Hybrid twin of RTM process at the scarce data limit

Sebastian Rodriguez, Eric Monteiro, Nazih Mechbal, Marc Rebillat, Francisco Chinesta

Hybrid twin of RTM process at the scarce data limit

Sebastian Rodriguez[1,*], Eric Monteiro[1], Nazih Mechbal[1], Marc Rebillat[1]
Francisco Chinesta[1,2]

[1]PIMM, ENSAM Institute of Technology, 151 Boulevard de l'Hôpital, 75013, Paris, France.
[2]ESI Group Chair, PIMM, ENSAM Institute of Technology, 151 Boulevard de l'Hôpital, 75013, Paris, France.

January 29, 2024

Abstract

To ensure correct filling in the resin transfer molding (RTM) process, adequate numerical models have to be developed in order to correctly capture its physics, so that this model can be considered for process optimization. However, the complexity of the phenomenon often makes it impossible for numerical models to accurately predict its behavior, limiting its usage. To overcome this limitation, numerical models are enriched with measured data to ensure their correct predictability. Nevertheless, the data used is often limited due to practical constraints, such as a limited number of sensors or the high costs of experimental campaigns. In this context, the present paper demonstrates the implementation of a numerical model enriched with data, called Hybrid Twin applied to the RTM process when few sensors are considered in the mold to be injected. The performances of the developed hybrid twin are tested in a virtual test for the injection of a 2D mold, where the hybrid twin constructed using a simplified numerical model allows to accurately predict a complex model's resin flow-front over its entire time history.

Keywords: Resin transfer molding, Virtual twin, Hybrid twin, sparse - Proper Generalized Decomposition, Model-order reduction, Inverse analysis.

1 Introduction

Over the years, composite materials have become more and more trendy, especially in industrial applications, where they are applied in various fields such as automotive, aeronautics, etc. Their advantageous mechanical properties, functional performances and reduced weight make them an attractive material for many high-tech applications. In this sense, the optimization of their manufacturing process is presented as a key aspect in the coming years, where manufacturing defects and production times must be minimized. Among the main manufacturing process of composite structures is Resin Transfer Molding (RTM) process [1, 2]. This process

consists of having a preform of the mold composed of fibrous material, to which a resin matrix is added by injection as illustrated in Figure 1.

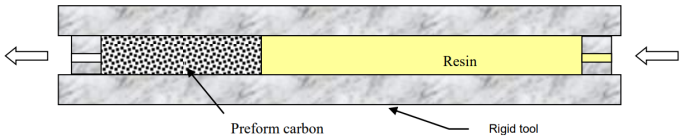


Figure 1: Illustration of the resin transfer molding (RTM) process.

To ensure correct filling in the RTM process, numerical models have to be used, where these models have to be rich enough to being able to predict this process under different operational conditions. For this reason, these models are usually built by considering real measured data, in order to correctly capture the physics of the procedure. However, in real-world situations, the data that can be used is limited due to practical constraints or high-sensing cost. In this context, the present paper focus in the development and implementation of a numerical model enriched with data applied to RTM when few sensors are considered, which is able to correctly predict the real flow-front of the process.

Numerical models based on physics laws and mathematics are able to perform numerical predictions efficiently. These models involves the use of well-known techniques as the Finite Element Method (FEM) [3], Finite Difference Method [4], etc. Along with detailed and well understood constitutive behaviors. This type of numerical model is here denoted as *virtual twin* [5]. Although numerical models help to predict many real systems, in many of them, especially those involving complex physical mechanisms, numerical simulations are limited in providing a correct prediction of the system. Additionally, as numerical models consider more complex and nonlinear behaviors in their formulations, their computational efforts to produce predictions increase, thus slowing down the design process and optimization of the system under study, something capital, for instance, when performing control of a manufacture process.

For this reason, at the end of the 20th century, data irrupted in scientific fields, where data-based models based on the use of linear and nonlinear regressions, machine learning [6], deep learning techniques [7, 8, 9], etc. Progressively helped to solved problems whose virtual twin counterpart where not able to correctly simulate due to the increase in complexity. Even more, these models were able to give predictions under real-time constraints [10, 11, 12], which is an important aspect when performing control of a system. These pure data-based models are commonly denoted as *digital twin*. However, these data-based models have a great disadvantage, which corresponds to the high quantity of data required for their construction and calibration, being a great limitation especially when experimental campaigns are expensive. Moreover, they are difficult to explain and to certify. Another risk is associated with their ability to be extrapolated far from the domain where they were learnt.

For this reason, a new type of twin has been proposed in [5], the *hybrid twin* (HT), which combines a physics-based model (virtual twin) with data, capable of overcoming the limitations of their virtual and digital counterparts [13, 14, 15]. The main idea is to fit a virtual twin with respect to the data measured by sensors in

real-time, allowing the numerical simulation to better predict the experience, by estimating and correcting on the fly the difference between the numerical predictions and reality. Indeed, this difference can be coarsely simulated by an *ignorance model*. In simple terms this means:

$$\text{Real experience} \approx \text{Hybrid twin} = \text{Virtual twin} + \text{Ignorance model} \tag{1}$$

This ignorance model needs fewer experimental measurements for its construction, since much of the prediction of the experience is captured by the calibrated numerical model. The only inconvenience of the above-mentioned requirements for the construction of the hybrid twin consists in the need to perform a calibration of the numerical model with respect to the measured data in real-time, which is not always possible due to the complexity of the model. This limitation is overcome by the use of model-order reduction (MOR) techniques applied to the numerical model (virtual twin). MOR techniques allows compressing and approximating the model and thus obtaining real-time predictions in function of the different parameters, allowing a real-time calibration with respect to the measured data. The MOR technique here is used as an efficient interpolation technique. It allows approximating a given solution under a specific format that permits the real-time feedback constraints while maintaining a good accuracy in the predicted solution.

The principal idea of model-order reductions methods consists in reducing the dimensionality of the original space where a given data lives, by capturing their principal features such as it can be approximated on a reduced space. Among the most powerful techniques that allow this compression while constructing an interpolated model, one finds the Proper Generalized Decomposition (PGD). The PGD was first introduced in the context of the Large Time Incremental (LATIN) method [16] under the vocable of *radial approximation*. It was originally applied to approximate a nonlinear solution in the space-time domain as a sum of product of functions with separate space-time variable representations by using the LATIN solver [17, 18, 19, 20, 21, 22, 23]. Since then, the PGD method has been extended from the approximation of space-time problems to deal with parametric problems [24, 25, 26], optimal transport problems [27], temporal multiscale decomposition [28, 29, 30, 31, 23], among many others.

The PGD method can be implemented in an *intrusive* or a *non-intrusive* way. The intrusive way consists in directly modifying the partial differential equation of a given problem in order to approximate its solution under a separate variable format, while the non-intrusive one simple consists in performing an "offline" calculation using the high fidelity model and then compressing its results. The non-intrusive version of the PGD is generally considered for the construction of a hybrid twin, where an offline computation over a parametric space using the virtual twin must be performed. Unfortunately, this offline calculation can be really expensive if a large number of parameters are taken into account due to the so-called curse of dimensionality [32]. In this context, a new version of the PGD has been introduced in order to overcome this difficulty, the so-called sparse-PGD [33, 34], allowing to compress and compute the PGD decomposition using unstructured and sparse data over a parametric space [35] while keeping the offline computations inexpensive.

Two principal advantages can be summarized when using a hybrid twin: (i) efficient inverse analysis and (ii) real-time control and optimization of a system. The first is a natural consequence of achieving real-time feedbacks of the virtual twin prediction

as a function of parameters when using model reduction techniques. The second is a direct consequence of the hybrid twin itself by allowing to correctly approximate a complex system, where its use is crucial for the application of control techniques to guide the real phenomena to a specific solution. The advantages of the hybrid twin, especially in the reduction of the data required for its construction, make it a really attractive technique for a wide range of applications, both in the area of solid and fluid mechanics, as well as in the improvement and monitoring of production processes. It is this advantage, which makes its application to the Resin Transfer Molding (RTM) process really attractive.

The first introduction of the hybrid twin framework for the RTM process was presented in [36], for the case of a single injection nozzle that allowed a radial spread of the resin inside the mold. Also in that work a camera was used for continuous monitoring of the resin inside the mold, having rich information about the process at every moment. The work presented in this paper deals with a more complicated set-up, since it is an injection process that first (i) produces a complex resin flow-front and second (ii) limited sensors are distributed along the mold.

In particular, the reference problem considered herein consists on a 2D problem. Although the ideas presented in this article are applied to this 2D case, their extension to 3D problems can be developed without major difficulties. The mold taken into account have an injection zone, where a unique source of pressure p_{imp} is imposed, the pressure of the empty part p_{out}, a preform zone where the "skeleton" of the piece composed of fibers is situated at the center of the mold and finally a "race-track zone" all along the boundary of the preform. The purpose of this zone is to induce a high permeability in it, so that the resin can flow faster through it. The data coming from the "real" experience used to enrich the numerical simulations are measured by some sensors distributed along the mold, which provide information on the arrival times of the resin in them. The mold is considered to contain $n_{s,x}n_{s,y}$ sensors in total, $n_{s,y}$ sensors along the y axis and $n_{s,x}$ sensors along the x axis. The main considerations of the 2D model are illustrated in Figure 2.

