POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Digital twins of structural systems using Dynamic Bayesian Networks and Reduced Order Models

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Roberto Valendino**

Student ID: 967407
Advisor: Prof. Andrea Manzoni
Co-advisor: Matteo Torzoni
Academic Year: 2021-2022

Abstract

Digital twin technologies aim to create a digital representation of a physical phenomena. These techniques has become very popular in recent years and are nowadays applied in several industrial contests. This work shows how dynamic Bayesian networks (DBNs) can be used to represent the interactions between the Digital and the Physical state and how they can be used to carry out control and prediction analysis. In particular, I will explore how dynamic Bayesian networks can be used to perform structural health monitoring (SHM), and how these information can be used to support decisions regarding its maintenance and repair. I will also examine the computational capabilities and accuracy of dynamic Bayesian networks, and propose a method to improve these capabilities by exploiting a reduced order method (ROM) to build the input dataset of the DBN when dealing with problems related with structural mechanic. Considering different case studies, this work provides a comprehensive overview of the power and versatility of dynamic Bayesian network to enable digital twins for SHM purposes.

Keywords: Digital Twin, Dynamic Bayesian Network, Inverse problem, Structural health monitoring, Reduced Order Models

Abstract in lingua italiana

Le tecnologie dei gemelli digitali mirano a creare una rappresentazione digitale di un fenomeno fisico. Queste tecniche sono diventate molto popolari negli ultimi anni e sono oggi applicate in diversi contesti industriali. Questo lavoro mostra come le reti bayesiane dinamiche (DBN) possano essere utilizzate per rappresentare le interazioni tra lo stato digitale e quello fisico e con lo scopo ultimo di effettuare analisi di controllo e previsione. In particolare, la tesi mira ad esplorare le reti bayesiane dinamiche per il monitoraggio della salute strutturale (SHM) e come queste informazioni possano essere sfruttate per supportare le decisioni relative alla manutenzione e alla riparazione di una struttura. Verranno inoltre esaminate le capacità computazionali e l'accuratezza delle reti bayesiane dinamiche e proporrò un metodo per migliorare queste capacità sfruttando un metodo di ordine ridotto (ROM) per costruire il set di dati di input del DBN quando si affrontano problemi legati alla meccanica strutturale. Considerando diversi casi di studio, questo lavoro fornisce una panoramica della potenza e della versatilità delle reti bayesiane dinamiche per la creazione di gemelli digitali per scopi di SHM.

Parole chiave: Gemello digitale, Rete Bayesiana dinamica, Problema inverso, Monitoraggio della salute strutturale, Modelli di ordine ridotto

Contents

Introduction

Digital twin methods aim to built a virtual representation of a physical object or process capable of collecting information from the real environment to represent, validate, and simulate the present and future behavior of the corresponding physical asset. For the purpose of closely characterize the operations of the original physical asset, the digital twin must be kept synchronized through the assimilation of observational data, and update of the parameters involved in the digital state, which characterize the variability in the physical asset. The updated digital state thus enables to predict the expected evolution of digital state and the associated uncertainty, as well as to inform an optimal planning of control inputs feeding back to the physical system. Digital twins are nowadays being used in several industries such as as manufacturing, healthcare, and transportation, [2, 5, 11, 21], gaining more and more importance and attention.

In this thesis, I focus on the use of Dynamic Bayesian networks (DBNs), that are probabilistic graphical models, as a mathematical foundation for enabling predictive Digital Twins at scale. Bayesian networks provide a framework for modeling the relationship between the physical asset, the digital twin and all the variables which describe their interactions. This approach offers several advantages over traditional digital twins such as the possibility of incorporating the uncertainty on the estimate of the parameters along all the duration of the considered simulation.

