International Conference on Adaptive Modeling and Simulation
ADMOS 2023
F.Larsson and P.Díez

INTERPRETABLE AND REUSABLE REDUCED ORDER MODELS FOR DIGITAL TWINS IN MANUFACTORY AS A SERVICE

Valentina Zambrano*†, Ismael Viejo†, José Manuel Rodríguez† Guillermo López†, Jesús Alfonso†, Daniel Cáceres†, María López-Blanco†, Prasad Talasila‡, Mario Sánchez†, and Susana Calvo†

† Instituto Tecnológico de Aragón - ITAINNOVA
C/ María de Luna 7-8, 50018 Zaragoza (Spain)
e-mail: vzambrano@itainnova.es - web page: https://www.itainnova.es

‡ Aarhus University
Nordre Ringgade 1, 8000 Aarhus (Denmark)

Key words: Reduced Order Modelling, Digital Twins, Manufactory as a Service, Interpretable Models, Machine Learning, Artificial Intelligence

Summary. The increasing interest on Digital Twins (DTs) solutions in Industry 4.0 (I4.0) is transforming industrial processes towards a more profitable and sustainable production. In an industrial environment DTs enable the creation of virtual replica of industrial products, services, and processes, allowing a more effective management.

The *DIGITbrain* project [1] aims for the development of an integrated digital platform to provide Small and Medium-sized Enterprises (SMEs) access to DT technology. Within this context several use cases have been created using different types of models. We have developed models using CAELIA™, an authoring tool developed at ITAINNOVA for the generation and the management of Reduced Order Modelling (ROM) -based models [2].

CAELIA™ROM models are obtained through the Twinkle library (which can work both on dense and sparse data, and it is especially designed for unstructured data), and are based on self-adapted Tensor Rank Decomposition (TRD).

Moreover, we have developed a real-time co-simulation structure linking Gazebo robot environment and controllers in Matlab using delay compensators [3]. The use case is a quarter car, as a sample for the foreseen vehicles to be integrated into a set of automated robots' fleet, by means of co-simulation. Such a system of robotic vehicles uses RabbitMQ for node-master communications, enabling remote control and autonomous movements.

All use cases developed at ITAINNOVA were conceived within the *DIGITbrain* environment, where all applications must be entirely reusable. A recombination of use case parts were proven to be reused in different scenarios, such as ROM, co-simulation, and Machine Learning (ML).

1 INTRODUCTION

Digital Twins (DTs) are virtual representations of the real world. There are several techniques belonging to the DT paradigm, such as the Reduced Order Modelling (ROM). ROM generates mathematical models to simulate real systems in real-time.

Due to their versatility, DTs are becoming increasingly popular Industry 4.0 (I4.0) for a real-time monitoring and optimization of both the production process and the products. DTs, and more specifically ROM models, allow manufacturers to improve their processes and products in an automated and efficient way. ROM is a numerical technique whose main goal is the simplification of complex, multi-variable, systems (e.g., computationally intensive simulations, big datasets, etc.) into a significantly efficient and straightforward mathematical expression. ROM allows creating DTs of complex systems by significantly reducing their complexity and preserving their main characteristics. The general concept of ROM can be understood by looking at Figure 1. ROM techniques can be divided into two main groups: intrusive and non-intrusive methods. Intrusive methods are based on equations and directly modify these to achieve a simplification of the system; hence these methods cannot be applied in purely data-driven scenarios. On the other hand, non-intrusive methods are based on data; therefore, these methods can be also applied when equations are known.

The ROM approach used at ITAINNOVA is performed through an in-house developed software named Twinkle: a non-intrusive data-driven library based on Tensor Rank Decomposition (TRD). Twinkle is a ROM library of CAELIA™[4], an in-house built toolkit, as detailed in [5, 2]. TRD is a mathematical method for tensor decomposition, based on the assumption that a problem of a given number of not necessarily independent variables can be rewritten as the product of one-dimensional functions, one for each of the variables of the system, as shown in Equation 1.

$$F(v_1, \ldots, v_D) = \sum_{t=1}^{T} \alpha_t \prod_{d=1}^{D} f_{t,d}(v_d) \tag{1}$$

Where D is the dimension of the space in which the system is defined (i.e., the number of variables describing it); is the ROM model's approximation order of the series expansion; α_t, where $t = 1, \ldots, T$, are weighting coefficients; the functions f_d, t are 1-D functions, each of them having separately one input variable as argument.