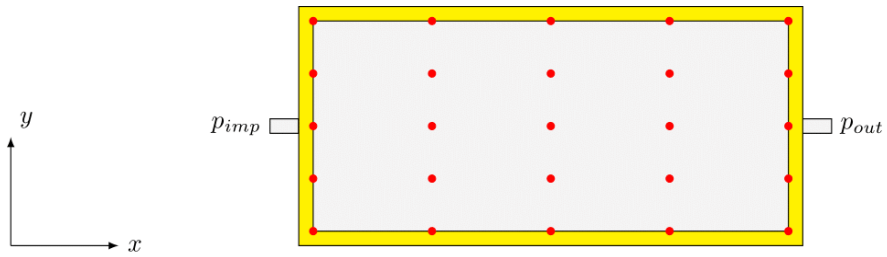


Figure 2: 2D mold with $n_{s,x} = n_{s,y} = 5$ sensors represented by red dots; the yellow color represents the race-track zone.

In this sense, this work presents the construction of a hybrid twin of the RTM process when limited data are available to correctly approximate the complex process. The present work serves, therefore, as a starting point for future developments to improve the RTM process, which may contemplate the minimization of defects in the manufactured part [37], while minimizing production times in an industrial scenario.

The present paper is structured as follows: section 2 presents the virtual twin related to the RTM process, section 3 presents the construction of the hybrid twin along with all the different algorithms used for its implementation. Following section 4 presents

a numerical example showing the performance of the hybrid twin for predicting a complex RTM process. Finally section 5 gives some conclusions and perspectives.

2 Virtual twin of the RTM process

The virtual twin corresponds to the physics-based model of the RTM process. The model considered here consists of a fixed-mesh formulation using finite element method (FEM) based on the work of [38, 39]. The idea consists in dividing the domain of the mold Ω on a domain occupied by the fluid resin and the empty domain denoted respectively $\Omega_f(t)$ and $\Omega_e(t)$, which evolves in time such as $\Omega = \Omega_f(t) \cup \Omega_e(t)$ (the inlet domain is considered in Ω_f at time $t = 0$). This strategy considers two variational formulations, one applied to the fluid domain, which considers all the boundary conditions of the problem and the other applied to the empty part of the mold, where an empty pressure is imposed, which is considered here as a null value. A weighting formulation is considered in the partially filled elements, where the weighting function depends on the fluid fraction in each element. This fluid fraction is updated for each time instant (allowing the simulation of fluid propagation). More details can be found on [38]. The governing equations are detailed below.

2.1 Governing equations

The phenomena of fluid passing throw a porous media is modelized by using Darcy's law. This law gives the velocity $\underline{v}(\underline{x}, t)$ of the fluid in function of the permeability tensor $\underline{\underline{K}}$ of the filled material and the pressure $p(\underline{x}, t)$ of the fluid as follows:

$$\underline{v}(\underline{x}, t) = -\frac{\underline{\underline{K}}}{\mu} \nabla p(\underline{x}, t) \ , \ \forall \underline{x} \in \Omega_f(t) \tag{2}$$

where the vector $\underline{x}$ denotes the Cartesian coordinates and μ the fluid viscosity. In addition, the fluid under consideration must also verify the incompressibility condition, which states:

$$\nabla \cdot \underline{v}(\underline{x}, t) = 0 \ , \ \forall \underline{x} \in \Omega_f(t) \tag{3}$$

where $\nabla \cdot \bullet$ denotes the divergence of $\bullet$. The fluid domain is characterized by a variable $I(\underline{x}, t)$ representing the fraction of fluid contained in each spatial point $\underline{x}$ at a given time t. This variable results from the solution of the transport equation:

$$\frac{\partial I(\underline{x}, t)}{\partial t} + \underline{v}(\underline{x}, t) \cdot \nabla I(\underline{x}, t) = 0 \tag{4}$$

and takes values in between 0 and 1, where 0 denotes no presence of fluid and 1 that the point is fully filled with fluid.

$$I(\underline{x}, t) = \begin{cases} 1 & \text{if } \underline{x} \in \Omega_f(t) \\ 0 & \text{if } \underline{x} \in \Omega_e(t) \end{cases} \tag{5}$$

The details for the solution of the transport equation (4), which involves the use of *volume of fluid* technique and the *Total Variation Diminishing* (TVD) method can be found in [38].

As exposed previously, here one considers modeling the fluid and the empty part of the mold at the same time in the same variational formulation. Since the pressure

in the empty zone verifies a null pressure, one can write the variational formulation associated to the empty region as:

Find $p \in \Omega_e(t)$ such that $\forall p^* \in \mathcal{U}(0)$,

$$\int_{\Omega_e(t)} p^* p(\underline{x}, t)\ d\Omega = 0 \quad , \quad \text{with } p(\underline{x}, t) = p_{out} = 0\ ,\ \forall \underline{x} \in \Omega_e(t) \tag{6}$$

where $\mathcal{U}(0)$ denotes the pressure space with null boundary conditions. On the other hand the variational formulation associated to the fluid is given by:

Find $p \in \Omega_f(t)$ such that $\forall p^* \in \mathcal{U}(0)$,

$$\int_{\Omega_f(t)} \nabla p^* \frac{\underline{\underline{K}}}{\mu} \nabla p\ d\Omega = 0 \tag{7}$$

By combining both variational formulations (6) and (7) and by using the fraction of fluid I, one finally obtain the variational formulation that must be solved, which is defined in the whole domain Ω as:

Find $p \in \Omega$ such that $\forall p^* \in \mathcal{U}(0)$,

$$\int_{\Omega} \left((I) \nabla p^* \frac{\underline{\underline{K}}}{\mu} \nabla p \ + \ (1 - I) p^* p \right) d\Omega = 0 \tag{8}$$

The time evolution of the RTM process is obtained by simply solving equation (8) together with the updating of the domain occupied by the fluid by solving equation (4) for each time instant until the fluid eventually occupies the entire domain Ω.

3 Hybrid twin of RTM process

At this point, a physics-based numerical model that allows us to simulate the RTM process has been presented. However, even a numerical model that includes all types of physics in the system in the vast majority of situations is not capable of providing accurate predictions with respect to a real process. The hybrid twin provides an answer to this need. The details and fundamental ingredients related to the construction of this hybrid twin are presented in the following sections.

3.1 Main ingredients for the construction of the hybrid twin

As presented in [36, 40, 5], the hybrid twin is constructed by modeling the ignorance between the virtual twin predictions and the real measurements for a set of given parameters. The main hypothesis in order to perform this consists in that the virtual twin even if it does not gives accurate results compared to the reality, gives a prediction that catches the principal aspects or behavior of a system. Due to the last, the ignorance model is considered to add a coarse correction to the virtual twin, where all together becomes the hybrid twin that allows to reproduce close predictions compared to the reality. Mathematically speaking, the governing equation of a hybrid twin [5] is given by:

$$\dot{\underline{x}}(t; \underline{\nu}) = \underline{A}(\underline{x}, t; \underline{\nu}) + \underline{B}(\underline{x}, t; \underline{\nu}) \tag{9}$$

which expresses the rate of change of the system state here noted $\dot{\underline{x}}(t; \underline{\nu})$ at time t by considering any parameter set ν of the system. Here, the state of the system

corresponds to the resin flow-front inside the mold, and as will be shown in the following sections, this flow-front is reconstructed using the measurement of resin arrival times at sensors. This equation is composed of two principal contributions: (i) the rate of change of the virtual twin for any parameter set $\underline{\nu}$ ($\underline{A}(\underline{x}, t, \underline{\nu})$) and (ii) a data-based model describing the rate of change of the gap between the prediction of the virtual twin and the measured data for the parameter set $\underline{\nu}$ ($\underline{B}(\underline{x}, t; \underline{\nu})$). Figure 3 summarizes the main ideas of the hybrid twin applied to the RTM process.

Since in real life, experimental measurements are in general very expensive, from the above, the ignorance model should be as coarse as possible. To ensure this, the virtual twin must first be fitted the best as possible with respect to an experimental measurement by identifying the best process parameters. In addition, since the hybrid twin is aiming to be used for the prediction of a real RTM process, its usage and construction should be performed online, since different pieces (with potentially different material properties) have to be manufactured. In this sense, it is of primal importance to be able to identify the parameters of the process under real-time constraints, to fit the virtual twin to the data and use the existing hybrid twin model associated to that parameter. To achieve this, the use of model-order reduction techniques is key for the real-time feedback and optimized inverse algorithms for fast determination of parameters. These techniques are presented in the following sections.

