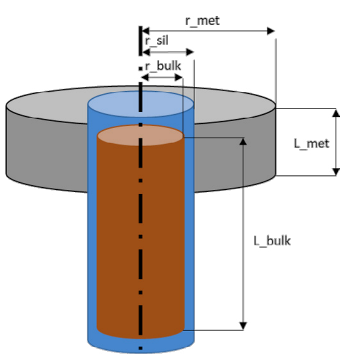

Fig. 1. Mould and bulk model.

The density, specific heat and thermal conductivity for table metal, silicone and bulk are considered constant. The model is thus composed of one bulk, one mould and one metal energy equation per horizontal slice. Filling is modelled as if the flow – that actually fills the slices from bottom to top – were instead distributed to all the slices together, and the same is done for emptying. Since filling and emptying are practically instantaneous with respect to the entire process duration, this non-physical approximation is acceptable. The rotary table is divided in sectors – see Figure 2 – based on the sequence of operations summarised below, that in turn dictate the mass and heat exchanges for the model in Figure 1.

2) Table zones and operations: With reference to Figure 3, incoming moulds initially exchange heat only with the table support (ΔQ_{p_sil}) and room air (ΔQ_{T_sil}, ΔQ_{T_p}). During preheating, additional heat to mould and table (ΔQ_{irr_sil}, ΔQ_{irr_p}) is provided by lamps. Then comes the prescribed inlet flow rate w_{bulk} in the filling position, introducing the heat exchanges between bulk and mould (ΔQ_{T_bs}), and between bulk and air (ΔQ_{T_bulk}).

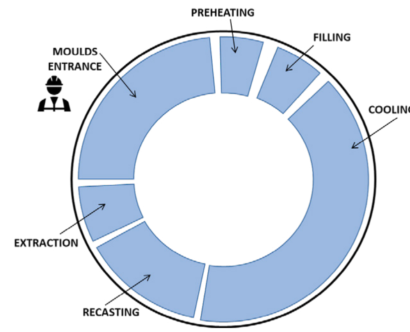

Fig. 2. Rotary table and working steps.

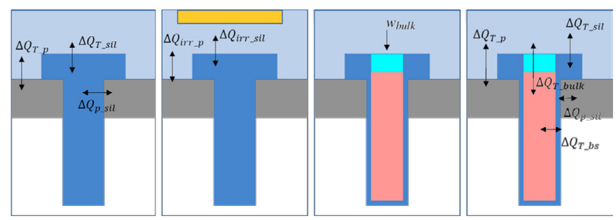

Fig. 3. Entering, preheating, filling and exchanges (left to right).

After filling, moulds enter the cooling tunnel. There is no mass exchange, whole heat ones are the same as before. The only differences are the air flow rate – i.e., velocity, temperature and moisture. The tunnel is modelled as a sequence of volumes, where dynamic balances of mass and energy apply, alternated with transport elements in the form of an algebraic correlation between flowrate and pressure difference. At the tunnel exit, the bulk is finally solidified, however it is slightly recast only on the upper surface, to be extracted and positioned into its primary packaging. Again, exchanges are the same, with the only addition of lamp heat in the recast positions and of the prescribed outgoing mass flowrate at the emptying station.

3) Model elements interconnection: Each mould/bulk model is equipped with a set of ports – in Modelica terms, *connectors* – to represent the mass and heat exchanges that may or may not exist depending on the station where the model is. For example, ports to represent thermal exchanges without mass carry a temperature and a thermal power.

Denoting by n_P the number of ports aboard a mould/bulk compound and by n_C the number of compounds on the table, this results in n_P vectors of ports of dimension n_C each. Analogously, each of the n_S stations on the table is equipped with the same set of ports. If some phenomenon does not occur at a certain station, this is represented by prescribing zero heat or mass flow on the corresponding port. The result is thus n_P vectors of ports of dimension n_S each.

The interconnection is realised by defining, at the complete model level, n_P matrices of size $n_C \times n_S$, to dictate the compound-to-stations interconnections as a function of the table angular position. Overall, with 40 table stations and thus 40 table stands, four mould/bulk compounds per stand, and 24 volumes – one for each working position – to discretise the cooling tunnel, the complete physical dynamic model is composed of about 6000 equations.