Among the others applications, digital twin have been proposed as a feasible solution to monitor the health state of a structure. Structural health monitoring (SHM) refers to strategies aimed at detecting changes and damages in structures from sensed data, thus allowing to promptly implement maintenance actions before the occurrence of major failures. As we can see in [1, 7, 8, 24, 27] in recent years digital twins have started to play an important role in the field of structural health monitoring. In this thesis, I propose the use of dynamic Bayesian network framework introduced in [14] to monitor the health state of a concrete beam and of a concrete frame in different situations.

I also focus on reduced order models (ROMs), which are mathematical models that aim to reduce the computational cost of a high-fidelity full order model (FOM) [10, 15, 16]. In particular reduced order models will be employed to compute faster but accurate input datasets of the Dynamic Bayesian networks. When dealing with structural problems the physical state can be modeled in terms of partial differential equations. Figure 1 shows the interactions among these techniques and how the simulation process considered in this thesis actually works.

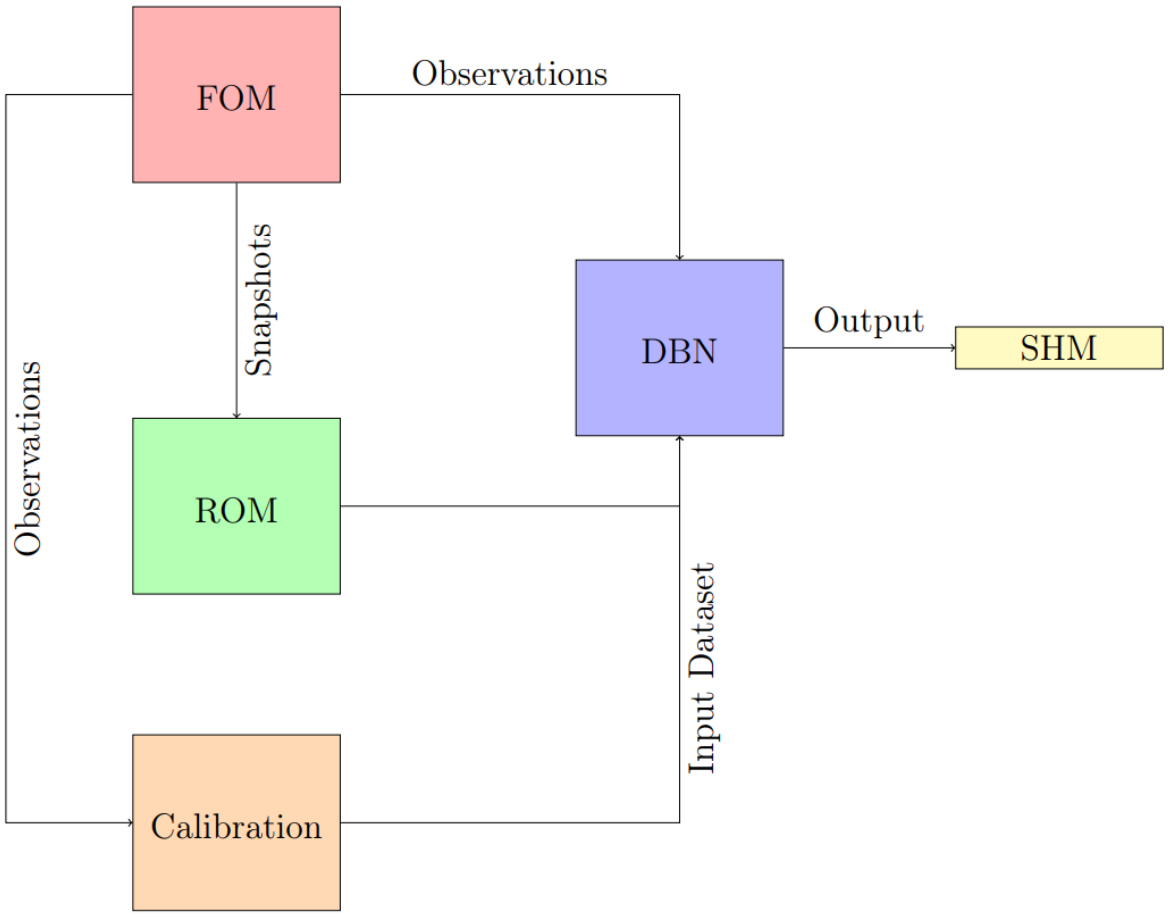


Figure 1: A schematic representation of the main methodologies involved in the considered digital twin framework.