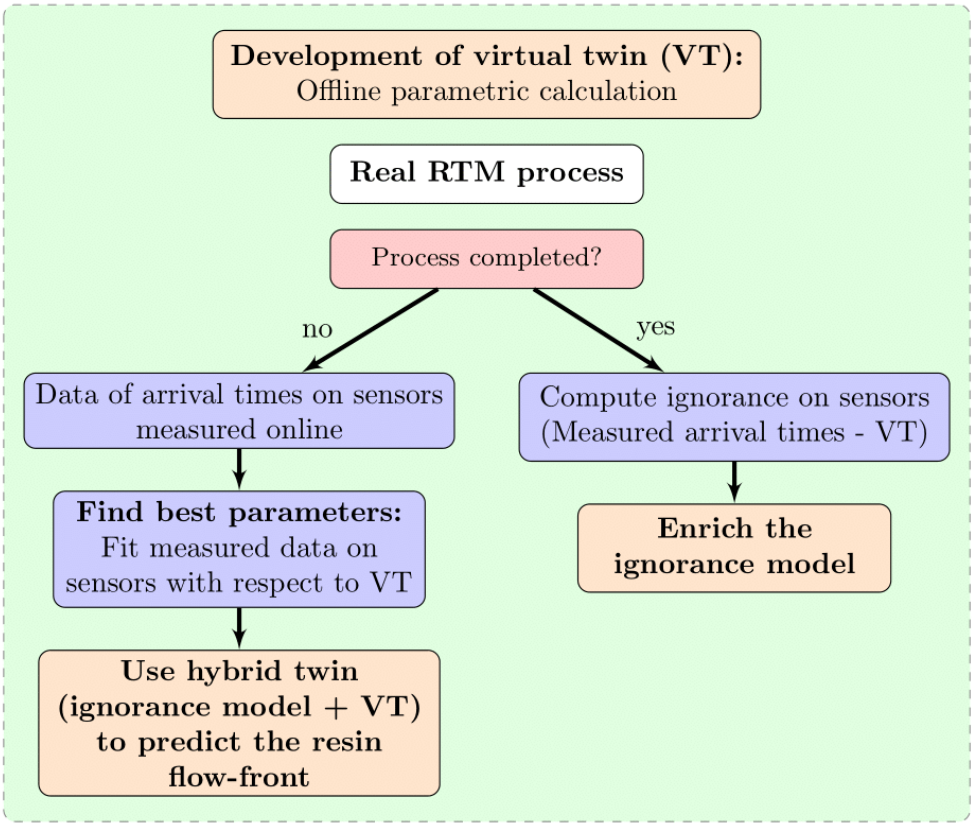


Figure 3: Hybrid twin methodology diagram applied to the RTM process.

3.2 Model-order reduction technique: the Proper Generalized Decomposition

Proper Generalized Decomposition is a model-order reduction technique, whose main objective consists in approximating a given data on a reduced space (compared to the original one) and represent the approximation in a separate variable multiplication format [26]. In the general case, one intends to approximate a given quantity $f(\nu_1, \nu_2, ..., \nu_n) \in \mathbb{R}^n \rightarrow \mathbb{R}$ in the following format:

$$f(\nu_1, \nu_2, ..., \nu_n) \approx \sum_{k=1}^{m} X_1^k(\nu_1) X_2^k(\nu_2) \ ... \ X_n^k(\nu_n) \tag{10}$$

where m denotes the number of multiplied terms that need to be summed up to approximate the reference function f, this variable is denoted as *modes* of the decomposition in the model-order reduction community. The PGD functions are built using adequate shape functions over each parametric dimension, interpolating the approximation over the whole parametric space.

However, its classical construction is performed over structured data (i.e the function f should be known at specific values in the parameter space, such as the known data is structured). This requirement impose a limitation when dealing with large amount of parameters since the number of data needed explodes with the number of parameters (the computation time needed to obtain the function f on those parameters grows exponentially too). This phenomenon is known as *curse of dimensionality*. Therefore, a more suitable version of the PGD decomposition is needed to solve the aforementioned limitation, especially if the virtual twin considered depends on many parameters. This improved version corresponds to the sparse-PGD [33]. This variant of the PGD allows to construct a low-rank approximation by considering non-structured data over a parametric space. This method is presented in detail in Appendix 6.

3.2.1 Compression and approximation of virtual twin by using sparse - PGD

The strategy used to compress the parametric results from the offline calculation of the virtual twin simply consists of approximating the arrival times of the resin at each sensor in the mold, which are illustrated in Figure 2. In detail, a sPGD approximation is applied to each sensor to approximate the arrival time of the resin on the sensor, depending on the parameters considered in the numerical model, such as injection pressure, permeability of the preform, viscosity of the resin, etc. In this sense, lets define the sPGD approximation of the arrival times on sensor i as follows:

$$T^{[i]}(\nu_1, \nu_2, \dots, \nu_n) \approx T^{[i]}_{m_i}(\nu_1, \nu_2, \dots, \nu_n) = \sum_{k=1}^{m_i} X_1^{k,i}(\nu_1) X_2^{k,i}(\nu_2) \ ... \ X_n^{k,i}(\nu_n) \tag{11}$$

where ν_j with $j \in [1, ..., n]$ represents any of the parameters considered and $T^{[i]}$ the measured arrival times of the resin over the different sensors.

3.3 Modeling the gap between virtual twin and measured data

The second major ingredient of the hybrid twin, which concerns the construction of the ignorance model is presented in this section. The modeling of the difference

between the virtual twin prediction and the measured data is performed in two main stages. On the one hand, the first step (i) is to identify the process parameters of the system so that the error between the virtual twin and the measured data is minimized. This is performed in order to ensure that the ignorance can be correctly approximated in a coarse way possible. Second (ii), once the optimal parameters are determined, the ignorance for this point in the parameter space is stored and the ignorance model is built or enriched online. This construction raises two possible scenarios. On the one hand, the set of parameters determined lies within the parameter interval of the hybrid twin definition, where this new point is simply considered to enrich the model. On the other hand, if the identified parameter set lies outside the domain of definition, an special treatment should be taken into account since the hybrid twin can't gives a prediction. These two scenarios are treated in the following sections.

3.3.1 Real-time inverse analysis

In order to perform the inverse analysis and determine the RTM parameters in real time throughout the process, the sPGD model reduction technique presented in section 3.2.1 is combined with a nonlinear optimization technique.

The use of the MOR technique sPGD with a nonlinear curve fitting technique was first exploited in [41] for thermal problems. There, the nonlinear optimization algorithm chosen and proven to gives best results corresponds to the Levenberg-Marquardt algorithm [42]. As presented in [41], the combination of model reductions along with high optimized nonlinear least squares curve-fitting models as the Levenberg-Marquardt algorithm allows to identify model parameters at very low computing times, in the order of milliseconds or seconds depending on the number of parameters considered, achieving the desired real-time identification. In our case, the method minimizes the following error:

$$\{\hat{\underline{\nu}}\} = \arg\min_{\{\underline{\nu}\}} \sum_{i=1}^{\text{Nsensors}} \left[\frac{T^{[i]} - T^{[i]}_{m_i}(\underline{\nu})}{T^{[i]}} \right]^2 \tag{12}$$

which involves the arrival times of the resin measured from the real experience ($T^{[i]}$) over all the sensors (Nsensors) considered and the one coming from the compressed virtual twin by using the sPGD ($T^{[i]}_{m_i}$).

The Levenberg-Marquardt algorithm combines two numerical minimization algorithms: the gradient descent method and the Gauss-Newton method. More details about the LM method can be found in [43]. Both gradient-descent and Gauss-Newton method requires the derivative of the considered function, here is where the advantage of the PGD format arises, in fact, any derivative applied to a PGD approximation in a general format can be simply expressed as follows:

$$\frac{\partial f(\nu_1, \nu_2, \cdots, \nu_n)}{\partial \nu_i} \approx \sum_{k=1}^{m} X_1^k(\nu_1) X_2^k(\nu_2) \cdots \frac{\partial X_i^k(\nu_i)}{\partial \nu_i} \cdots X_n^k(\nu_n) \tag{13}$$

This means that to perform the derivative of the considered function with respect to a parameter, it is only necessary to perform the derivative on the respective PGD function related to this parameter. Moreover, the obtained expression is also decomposed by the PGD, making its determination really efficient. This property offers many numerical advantages, which together allow fulfilling the identification restrictions in real-time.

3.3.2 Modeling the discrepancy between the virtual twin and experience

Up to now, the tools to build a model of the so-called *ignorance* have been introduced. Once some experiment realizations and their respective identification of parameters is performed, the information of the gap between the virtual twin prediction and the measurement is stored, which is given by:

$$\Delta T^{[i]}(\hat{\underline{\nu}}) = \underbrace{T^{[i]}(\hat{\underline{\nu}})}_{\text{Measurement}} - \underbrace{T^{[i]}_{m_i}(\hat{\underline{\nu}})}_{\text{sPGD}} \tag{14}$$

Once a sufficient amount of data have been collected, the ignorance model is interpolated along the parametric space using these points. The reader may note that these points are potentially unstructured in nature. From this, it can be quickly concluded that the ignorance model must be built using some algorithm that can create a meta-model from the unstructured data. From here there exists two options for the creation of the ignorance model, on the one hand (i) one can use the model reduction technique sPGD, or on the other hand (ii) Deep Learning techniques can be used. In this paper, the sPGD is used to model this gap, where one defines the approximation as $\Delta T^{[i]}_{\tilde{m}_i}(\underline{\nu})$, such as:

$$\Delta T^{[i]}(\underline{\nu}) \approx \Delta T^{[i]}_{\tilde{m}_i}(\underline{\nu}) \tag{15}$$

where for this case $\tilde{m}$ corresponds to the number of modes considered for the sPGD approximation of the ignorance corresponding to sensor i.