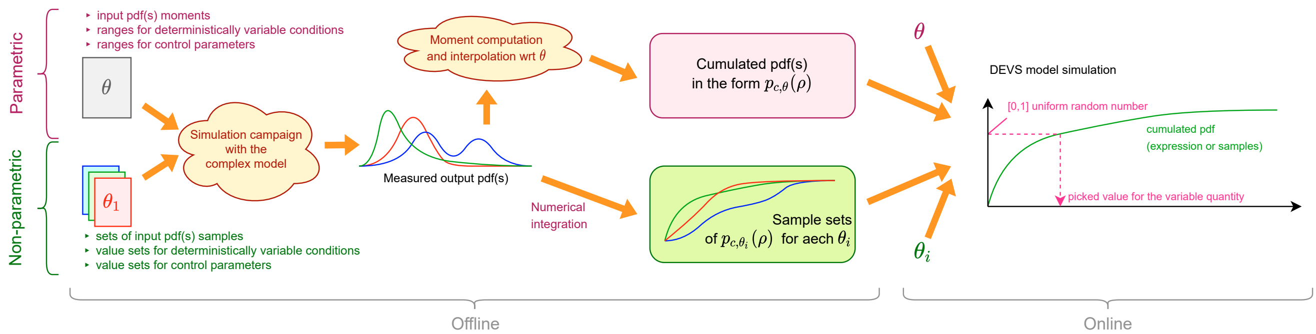

Fig. 4. Operations flow – automatable – to relate the complex (physical) and the DEVS model along the parametric and the non-parametric approach.

B. DEVS model

The DEVS model takes the form of a queue and server node for use in a network to represent the production flow. The job token corresponds to a single lipstick (or – equivalently for our purposes – to a set of four if moulds come in quadruplets handled together). Tokens flowing through the network carry information about the represented product unit(s) to be used as input to the downstream processing nodes if this is relevant.

In the application we consider, the processing time for a token is dictated by the table speed and generally kept constant. The production system contains adequate upstream and downstream buffers to make this acceptable. What counts in particular is the product condition at the exit of the cooling tunnel, as this affects the following operations besides having an obvious impact on energy consumption.

In the space of this paper we limit the scope to the product temperature, but apparently the same considerations would apply to other properties computable from the simulation of the detailed model. For the DEVS one, it is required to determine the cumulated probability density $p_{c,\theta}(\rho)$ for the property ρ of interest as a function of the qualifiers set θ for the inlet/ambient variability and the relevant process/control parameters, so that the arrival of a token to process can produce its simulated outlet condition as $p_{c,\theta}^{-1}(\varepsilon)$, where ε is a [0,1] uniformly distributed random number.

C. Establishing the relationship

In principle, one could tackle the problem with a parametric approach. This would amount to computing the required number of moments $M_{O,i,j}$ for each of the output probability distrbutions of interest $p_{O,i}(\cdot)$, $i = 1 \dots n_O$, $j = 1 \dots n_{M_O,i}$ as functions of the required number of moments $M_{I,k,\ell}$ for each of the input distributions $p_{I,k}(\cdot)$, $k = 1 \dots n_I$, $\ell = 1 \dots n_{M_I,k}$ identified as the physical sources for the output variability; these input moments would contribute to the qualifier vector θ together with deterministic quantities, most typically the parameters for the controls aboard the system.

Such a procedure is conceptually neat, would be in practice quite articulated but in general straightforward to set up, and above all would build on solid methodological result like those presented in [10], [8]. However, given the particular domain we are addressing, there is a quite strong argument owing to which the idea above may often not be advisable, and this argument comes from the combination of two facts.