The first term of the sum in Equation 1 represents a first approximation of the system, being α_1 usually the largest coefficient, while the following terms are corrections to it.

TRD method determines the 1-D functions by means of an iterative procedure that converges quite fast. It also enables to describe the behaviour of complex systems where the variables can have an entangled influence on each other, being the result a simplified mathematical function that accounts for each variable's effect separately, as in Equation 1.

Moreover, Twinkle library can work with full, sparse, and even unstructured data. Thanks to the TRD approach on which its ROM method is based, Twinkle allows preserving the main features of a system and to retrieve a completely physically interpretable result. Twinkle's output is no black-box, but a well-known mathematical expression where the contribution of each system's variable is isolated, as shown in Equation 1.

The H2020 European Project *DIGITbrain* [1] aims to bring the DT technology to Small and Medium Enterprises (SMEs) by the implementation of shared platform for modular construction of DTs, as described in detail in [2]. One important asset in *DIGITbrain* is the model, which along with algorithm, microservices and data, constitute a DT. Thanks to the use of generic metadata [6] (for further information please refer to the documentation web page [7]), which were defined within the Project. Moreover, as described in [2], several types of models are supported in the Project, such as ROM, Machine Learning (ML), Life Cycle Assessment (LCA), System of Systems (SoS), co-simulation, among others. In the following sections some model applications are described.

2 APPLICATIONS

In this section three different applications, developed at ITAINNOVA, are described. One implementation is based on the findings obtained in [2], where a thermoplastic injection moulding application was developed, while the other two models are oriented to laboratory-on-chip on edge computing and co-simulation for distributed environment, respectively.

2.1 Thermoplastic injection moulding

As described in [2], a sample application for thermoplastic injection moulding was developed using CAELIA™'s ROM generator library Twinkle [5]. The computed model was kept generic for an arbitrary injection process, and it was improved during the next steps of the *DIGITbrain* Project.

To compute a DT of a thermoplastic material injection moulding, all parameters involved need to be optimized during the whole process. By following this approach, a real-time DT was obtained. At first, the solution was deployed on the Projet's cloud-based platform as a job (for further details, please, refer to [2]). The solution was further modified and only the ROM model evaluation functionality of Twinkle was considered. The model evaluation was executed on several data points and on different ROM models, so that results for untested parameters combinations were obtained for different output variables at the same time. The functionality evaluates the models' response by solving several TRD equations (Equation 1).

The revisited injection application has been published on the *DIGITbrain* platform [1] as an App, and, later as a Data-Model-Algorithm (DMA) tuple, where all the different assets and microservices are published separately on the integrated platform for reuse.

2.2 Laboratory-on-chip

A factory-on-chip laboratory was built up at ITAINNOVA (see Figure 2); it was used to carry out experiments for real-time system control, simulations and optimization on the edge. The analysis and model construction were performed on a microfluidic chip; the chip consists of two longitudinal channels separated with a flexible membrane that is deformed depending on the flow that is going through each channel. The microfluidic device was selected for biological applications, e.g., heart, lung, and intestine cells culture, where tissue movements are emulated and hence a high precision in system control is required. The geometry of the microfluidic device is shown in Figure 3.

During the first set of experiments a Digital Image Correlation (DIC) technique was used to ensure a ground truth for images and deformations obtained in the experiments. DIC technique allows to measure displacements and deformations on a solid material surface through contactless non-intrusive sensors. The membrane displacement field was obtained using a DIC system imaging validation on the factory-on-chip laboratory set up at ITAINNOVA, as shown in Figure 4.

In the experiments carried out at ITAINNOVA, chip deformation, together with hydrodynamic instabilities and fluctuations that arise at fluid interfaces were studied for different operation conditions, i.e., different fluid pressure values. Moreover, geometry meshes and boundary conditions were studied and generated. The behaviour seen in the microfluidic laboratory corresponds to a strong coupling between the fluidal and mechanical domain. Pressures in the fluid domain infer a pressure field on the membrane surface, which leads to further changes in the fluid domain pressure distribution, until a thermodynamic equilibrium is reached. Due to this behaviour, the simulation models to be used need to be a two-way CFD-FEM coupling solver. An analytical approach was developed considering the strong coupling between the fluidal and mechanical domain; the inputs for this analytical approach being the mass flow or pressure inlet in each channel, while the output will be the theoretical pressure on the membrane surface. The pressures obtained in the previous step via the analytical model are to be used for FEM analysis.