3.4 Flow-front prediction for parameters inside the definition domain

When the identified parameters of a process lies inside the parameter domain of the current hybrid twin, the flow-front prediction of the real process at these parameters is constructed by using the arrival times.

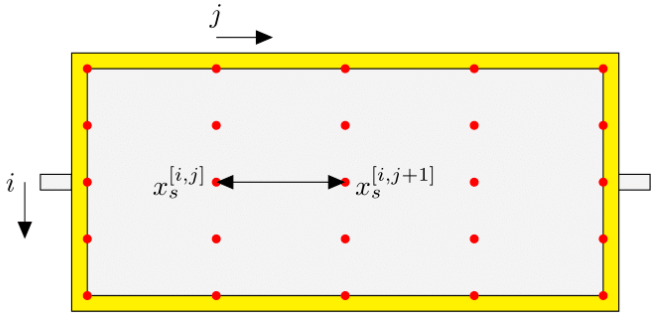


Figure 4: 2D mold with sensors, i denotes the index of the vertical sensors while j the index of the horizontal sensors.

Here, due to the complexity of the problem, where a complex flow-front can occur during the filling of the mold, the position of the flow-front is determined for each line of sensors separately. This is, the rate of change is defined for each of the i sensors located along the y direction (see Figure 4), where the rate of change of the flow-front positioned along the x coordinate associated with the line i of sensors denoted $x^{[i]}(t)$, is approximated as follows:

$\forall i \in [1, ..., n_{s,y}],\ \forall j \in [1, ..., n_{s,x} - 1],$

$$\dot{x}^{[i]}(t; \underline{\nu}) = \frac{x_s^{[i,j+1]} - x_s^{[i,j]}}{\hat{T}^{[i,j+1]}(\underline{\nu}) - \hat{T}^{[i,j]}(\underline{\nu})} \quad \text{when} \quad x^{[i]}(t) \in [x_s^{[i,j+1]} - x_s^{[i,j]}] \tag{16}$$

where $x_s^{[i,j]}$ corresponds to the x coordinate of sensor (i, j) (see Figure 4), $n_{s,x}$ and $n_{s,y}$ the number of sensors considered in the x and y direction respectively, and where it is defined:

$$\hat{T}^{[i,j]}(\underline{\nu}) = \underbrace{T_{m_i}^{[i,j]}(\underline{\nu})}_{\text{sPGD}} + \underbrace{\Delta T_{\tilde{m}_i}^{[i,j]}(\underline{\nu})}_{\text{sPGD of the gap}} \tag{17}$$

The last expression simply state that the velocity of the flow-front along the line of sensor i, when the flow-front is located in between the interval defined by sensors (i, j) and $(i, j + 1)$, is given its horizontal separation divided by the time it takes. Where this time is predicted by the hybrid twin. This approximation is considered since there is no knowledge of the velocity between two horizontally arranged sensors.

Lets note that expression (16) considers the prediction of the real flow-front for each sensor i, therefore, the whole flow-front in the x and y coordinates is reconstructed by simply interpolating the response on the y direction. The accuracy of the reconstruction is illustrated in Figure 5, where the flow-front given by a FEM simulation is correctly approximated using the arrival times knowledge at sensors.

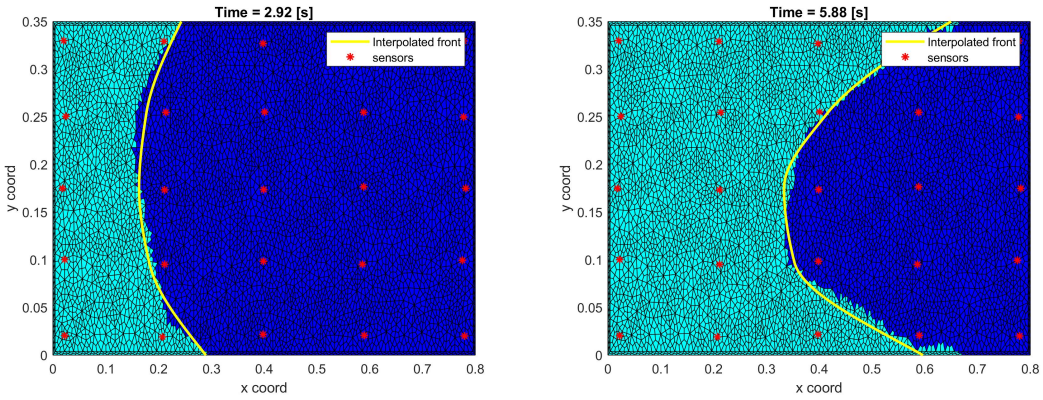


Figure 5: Comparison between the flow-front given by the FEM resolution and the one obtained by interpolation using the arrival times when $n_{s,x} = n_{s,y} = 5$, for two different time instants.

3.5 Flow-front prediction for parameters outside the definition domain

In certain situations the identified parameters or the value of a used parameter may lie outside the construction range of the actual hybrid twin. In this situation, no information is available on the gap between reality and the virtual twin, for this reason a different strategy must be considered. This strategy simply consists in directly approximating the arrival times on sensors for this new set of parameters.

The main idea consists in assuming that the response of the real system for this new set of parameters lives in a reduced space common to all the solutions of the real

system measured up to the present time, so that the new solution can eventually also be approximated in this space, i.e. by the appropriate use of basis functions. These basis functions can be determined in different ways, however, the most direct and efficient way to obtain them is simply by applying the Proper Orthogonal Decomposition (POD) method to all the solutions of the real system obtained previously.

The POD method is a model reduction technique that allows to obtain optimal orthonormal basis that better allows to approximate a given data set [44, 45]. By denoting as $\underline{\nu}_{\text{out}}$ the parametric vector that lies outside the hybrid twin definition, and considering the POD, the arrival times can be approximated as follows:

$$\underline{\boldsymbol{T}}(\underline{\nu}_{\text{out}}) \approx \underline{\underline{\boldsymbol{\Phi}}}\ \underline{\boldsymbol{\alpha}} \quad \text{with} \quad \underline{\underline{\boldsymbol{\Phi}}} = \left[\underline{\phi}_1, \underline{\phi}_2, ..., \underline{\phi}_m\right] \tag{18}$$

where $\underline{\boldsymbol{T}}(\underline{\nu}_{\text{out}})$ denotes the arrival times on all the sensors distributed along the mold, $\underline{\underline{\boldsymbol{\Phi}}}$ the matrix that contains the POD basis $\phi_i\ \forall i \in [1, ..., m]$ and $\underline{\boldsymbol{\alpha}}$ the coefficients that must be determined in order to correctly predict these arrival times for the new parametric set. In view of taking real-time actions (particularly in the case of RTM process control), it is necessary to determine these constants only taking into account part of the arrival time information, that is, as the flow advances in the mold, in order to quickly predict what will be the arrival times in the following sensors. Therefore, for the approximate determination of the vector of coefficients $\underline{\boldsymbol{\alpha}}$, the method called Gappy POD [46] is used here.

The Gappy POD approximate the decomposition (18) when the data is not available over the whole domain. In this sense, lets define $\underline{\hat{\phi}}_i = \underline{\phi}_i(\underline{b}), \forall i \in [1, ..., m]$, where $\underline{b}$ represent the selection of the DOFs associated to the known data, i.e. sensors where there is a measurement of arrival times. In this way, one can define:

$$\underline{\hat{\boldsymbol{T}}}(\underline{\nu}_{\text{out}}) = \underline{\boldsymbol{T}}(\underline{b}; \underline{\nu}_{\text{out}}) \quad \text{and} \quad \underline{\underline{\hat{\boldsymbol{\Phi}}}} = \left[\underline{\hat{\phi}}_1, \underline{\hat{\phi}}_2, ..., \underline{\hat{\phi}}_m\right] \tag{19}$$

where $\underline{\hat{\boldsymbol{T}}}(\underline{\nu}_{\text{out}})$ denotes the measured arrival times on few sensors (the sensors touched by the flow-front at the instant when the Gappy-POD is applied). In this situation, the problem to be solved in order to obtain the approximation of coefficients can be written as:

$$\{\underline{\boldsymbol{\alpha}}_{\text{approx}}\} = \arg\min_{\{\underline{\boldsymbol{\alpha}}\}} \left\| \underline{\hat{\boldsymbol{T}}}(\underline{\nu}_{\text{out}}) - \underline{\underline{\hat{\boldsymbol{\Phi}}}}\ \underline{\boldsymbol{\alpha}} \right\|_2^2 \tag{20}$$