First, some of the output properties to address depend on deterministic parameters in an inherently non-smooth manner. The most immediate example is the way a settling time (that can dictate when an operation is considered finished, hence a service time an a queue network) depends on the threshold for the transient end, and also on the control parameters (when an oscillation arises as wide as the said threshold, the settling time has a jump). As such, changing e.g. a control parameter could result in a *qualitative* modification of the *shape* of some output property distribution, making it questionable – if ever possible – to decide *a priori* how many moments are necessary to represent it properly.

Second, sticking again to the DEVS/queue case, as it may happen that the input variability for some node in a queue network is actually the output variability of some other node, the same difficulty about the necessary number of moments applies to input distributions as well. Of course this is not the case if for example the input variability comes from a population of acquired parts that can be characterised offline, but we aim for a procedure that can be applied in the most general case.

As a result, the two approaches can be summarised as per the operations flow in Figure 4 the parametric approach can work only if one has certainties about the number of moments that suffice to describe the distributions to include in the models. In the opposite case one has to take a non-parametric approach, and determine the required output distributions as a set of values to be interpolated at the time of simulating the DEVS model, as sketched above. This amounts to

1) identifying the deterministic parameter(s) – θ_D to name them – and defining the set Θ_d of the choices to consider for them (e.g., the control parametrisations of interest);
2) characterising the input variability, either as moments if this makes sense, or in general as samples of its cumulated probability density;
3) performing a large set of simulations for each $\theta_D \in \Theta_D$ and compute numerically the corresponding samples for the output cumulated probability densities of interest.

As the price for its generality, this approach has the drawback that if something is changed in the model of some queue network node, all of the above needs re-doing for all the nodes who take its output—which apparently would not be the case if probability-related information could move around the network as a pre-defined set of moments. The question is therefore whether the proposed non-parametric approach is feasible in practice, and to this question we attempt to respond by applying the approach to the lipstick drying case.

V. Simulation results

We now present an inevitably small example with the lipstick drying process. Assuming "good" control for table speed and air conditions and flowrate, the main *a priori* uncertainty resides in the fluid bulk inlet temperature T_i. For this example we assume a Gaussaian distribution of the said temperature, with average μ_{T_i} and standard deviation σ_{T_i}. The corresponding *a posteriori* uncertainty of interest is that of the final temperature T_o of the solidified product. We took two values for the table rotation speed, five for μ_{T_i}, five for σ_{T_i}, and only one air flow and inlet condition for brevity. For each table speed – and with the so defined operating points – we ran 5000 simulations to analyse the *a posteriori* uncertainty. We report below the outcome of using both the approaches outlined in Figure 4.

A. Parametric approach

Along this approach one assumes that the *a posteriori* uncertainty can be described with a given number of moments, and computes these based on the distribution(s) coming from the simulation set.

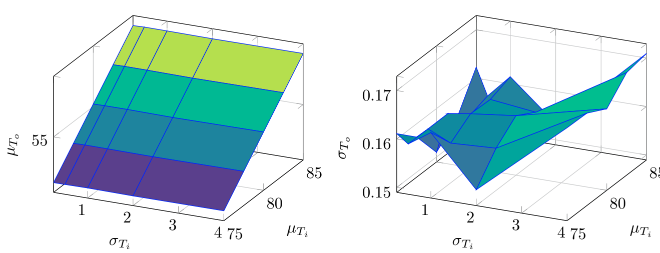

Fig. 5. Parametric approach – *a posteriori* uncertainty moments μ_{T_o} and σ_{T_o} versus μ_{T_i} and σ_{T_i} – all in °C – for ω_T corresponding to 1000 parts per hour (pph).

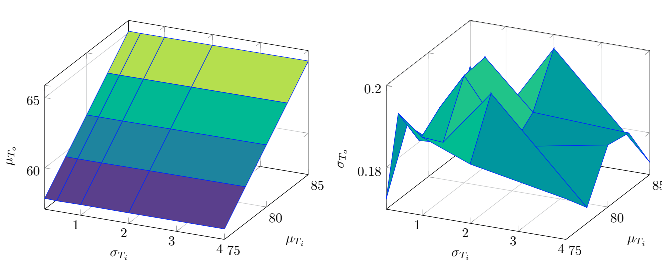

Fig. 6. Parametric approach – *a posteriori* uncertainty moments μ_{T_o} and σ_{T_o} versus μ_{T_i} and σ_{T_i} – all in °C – for ω_T corresponding to 1400 parts per hour (pph).