A ROM model can then be generated using FEM simulations, where the DoE input is a pressure field and the output is the membrane deformation or stress fields. As described in Section 2.1 and [2], the obtained ROM can be used on the platform to be evaluated in real-time for several set of data points.

2.3 Distributed environment

This application, developed at ITAINNOVA in collaboration with Aarhus University, is oriented to a distributed environment of autonomous vehicles, by means of co-simulation models to be published on the *DIGITbrain* Project, as previously discussed in [8]. A simplified use case was defined to check the correct functionality and connectivity of a complete robotic system. The main building bricks used in this application are the following.

- *Maestro* [9] as orchestrator of the co-simulation.
- *Matlab* as possible alternative orchestrator or as an agent of the co-simulation.
- *ROS-Gazebo* [10] simulation environment from the robotics field that can work as an element of the co-simulation.

The simplified model is the quarter car model in the figure below, including a wheel described as a spring and a damper which suffers the perturbations coming from the road. These perturbations are transmitted through the suspension system represented by a second spring-damper-actuator joined to the upper chassis. The actuator in the suspension stands for the active controller.

The model evaluation is done in successive complexity levels, as it follows.

- *FMU co-simulation running under Maestro or in Matlab:* in this case two FMUs for the wheels and the chassis are used for building up the complete model.
- *Extension of the previous co-simulation:* the wheel-chassis representation being replaced by a complete model in the Gazebo environment.
- *Link Gazebo, Maestro, and Matlab* in the same co-simulation; the main interest in using this configuration is that the robot systems are normally tested in the ROS Gazebo environment, but the design of the controllers is partly done in Matlab.

The complete chassis-wheel elements of the quarter car model were described using the ROS Gazebo environment. The Gazebo model has the following exchange variables.

- Inputs
  - Force coming from the controller for simulating the active suspension.
  - Road profile (Vertical position, speed, and acceleration).
- Outputs
  - Position, and speed of the wheel, and the chassis masses.

To represent the perturbations of the road, an intermediate heavy mass is added, as shown in Figure 5, where $M1$ represents the chassis, $M2$ the wheel, and $M3$ is the mass for transmitting the perturbations coming from the road ($F3$). F1 is the actuation force coming from the controller. In Gazebo the system is described using primitives, so the quarter car mass $M1$ is represented by a cube, the wheel mass $M2$ by a cylinder and the road perturbation $M3$ by a $2x2m$ tape $10cm$ thick.

The complete co-simulation model of the distributed robot fleet can be published on the *DIGITbrain* Project's integrated platform for use and individual assets future re-use.

3 CONCLUSIONS

The European H2020 Project *DIGITbrain* aims at creating an integrated digital platform to promote the use of DTs in the industry. In this paper several applications developed at ITAINNOVA, within the *DIGITbrain* Project, have been detailed. Through the different examples listed in this paper, the reader can easily see the possibility of using several types of models in the integrated platform provided by the *DIGITbrain* Project, such as ROM and co-simulation models, among many others. Moreover, the successful publication of the presented examples into several components and assets, i.e., Data, Model, Algorithm and Microservices, hints at the independence of their future usability in different applications.

REFERENCES

[1] *DIGITbrain Project*. [Online]. Available: *https : //digitbrain.eu/*

[2] Zambrano, V., Mueller-Roemer, J., Sandberg, M., Talasila, P., Zanin, D., Larsen, P. G., ... and Stork, A. (2022). *Industrial digitalization in the industry 4.0 era: Classification, reuse and authoring of digital models on Digital Twin platforms*. Array, 14, 100176

[3] Alfonso, J., Rodriguez, J. M., Bernad, C., Beliautsou, V., Ivanov, V., and Castellanos, J. A. (2022, May). *Geographically distributed real-time co-simulation of electric vehicle*. In 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT) (Vol. 1, pp. 1002-1007). IEEE.

[4] *GitHub CAELIA*. [Online]. Available: *https : //github.com/caeliaITAINNOVA/*

[5] Zambrano, V., Rodríguez-Barrachina, R., Calvo, S., and Izquierdo, S. (2020). *TWINKLE: A digital-twin-building kernel for real-time computer-aided engineering*. SoftwareX, 11, 100419.