with $\|\bullet\|_2^2$ the Euclidean norm. The main idea consists in determining in an approximate way the vector of coefficients by minimizing (20). By minimizing one obtains:

$$\underline{\underline{\boldsymbol{M}}}\ \underline{\boldsymbol{\alpha}}_{\text{approx}} = \underline{\boldsymbol{f}} \tag{21}$$

where $\underline{\underline{\boldsymbol{M}}} = \underline{\underline{\hat{\boldsymbol{\Phi}}}}^T \underline{\underline{\hat{\boldsymbol{\Phi}}}}$ and $\underline{\boldsymbol{f}} = \underline{\underline{\hat{\boldsymbol{\Phi}}}}^T \underline{\hat{\boldsymbol{T}}}(\underline{\nu}_{\text{out}})$. By solving equation (21) to obtain an approximation of the vector of coefficients, the prediction of the arrival times at all sensors in the mold can be simply recovered by applying the following expression:

$$\underline{\boldsymbol{T}}(\underline{\nu}_{\text{out}}) \approx \underline{\underline{\boldsymbol{\Phi}}}\ \underline{\boldsymbol{\alpha}}_{\text{approx}} \tag{22}$$

where the prediction of the arrival times over all the sensors can be used to approximate the flow-front velocity as presented in section 3.4.

Once the real process finish, the real arrival times are measured for this new parameter set and they are used to enrich the hybrid twin, i.e., an approximation of the ignorance model is recalculated using this new data. In this way the hybrid twin is incrementally enriched online, delivering better predictions of reality as new real experiences of the RTM process under consideration continue to be made.

4 Numerical example

In the present section, numerical examples are presented in order to show the capability of the hybrid twin to better approximate a complex system in the context of the RTM manufacturing process, where the numerical model or virtual twin is enriched with sensor data. The numerical examples shown below are purely academic, so the "real experience" here is simply considered as a numerical simulation where a richer and more complex physical model is considered compared to the virtual twin which will be considered as a more simplified model.

The virtual twin considered consists of a 2D mold that have an injection pressure $p_{imp}(t)$ which can be constant or variable over time. The output pressure is considered to be null. The virtual twin consists of an uniform preform with a race-tracing zone, on the other hand the real experience consists on a more complex model where a preform with 3 different zones with different permeabilities are considered as well as the race-tracing zone. For both cases we considered a number of sensors in the x-direction $n_{s,x} = 5$ and $n_{s,y} = 5$ sensors in the y-direction. The virtual twin as well as the complex model are depicted in Figures 6 and 7 respectively, for both cases the sensor distribution is represented by red dots. The dimensions considered for both the numerical model and the "reality" are the same and given as follows:

$$d_1 = 0.8\ [m] \quad ; \quad d_2 = 0.35\ [m] \tag{23}$$

As presented in section 3.2.1, an offline calculation of the virtual twin must be first performed in function of defined parameters in order to fit as best as possible the virtual twin against some measured data. In this sense, this offline calculation is performed considering as parameters the permeabilities k_x and k_y in the x and y directions as well as the injection pressure, which is considered constant along the injection process. For this, a sparse sampling of these parameters are considered in order to apply the sPGD presented in section 6. The intervals considered for these parameters are given below:

$$\begin{aligned} k_x &= [0.0980, 0.9800] \times 10^{-8}\ [m^2] \\ k_y &= [0.0980, 0.9800] \times 10^{-9}\ [m^2] \\ p_{imp} &= [8, 32]\ [bar] \end{aligned} \tag{24}$$

It also considered a preform angle equals to 50 degrees (see section 2), a viscosity of the resin equals $\mu = 0.16\ [Pa\ s]$ and a high and constant permeability over the race-track zone (yellow zone of Figure 6) with a value of $2.25 \times 10{-}6\ [m^2]$. For the construction of the sPGD approximation of the virtual twin, 100 runs have been performed by selecting the parameters in (24) using the Latin hypercube sampling method.

The real experience emulation considered here corresponds to a more complex model. The permeabilities considered for the real experience emulation are taken equals to: The viscosity, the angle of the preforms and the permeabilities of the race-track

Table 1: Permeability constants for the emulation of the real experience ($[m^2]$)

$k_{x,1}$ (10^{-8})	$k_{x,2}$ (10^{-8})	$k_{x,3}$ (10^{-8})	$k_{y,1}$ (10^{-9})	$k_{y,2}$ (10^{-9})	$k_{y,3}$ (10^{-9})
0.7595	0.3185	0.0980	0.7595	0.3185	0.0980

zone are taken equals as the virtual twin model in order to simplify the example.

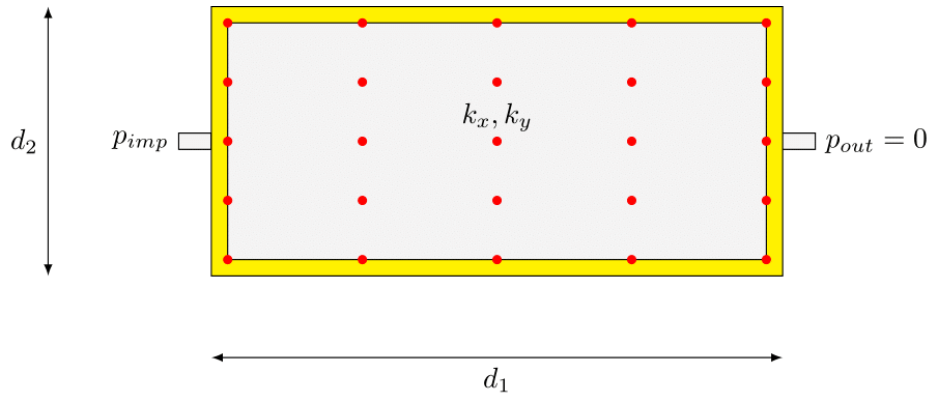


Figure 6: Simplified model considered.

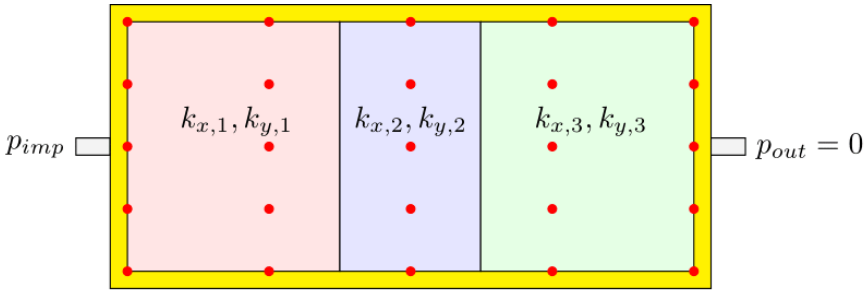


Figure 7: System representing the "reality". The zones with different permeabilities are illustrated with different colors and separated by lines.

As presented in section 3, the hybrid twin is constructed using data from the real experience. Due to this, the knowledge of 3 emulated RTM experiences for three different pressures is considered, where the permeabilities are assumed to be equal to those given in 1 for all the tests, and where the results of arrival times at the sensors are stored. The pressures considered are the followings:

$$p_{imp} = [8, 14, 20] \ [bar] \tag{25}$$

The first step for building the hybrid twin consists in determining the process parameters that best fit the virtual twin prediction to the real experience. To do so, only the first experience is considered. By applying the algorithm of section 3.3.1, the following parameters are obtained:

$$\begin{aligned} \hat{k}_x &= 0.2340 \times 10^{-8} \ [m^2] \\ \hat{k}_y &= 0.2090 \times 10^{-9} \ [m^2] \end{aligned} \tag{26}$$

where an error of 21[%] computed on the arrival times at the sensors is obtained. By using these identified parameters, the hybrid twin is constructed by modeling the gap between the virtual twin and the real experience emulation for the three experiences, following the strategy presented in section 3.4.

4.1 Prediction for parameters inside the HT definition

The hybrid twin constructed using the pressures values from equation (25) (which consists only on 3 measures and constants pressure values over time) can be used to predict the flow-front when the injection pressure varies within this domain of

definition in a continuous way. To illustrate this, consider a time-varying inlet pressure as shown in Figure 8.

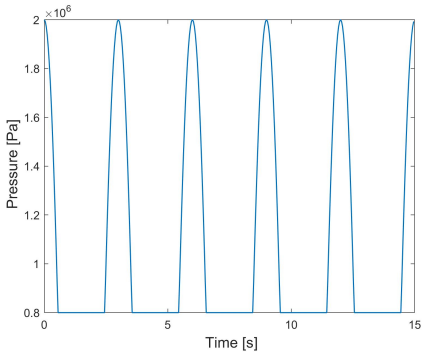


Figure 8: Pressure versus time considered.

For this case, the prediction of the flow-front using only the virtual twin is presented in Figure 9 (using only the virtual twin contribution in equation (16)) while the one using the constructed hybrid twin in Figure 10.