Figures 5 and 6 show the results in the case at hand, assuming that average and standard deviation suffice to describe the *a posteriori* uncertainty. Specifically, the average μ_{T_o} and the standard deviation σ_{T_o} of T_o are plotted versus those of T_i for two values of the table velocity ω_T (one per figure). As can be seen there is an evident correlation for μ_{T_o}, while on σ_{T_o} finite-population effects do appear (indicating however a larger average value for 1400 parts per hour). In any case, assuming the used moments to be an adequate representation of the *a posteriori* uncertainty – which depends on the use one has to make of the DEVS model, a matter not discussed herein – interpolating also between the operating points coming from the simulation appears possible.

B. Nonparametric approach

Along this approach one just considers the simulated operating points – hence no interpolation is envisaged – and makes no assumption on the aspect of the *a posteriori* uncertainty distribution(s), providing the required cumulated probability density/ies as a set of samples to be interpolated when employing it as per Figure 4 (right).

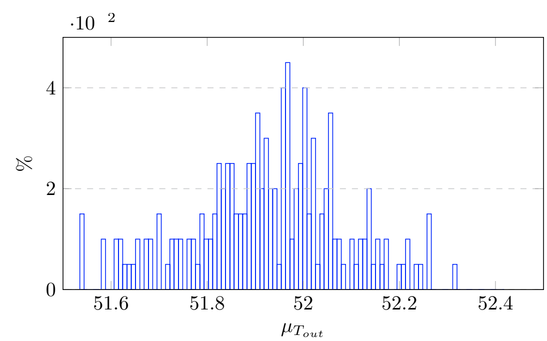

Fig. 7. Probability density function for 1000 pph, $\mu_{T_i} = 75°C$ and $\sigma_{T_i} = 0.25$

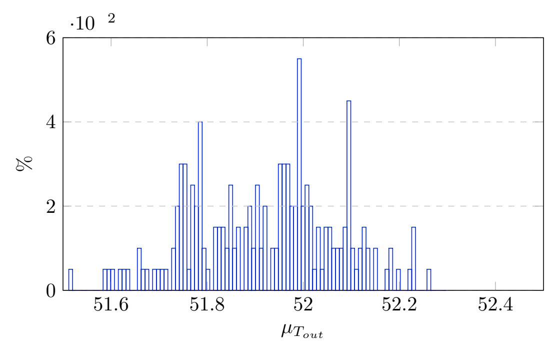

Fig. 8. Probability density function for 1000 pph, $\mu_{T_i} = 75°C$ and $\sigma_{T_i} = 1$

Figures 7 and 8 show two such probability density functions (not cumulated for better readability). As can be seen comparing the figures, in some cases the aspect is "Gaussian enough" – or more in general regular enough – for being represented with a reasonably low number of moments, while in some other cases this is more questionable.

C. Lessons learnt

From the presented study (and others not shown here) we can distil some lessons. First, the parametric approach is in fact feasible, however only it at least the shape of the probability density functions to expect is reliably known. Symmetrically, the nonparametric approach is inherently agnostic to such assumptions but not so rich as for the yielded information, as any attempt to somehow interpolate in between the computed operating points did prove infeasible in practice. We also detected a significant influence of control parameters in changing distribution shapes (though this happens primarily for threshold-based quantities such as settling times, not shown in the presented study). Hence the nonparametric approach appears in general safer for quantitative evaluations, while the parametric one seems more suited to preliminary design activities and/or relative comparisons. Finally, the time to perform a set of simulations can vary a lot, from minutes in simple cases up to hours in complex ones. We do not see this as a significant obstacles, also because in our runs we did not exploit at all the apparently high level of parallelism encountered, but nonetheless the computational aspect still needs attention.