[6] *GitHub DIGITbrain*. [Online]. Available: *https : //github.com/DIGITbrain/digitbrain.github.io*

[7] *DIGITbrain Documentation*. [Online]. Available: *https : //digitbrain.github.io/tables/*

[8] Talasila, P., Crăciunean, D. C., Bogdan-Constantin, P., Larsen, P. G., Zamfirescu, C., and Scovill, A. (2022). *Comparison Between the HUBCAP and DIGITBrain Platforms for Model-Based Design and Evaluation of Digital Twins*. Software Engineering and Formal Methods, SEFM 2021 Collocated Workshops: CIFMA, CoSim-CPS, OpenCERT, ASYDE, Virtual Event, December 6–10, 2021, Revised Selected Papers (pp. 238-244). Cham: Springer International Publishing.

[9] Thule, C., Lausdahl, K., Gomes, C., Meisl, G., Larsen, P. G. (2019). *Maestro: The INTO-CPS co-simulation framework*. Simulation Modelling Practice and Theory (Vol. 92, pp. 45-61). ISSN 1569-190X.

[10] *Gazebo tutorial*. [Online]. Available: $https : //classic.gazebosim.org/tutorials?tut = ros_overview$

4 FIGURES

Figure 1: Stanford bunny. Harvard University, Microsoft Research.

Figure 2: Factory-on-chip laboratory at ITAINNOVA.

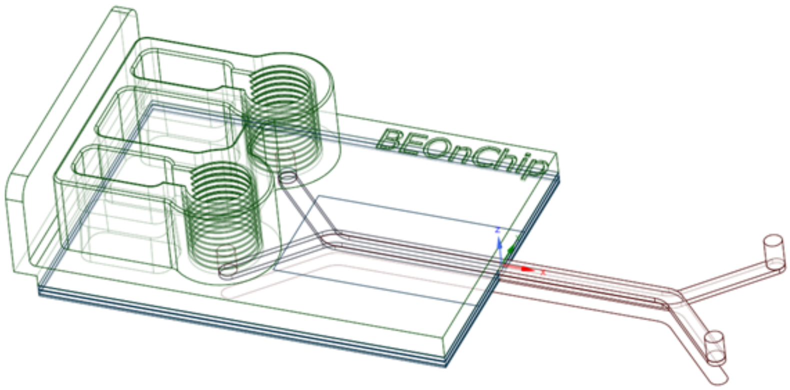

Figure 3: Geometry of the microfluidic device used at ITAINNOVA within the microfluidic laboratory.

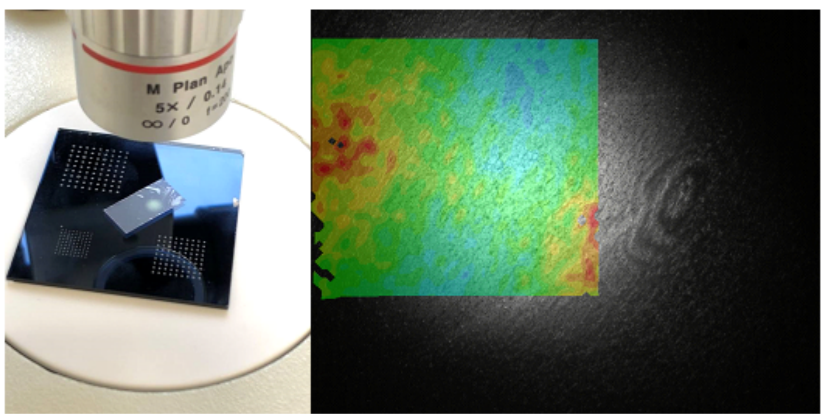

Figure 4: Example of a DIC system measurements on a membrane.

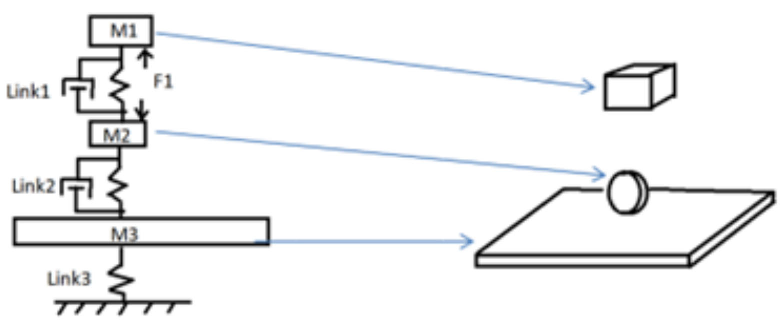

Figure 5: Definition of a quarter car model in Gazebo.