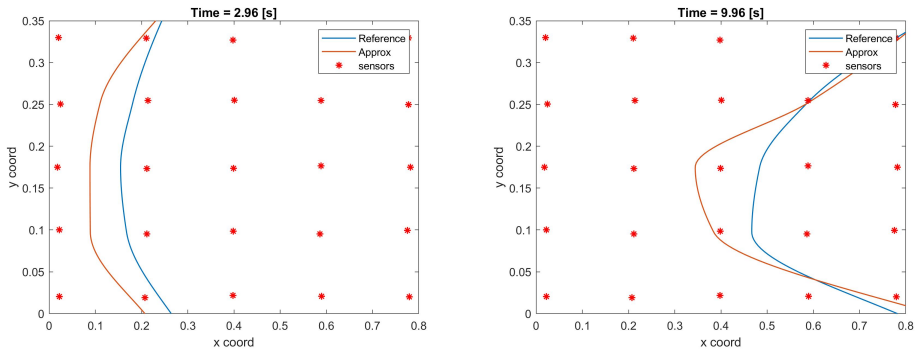


Figure 9: Comparison between the flow-front prediction of the virtual twin and the simulated experience for two different time instants.

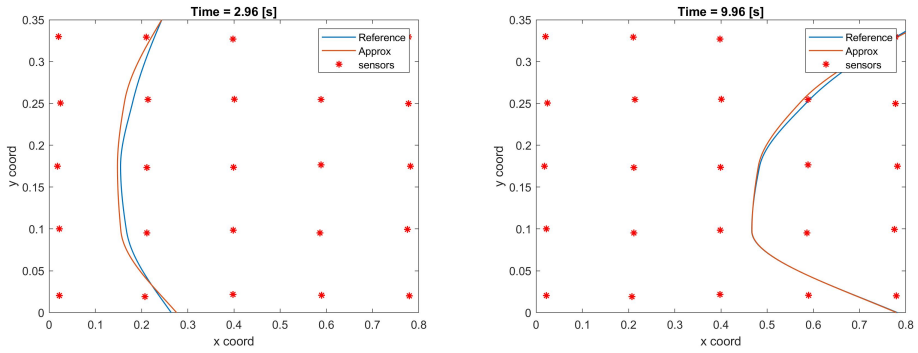


Figure 10: Comparison between the flow-front prediction of the hybrid twin and the simulated experience for two different time instants.

For the considered example, the hybrid twin achieved to approximate with an error of 2[%] the flow-front of the considered "real" process.

4.2 Prediction for parameters outside the HT definition

Now let us consider that the experience takes into account a time-varying pressure that does not belong to the pressure definition domain under consideration to build the hybrid twin. This pressure is presented in Figure 11, where the pressure considered surpass the interval of pressure (see (25)) used to build the hybrid twin. In fact the interval used here takes values in between 0 $[bar]$ and 32 $[bar]$. In this situation, resin arrival times for this new pressure cannot be predicted using the hybrid twin, so to give predictions in this situation the algorithm presented in section 3.5, which is based on Gappy POD, must be applied. For this situation, the predicted

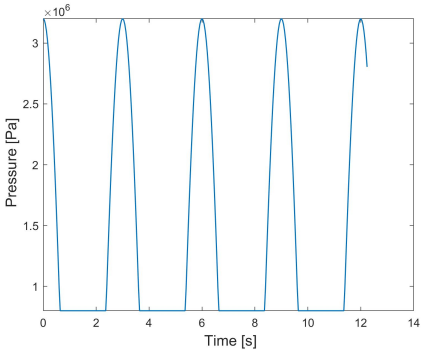
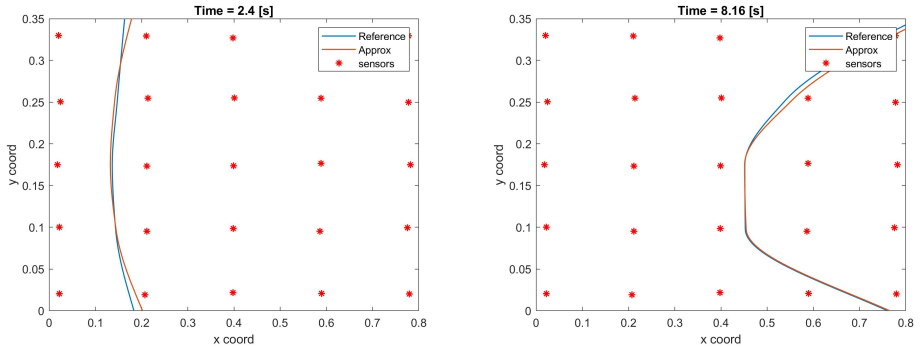


Figure 11: Pressure versus time considered.



(a) Flow-front of hybrid twin. (b) Flow-front of hybrid twin.

Figure 12: Comparison between the flow-front prediction when using the Gappy POD and the simulated experience for two different time instants.

flow-front achieved an error of 1.5[%] with respect to the real process. As can be seen from the above results, using a known reduced basis determined by previous arrival times measurements on the sensors allows to correctly predict the history of the flow-front for the new parameter set that lies outside the definition domain of the current hybrid twin.

5 Conclusions

This paper shows the main developments and numerical strategies that allow the construction of a hybrid twin applied to the resin transfer molding process. The hybrid twin is a numerical model capable of approximating a real process, which is built by making use of a physics-based model and real-world data measured by

sensors. The physics-based model used allows to capture much of the behavior of a real system as a function of any parameter considered, while the measured real-world data is used to model the difference between the physical model and reality, called the ignorance model, allowing the required amount of sensed information to be small.

Because the hybrid twin is built to be used during a real process, a real-time prediction of the physics-based model and the ignorance during the process must be given. For this reason, the physics-based model is approximated using the sparse-PGD model reduction technique and the process parameters are obtained by fitting in real-time this model against measured data using a highly optimized Levenberg-Marquardt algorithm.

The hybrid twin presented in this paper is capable of correctly predicting the RTM process when dealing with a complex system for the manufacture of mechanical parts when using synthetic data. However, its capabilities have yet to be tested in a real process, which is considered as a prospect for future works. Here, its performance was demonstrated by correctly approximating the resin flow-front inside a mold whose permeabilities changes along its domain by enriching a model that considered uniform permeabilities through a limited use of experimental setups and number of sensors.

Finally, the advantages of the hybrid twin applied to the RTM process open interesting perspectives. For instance, a natural one is to control the injection process by considering more than one inlet pressure, where the spread and shape of the flow-front could be controlled by changing the inlet pressures, which could considerably improve the quality of the manufactured parts and reduce their production time. Additionally, more physical phenomena of RTM such as curing during the filling process [39] could be taken into account to extend the capabilities of the hybrid twin and improve the manufacturing process in this aspect as well.

Acknowledgments

The European MORPHO project is gratefully acknowledged for funding this research activity.

Declarations

- Funding: This research was funded by the H2020 programme, under EU.3.4. - Societal Challenges - Smart, Green And Integrated Transport.
- Conflict of Interest: The authors declare that they have no conflict of interest.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Yes.
- Availability of data and materials: Not applicable.
- Code availability: Provided under request.

- Authors' contributions: Conceptualization, F.C., E.M., M.R, N.M.; methodology, S.R.; software, S.R.; supervision, F.C., E.M., M.R, N.M. All authors have read and agreed to the published version of the manuscript.

6 Appendix A: Approximation of sparse data; the sparse - PGD

In real-life situations, having access to abundant and structured data is not always possible. In this situation a direct application of the PGD method is unfortunately not possible. To solve this limitation, in [33] an extension of the PGD method was proposed to deal with unstructured data and overcome in this sense the curse of dimensionality. This method is called sparse-PGD.