VI. Conclusions and future work

We presented an industrial case study to demonstrate the feasibility of instating relationships among model-based control targeted DTs from different viewpoints, within a knowledge base made of both models and data. The reported simulation results show the viability of the approach in evaluating the DT-DT relationships without extensive use of the physical counterpart data.

It is important to note that identifying DT-DT relationships requires a major analyst effort in dealing with the other professional viewpoints, defining what "consistency" means, and devising the right path to follow to establish those relationships—i.e., to determine the crucial uncertainty to be represented, that in our case study means choosing the knobs to act on the system. Nevertheless, enforcing those relationships is crucial for the integration and the lifelong management of the whole phases of an industrial asset's life cycle, i.e. engineering, commissioning, control and operations, from different viewpoints.

Future work will focus on continuing toward the specification of other DT-DT relationships, to be integrated in the future into DT-centric industrial tools. Also, research effort is being spent, in collaboration with compiler developers, to have relationships automatically enforced in a computationally efficient way.

Acknowledgements

Support for this work came from the Lombardia regional research and innovation project AD-COM, "ADvanced COsmetic Manufacturing" (ID 214632).

References

[1] P.P.S Chen. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[2] C. Cimino, G. Ferretti, and A. Leva. The role of dynamics in digital twins and its problem-tailored representation. *Proceedings 21st IFAC World Congress, Berlin 2020*.

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

[4] C. Cimino, A. Leva, E. Negri, and M. Macchi. An integrated simulation paradigm for lifecycle-covering maintenance in the industry 4.0 context. *IFAC-PapersOnLine*, 53(1):307–312, 2020.

[5] C. Cimino, E. Negri, and L. Fumagalli. Review of digital twin applications in manufacturing. *Computers in Industry*, 113(11):103130, 2019.

[6] E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[7] E.F. Codd. Further normalization of the data base relational model. In R. Rustin, editor, *Data base systems*, pages 33–64. Prentice-Hall, Englewood Cliffs, NJ, UA, 1972.

[8] C. Dang and J. Xu. Novel algorithm for reconstruction of a distribution by fitting its first-four statistical moments. *Applied Mathematical Modelling*, 71:505–524, 2019.

[9] P. Fritzson. *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, Hoboken, NJ, USA, 2014.

[10] V. John, I. Angelov, A.A. Öncül, and D. Thévenin. Techniques for the reconstruction of a distribution from a finite number of its moments. *Chemical Engineering Science*, 62(11):2890–2904, 2007.

[11] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital Twin in manufacturing: a categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018.

[12] F. Longo, L. Nicoletti, and A. Padovano. Ubiquitous knowledge empowers the Smart Factory: the impacts of a Service-oriented Digital Twin on enterprises' performance. *Annual Reviews in Control*, 47:221–236, 2019.

[13] Y. Lu, C. Liu, I. Kevin, K. Wang, H. Huang, and X. Xu. Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61:101837, 2020.

[14] P.J. Mosterman and J. Zander. Industry 4.0 as a cyber-physical system study. *Software & Systems Modeling*, 15(1):17–29, 2016.

[15] L. Pérez, S. Rodrguez-Jiménez, N. Rodrguez, R. Usamentiaga, and D.F. García. Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning. *Applied Sciences (Switzerland)*, 10(10):Article No. 3633, 2020.

[16] M. Schamp, S. Hoedt, A. Claeys, E.H. Aghezzaf, and J. J. Cottyn. Impact of a virtual twin on commissioning time and quality. *IFAC-PapersOnLine*, 51(111):1047–1052, 2018.

[17] M. Sjarov, T. Lechler, J. Fuchs, M. Brossog, A. Selmaier, F. Faltus, T. Donhauser, and J. Franke. The Digital Twin concept in industry–a review and systematization. In *Proc. 25th IEEE International Conference on Emerging Technologies and Factory Automation*, volume 1, pages 1789–1796, Vienna, Austria, 2020.

[18] F. Tao, H. Zhang, A. Liu, and A.Y.C. Nee. Digital twin in industry: state-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, 2018.

[19] K. Thramboulidis. A cyber–physical system-based approach for industrial automation systems. *Computers in Industry*, 72:92–102, 2015.