To illustrate the main ideas on how to determine the PGD decomposition for the case of sparse data, an approximation of a function $f(\nu_1, \nu_2) \in \mathbb{R}^2 \rightarrow \mathbb{R}$ is considered here, where its value is known over some P sampling points. In this case, the PGD decomposition is simply determined by solving the following minimization problem:

$$\min \|u_m(\nu_1, \nu_2) - f(\nu_1, \nu_2)\|_2^2 \tag{27}$$

with $\|\bullet\|_2^2 = \int_\Omega (\bullet)^2 \, d\nu_1 d\nu_2$ and $\Omega = \Omega_{\nu_1} \times \Omega_{\nu_2}$. From the last expression, $u_m(\nu_1, \nu_2)$ corresponds to the PGD decomposition of $f(\nu_1, \nu_2)$, where m terms are considered, this is:

$$u_m(\nu_1, \nu_2) = \sum_{k=1}^{m} X_1^k(\nu_1) X_2^k(\nu_2) \tag{28}$$

with $X_1^k(\nu_1)$ and $X_2^k(\nu_2)$, $\forall k \in [1, ..., m]$ the separate variable functions to be determined. Here the PGD functions are not interpolated using, for example, the Finite Element Method (FEM) [3], in this case the functions used are globally defined over the whole interval considered for each variable. The functions that can be used could be for instance, Kriging interpolants functions [33] or radial basis functions [47], just to cite a few. In this context, the PGD functions for the 2D case are constructed as follows:

$$X_1^k(\nu_1) = \sum_{j=1}^{n_{cp}} N_j^k(\nu_1) a_{\nu_1,j}^k = (\underline{\boldsymbol{N}}_{\nu_1}^k)^T \underline{\boldsymbol{a}}_{\nu_1}^k \tag{29}$$

$$X_2^k(\nu_2) = \sum_{j=1}^{n_{cp}} N_j^k(\nu_2) a_{\nu_2,j}^k = (\underline{\boldsymbol{N}}_{\nu_2}^k)^T \underline{\boldsymbol{a}}_{\nu_2}^k \tag{30}$$

where $N_j^k(\nu_i)$ and $a_{\nu_i,j}^k$ denotes the shape functions and its respective nodal value for variable ν_i evaluated on the control point j at PGD mode k. If the shape functions are considered to be radial functions of multiquadric type [47], they are given as follows:

$$N_j(\nu_i) = \sqrt{c^2 \left[(\nu_i)_j - (\nu_i)\right]^2 + 1} \quad \text{for} \quad j \in [1, ..., n_{cp}] \text{ and } i \in [1, 2] \tag{31}$$

with $(\nu_i)_j$ the coordinate of the control point associated to variable ν_i and c a parameter that drives the shape of the function, chosen such as it allows a non over-fitted PGD approximation. These PGD functions are classically determined one after the other in an incremental way, also denoted Greedy process. That

is, one considers as known $m-1$ terms of the decomposition (28), and seeks the determination of the mode m, i.e., $u_m(\nu_1,\nu_2) = u_{m-1}(\nu_1,\nu_2) + X_1^m(\nu_1)X_2^m(\nu_2)$. In this case one reformulates (27) as follows:

$$\{X_1^m(\nu_1), X_2^m(\nu_2)\} = \underset{\{X_1^m(\nu_1),X_2^m(\nu_2)\}}{\arg\min} \|X_1^m(\nu_1)X_2^m(\nu_2) - f_{\text{res}}(\nu_1,\nu_2)\|_2^2 \tag{32}$$

with $f_{\text{res}}(\nu_1,\nu_2) = f(\nu_1,\nu_2) - u_{m-1}(\nu_1,\nu_2)$. By minimizing the expression (32) yields: $\forall w^*(\nu_1,\nu_2)$,

$$\int_\Omega w^*(\nu_1,\nu_2)\left[X_1^m(\nu_1)X_2^m(\nu_2) - f_{\text{res}}(\nu_1,\nu_2)\right] d\nu_1 d\nu_2 = 0 \tag{33}$$

However since the information is just known at P sampling points $((\nu_1)_i,(\nu_2)_i), \forall i \in [1,...,P]$, the test function $w^*(\nu_1,\nu_2)$ is expressed as a set of Dirac delta functions collocated at these points, this is:

$$\begin{aligned} w^*(\nu_1,\nu_2) &= u^*(\nu_1,\nu_2)\sum_{i=1}^{P}\delta((\nu_1)_i,(\nu_2)_i) \\ &= \left[{X_1}^*(\nu_1)X_2^m(\nu_2) + X_1^m(\nu_1){X_2}^*(\nu_2)\right]\sum_{i=1}^{P}\delta((\nu_1)_i,(\nu_2)_i) \end{aligned} \tag{34}$$

where $\delta((\nu_1)_i,(\nu_2)_i)$ here denotes the Dirac delta function evaluated at points $((\nu_1)_i,(\nu_2)_i)$. By replacing $w^*(\nu_1,\nu_2)$ by its sparse version of equation (34) into (33), and developing the expression yields the problems that must be solved to obtain the PGD functions:

Compute $X_1^m(\nu_1)$ by solving:

$$\int_\Omega {X_1}^*(\nu_1)\sum_{i=1}^{P}\left[X_1^m(\nu_1)X_2^m(\nu_2)^2 - X_2^m(\nu_2)f_{\text{res}}(\nu_1,\nu_2)\right]\delta((\nu_1)_i,(\nu_2)_i)\; d\nu_1 d\nu_2 = 0 \tag{35}$$

Compute $X_2^m(\nu_2)$ by solving:

$$\int_\Omega {X_2}^*(\nu_2)\sum_{i=1}^{P}\left[X_2^m(\nu_2)X_1^m(\nu_1)^2 - X_1^m(\nu_1)f_{\text{res}}(\nu_1,\nu_2)\right]\delta((\nu_1)_i,(\nu_2)_i)\; d\nu_1 d\nu_2 = 0 \tag{36}$$

Equations (35) and (36) should be solved alternatively following a fixed-point strategy, where for the resolution of equation (35) one considers known the value of function $X_2^m(\nu_2)$, and for the resolution of (36) one considers $X_1^m(\nu_1)$ known. To illustrate this resolution, but only for the determination of $X_1^m(\nu_1)$, lets define:

$$\begin{aligned} \underline{\underline{\boldsymbol{A}}}^{kl}_{(\nu_1)_i} &= \underline{\boldsymbol{N}}^k_{\nu_1}((\nu_1)_i)\otimes\underline{\boldsymbol{N}}^l_{\nu_1}((\nu_1)_i) \\ \underline{\underline{\boldsymbol{A}}}^{kl}_{(\nu_2)_i} &= \underline{\boldsymbol{N}}^k_{\nu_2}((\nu_2)_i)\otimes\underline{\boldsymbol{N}}^l_{\nu_2}((\nu_2)_i) \end{aligned} \tag{37}$$

where $\otimes$ denotes a Kronecker tensor product. By using the operators defined in (37) and approximations (29) and (30), one obtains the following system of equations needed to be solved to compute the discretized DOFs related to $X_1^m(\nu_1)$ ($\underline{\boldsymbol{a}}^m_{\nu_1}$).

$$\underline{\underline{\boldsymbol{M}}}_{\nu_1}\underline{\boldsymbol{a}}^m_{\nu_1} = \underline{\boldsymbol{f}}_{\nu_1} \tag{38}$$

where the matrix and vector are respectively given as:

$$\begin{aligned}\underline{\underline{\boldsymbol{M}}}_{\nu_1} &= \sum_{i=1}^{P}\left[(\underline{\boldsymbol{a}}_{\nu_2}^m)^T\,\underline{\underline{\boldsymbol{A}}}_{(\nu_2)_i}^{mm}\,\underline{\boldsymbol{a}}_{\nu_2}^m\right]\underline{\underline{\boldsymbol{A}}}_{(\nu_1)_i}\\ \underline{\boldsymbol{f}}_{\nu_1} &= \sum_{i=1}^{P} f_{\text{res}}((\nu_1)_i,(\nu_2)_i)\,\underline{\boldsymbol{N}}_{\nu_1}^m((\nu_1)_i)\,\underline{\boldsymbol{N}}_{\nu_2}^m((\nu_2)_i)^T\,\underline{\boldsymbol{a}}_{\nu_2}^m\end{aligned} \tag{39}$$

The extension of the above method to the case of a multidimensional function is straightforward and follows the same methodology, for this case one has:

$$\min \|u_m(\nu_1,\nu_2,\dots,\nu_n) - f(\nu_1,\nu_2,\dots,\nu_n)\|_2^2 \tag{40}$$

where:

$$u_m(\nu_1,\nu_2,\dots,\nu_n) = \sum_{k=1}^{m} X_1^k(\nu_1)X_2^k(\nu_2)\,\dots\,X_n^k(\nu_n) \tag{41}$$

with $\|\bullet\|_2^2 = \int_\Omega(\bullet)^2\,d\nu_1 d\nu_2\,\dots\,d\nu_n$, where for this case $\Omega = \Omega_{\nu_1}\times\Omega_{\nu_2}\times\dots\times\Omega_{\nu_n}$. For this case the same resolution procedure by the fixed-point technique must be applied. The determination of the nodal values of the m PGD function corresponding to parameter ν_q with $q\in[1,\dots,n]$ is simply given as:

$$\underline{\underline{\boldsymbol{M}}}_{\nu_q}\,\underline{\boldsymbol{a}}_{\nu_q}^m = \underline{\boldsymbol{f}}_{\nu_q} \tag{42}$$

where the respective matrix and vector are computed as follows:

$$\begin{aligned}\underline{\underline{\boldsymbol{M}}}_{\nu_q} &= \sum_{i=1}^{P}\underline{\underline{\boldsymbol{A}}}_{(\nu_q)_i}^{mm}\prod_{\forall j\neq q}\left[(\underline{\boldsymbol{a}}_{\nu_j}^m)^T\,\underline{\underline{\boldsymbol{A}}}_{(\nu_j)_i}^{mm}\,\underline{\boldsymbol{a}}_{\nu_j}^m\right]\\ \underline{\boldsymbol{f}}_{\nu_q} &= \sum_{i=1}^{P} f_{\text{res}}(\underline{\nu}_i)\,\underline{\boldsymbol{N}}_{\nu_q}^m((\nu_q)_i)\prod_{\forall j\neq q}\underline{\boldsymbol{N}}_{\nu_j}^m((\nu_j)_i)^T\,\underline{\boldsymbol{a}}_{\nu_j}^m\end{aligned} \tag{43}$$

with $f_{\text{res}}(\nu_1,\nu_2,\dots,\nu_n) = f(\nu_1,\nu_2,\dots,\nu_n) - u_{m-1}(\nu_1,\nu_2,\dots,\nu_n)$.

References

[1] Lihwa Fong and SG Advani. Resin transfer molding. In *Handbook of Composites*, pages 433–455. Springer, Boston, MA, 1998.

[2] Kevin D Potter. The early history of the resin transfer moulding process for aerospace applications. *Composites Part A: applied science and manufacturing*, 30(5):619–621, 1999.

[3] Olek C Zienkiewicz and Robert Leroy Taylor. *The finite element method for solid and structural mechanics*. Elsevier, Amsterdam, 2005.

[4] Gordon D Smith, Gordon D Smith, and Gordon Dennis Smith Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, Oxford, 1985.

[5] Francisco Chinesta, Elias Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, and Fouad El Khaldi. Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of computational methods in engineering*, 27(1):105–134, 2020.

[6] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., New York, 2010.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, MA, USA, 2016.

[8] M Arif Wani, Farooq Ahmad Bhat, Saduf Afzal, and Asif Iqbal Khan. *Advances in deep learning*. Springer, Singapore, 2020.

[9] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, Birmingham, UK, 2017.

[10] Beatriz Moya, David González, Icíar Alfaro, Francisco Chinesta, and Elias Cueto. Learning slosh dynamics by means of data. *Computational Mechanics*, 64(2):511–523, 2019.

[11] Agathe Reille, Nicolas Hascoet, Chady Ghnatios, Amine Ammar, Elias Cueto, Jean Louis Duval, Francisco Chinesta, and Roland Keunings. Incremental dynamic mode decomposition: A reduced-model learner operating at the low-data limit. *Comptes Rendus Mécanique*, 347(11):780–792, 2019.

[12] Quercus Hernández, Alberto Badías, David González, Francisco Chinesta, and Elías Cueto. Structure-preserving neural networks. *Journal of Computational Physics*, 426:109950, 2021.

[13] Abel Sancarlos, Morgan Cameron, Jean-Marc Le Peuvedic, Juliette Groulier, Jean-Louis Duval, Elias Cueto, and Francisco Chinesta. Learning stable reduced-order models for hybrid twins. *Data-Centric Engineering*, 2, 2021.

[14] Abel Sancarlos, Morgan Cameron, Andreas Abel, Elias Cueto, Jean-Louis Duval, and Francisco Chinesta. From rom of electrochemistry to ai-based battery digital and hybrid twin. *Archives of Computational Methods in Engineering*, 28(3):979–1015, 2021.

[15] Victor Champaney, Francisco Chinesta, and Elias Cueto. Engineering empowered by physics-based and data-driven hybrid models: A methodological overview. *International Journal of Material Forming*, 15(3):1–14, 2022.

[16] Pierre Ladevèze. Sur une famille d'algorithmes en mécanique des structures. *Comptes-rendus des séances de l'Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l'univers, sciences de la terre*, 300(2):41–44, 1985.

[17] Pierre-Alain Boucard and Pierre Ladevèze. A multiple solution method for non-linear structural mechanics. *Mechanical Engineering*, 50:317–328, 1999.

[18] Pierre Ladevèze, J-C Passieux, and David Néron. The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21-22):1287–1296, 2010.

[19] Nicolas Relun, David Néron, and Pierre-Alain Boucard. Multiscale elastic-viscoplastic computational analysis: a detailed example. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 20(7-8):379–409, 2011.

[20] Nicolas Relun, David Néron, and Pierre-Alain Boucard. A model reduction technique based on the pgd for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51(1):83–92, 2013.

[21] David Néron, Pierre-Alain Boucard, and Nicolas Relun. Time-space pgd for the rapid solution of 3d nonlinear parametrized problems in the many-query context. *International Journal for Numerical Methods in Engineering*, 103(4):275–292, 2015.

[22] S Rodriguez, David Néron, P-E Charbonnel, Pierre Ladevèze, and G Nahas. Non incremental latin-pgd solver for non-linear vibratoric dynamics problems. In *14ème Colloque National en Calcul des Structures, CSMA 2019*, 2019.

[23] Sebastian Rodriguez Iturra. *Abaques virtuelle pour le génie parasismique incluant des parametres associes au chargement*. PhD thesis, Université Paris-Saclay, 2021.

[24] Amine Ammar, Béchir Mokdad, Francisco Chinesta, and Roland Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of non-Newtonian fluid Mechanics*, 139(3):153–176, 2006.

[25] Francisco Chinesta, Amine Ammar, and Elías Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational methods in Engineering*, 17(4):327–350, 2010.

[26] Francisco Chinesta and Elias Cueto. *PGD-based modeling of materials, structures and processes*. Springer, Switzerland, 2014.

[27] Sergio Torregrosa, Victor Champaney, Amine Ammar, Vincent Herbert, and Francisco Chinesta. Surrogate parametric metamodel based on optimal transport. *Mathematics and Computers in Simulation*, 194:36–63, 2022.

[28] Amine Ammar, Francisco Chinesta, Elías Cueto, and Manuel Doblaré. Proper generalized decomposition of time-multiscale models. *International Journal for Numerical Methods in Engineering*, 90(5):569–596, 2012.

[29] Rubén Ibáñez, Amine Ammar, Elías Cueto, Antonio Huerta, Jean-Louis Duval, and Francisco Chinesta. Multiscale proper generalized decomposition based on the partition of unity. *International Journal for Numerical Methods in Engineering*, 120(6):727–747, 2019.

[30] Angelo Pasquale, Amine Ammar, Antonio Falcó, Simona Perotto, Elías Cueto, Jean-Louis Duval, and Francisco Chinesta. A separated representation involving multiple time scales within the proper generalized decomposition framework. *Advanced Modeling and Simulation in Engineering Sciences*, 8(1):1–22, 2021.

[31] J-C Passieux, Pierre Ladevèze, and David Néron. A scalable time–space multiscale domain decomposition method: adaptive time scale separation. *Computational Mechanics*, 46(4):621–633, 2010.

[32] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, New York, 2007.

[33] Rubén Ibáñez, Emmanuelle Abisset-Chavanne, Amine Ammar, David González, Elías Cueto, Antonio Huerta, Jean Louis Duval, and Francisco Chinesta. A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. *Complexity*, 2018, 2018.

[34] Abel Sancarlos, Victor Champaney, Jean-Louis Duval, Elias Cueto, and Francisco Chinesta. Pgd-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit. *arXiv preprint arXiv:2103.05358*, 2021.

[35] Abel Sancarlos, Elias Cueto, Francisco Chinesta, and Jean-Louis Duval. A novel sparse reduced order formulation for modeling electromagnetic forces in electric motors. *SN Applied Sciences*, 3(3):1–19, 2021.

[36] Giacomo Quaranta, Emmanuelle Abisset-Chavanne, Francisco Chinesta, and Jean-Louis Duval. A cyber physical system approach for composite part: From smart manufacturing to predictive maintenance. In *AIP conference proceedings*, volume 1960, page 020025. AIP Publishing LLC, 2018.

[37] Fernando Sánchez, JA García, Francisco Chinesta, Ll Gascón, C Zhang, Z Liang, and Biao Wang. A process performance index based on gate-distance and incubation time for the optimization of gate locations in liquid composite molding processes. *Composites Part A: Applied Science and Manufacturing*, 37(6):903–912, 2006.

[38] JA Garcıa, Ll Gascón, and Francisco Chinesta. A fixed mesh numerical method for modelling the flow in liquid composites moulding processes using a volume of fluid technique. *Computer methods in applied mechanics and engineering*, 192(7-8):877–893, 2003.

[39] Fernando Sanchez, Juan Antonio Garcia, Ll Gascón, and Francisco Chinesta. Towards an efficient numerical treatment of the transport problems in the resin transfer molding simulation. *Computer methods in applied mechanics and engineering*, 196(21-24):2300–2312, 2007.

[40] Giacomo Quaranta. *Efficient simulation tools for real-time monitoring and control using model order reduction and data-driven techniques*. Thesis, École centrale de Nantes ; Universitat politécnica de Catalunya, September 2019.

[41] Ch Ghnatios, Françoise Masson, Antonio Huerta, Adrien Leygue, Elías Cueto, and Francisco Chinesta. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213:29–41, 2012.

[42] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, Philadelphia, Penn., 1996.

[43] Henri P Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, 19, 2019.

[44] Kari Karhunen. Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34, 1946.

[45] M Loeve. Fonctions aléatoires du second ordre. *Lévy, Processus Stochastiques et Mouvement Brownien, Gauthier-Villars, Paris*, 1948.

[46] Richard Everson and Lawrence Sirovich. Karhunen–loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[47] Gregory E Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, Hackensack, NJ, 2007.