The observational data collected from a sensing system deployed on the structure are assumed to be simulated using a high-fidelity full-order model (FOM) of the monitored structure. On the other hand, the dataset provided to the DBN to perform the digital state updating is instead generated by exploiting a faster reduced-order model (ROM). In the thesis I will explain how these relationships and processes works. The thesis is

organized as follows:

- Chapter 1 introduces and describes dynamic Bayesian networks and reduced order models, and their theoretical background.
- Chapter 2 describes the first test case I considered where dynamic Bayesian networks are used to describe the evolution of the health state of a concrete beam that can crack.
- Chapter 3 describes the second test case I considered where ROMs have been employed to create the input dataset of the dynamic Bayesian network.
- Chapter 4 describes the third test case I considered where dynamic Bayesian networks have been employed in a time-dependent framework to make structural health monitoring of a four-stories frame.
- Chapter 5 finally reports the main conclusions of this work and propose some ideas for possible future developments.

1 | Digital twinning

In this chapter I will introduce the concept of digital twin by showing the relevance of these methods in engineering (section 1.1). Then I will analyze the methods introduced in [14] based on dynamic Bayesian networks (section 1.2) and introduce reduced order models showing how effective these mathematical tools are in providing new data for digital twin analysis (section 1.3). In particular, I focus my attention on the Galerkin Reduces Basis method that I will use in the analyses carried out throughout of the thesis.

1.1. Digital twins

The first definition of digital twin method was given in [9] where *"A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin."* The latter definition is specific to digital twins used in the aerospace domain. In general, a digital twin is a computational model which aim to represent a physical phenomenon in the digital space for the purpose of closely characterizing the operations of the original physical asset. The physical and the digital twins can be represented as two dynamical systems which evolve during a time interval $(0, T)$. During the simulation interval, the digital twin must be updated; to this aim, it receives information from its physical counterpart and it becomes a realistic copy of the original physical asset. The updated digital state allows the users to make prediction and control analysis on the physical phenomena; for this reason, it is important to find a method to suitably represent changes in the physical state in real time.

The digital twin methods are gaining more and more importance in a wide range of applications such as automotive, aerospace engineering, energy and civil engineering. For example, as we can see in [26], digital twins can be used to estimate the capacity fade of a battery and choosing the optimal timing to remove the battery. Another important application is the one concerning the fault diagnosis and the monitoring of damage and health of a structure, as we can see in [18, 27, 28].

Two different types of digital twinning methods can be found in the literature: those

characterized by three dimensions and those characterized by five dimensions. The three-dimensional digital twins are characterized by a physical state (PS), a digital state (DS) and by the exchange of information and data between them. The five-dimensional digital twins are more complex models, that can be described by the following relationship

$$DS = f(FS, DS, P2V, V2P, OPT). \tag{1.1}$$

This equation represent the relationship between the five dimensions which characterize the digital twin: the physical state (PS), the digital state (DS), the updating on the digital model based on the information coming from the physical state ($P2V$), the prediction process which can be used for control the physical state ($V2P$) and the optimization process (OPT) which optimize all the functionalities of the other dimension.

1.2. Probabilistic graphical model for digital twin

In [14], a new method based on dynamic Bayesian graphical networks has been introduced, which allows to accurately describe the information exchange that takes place between the physical asset and the digital model; in this section I will describe the main innovative ideas and features that allow to describe this new digital twin model. Starting from the equation (1.1) and setting the goal of representing a physical phenomena, which evolves in a time interval $(0, T)$, the relevant quantities which characterize a digital twin model could be summarized as:

- S_t, the physical state at time t, which is only indirectly observable via observational data O_t;
- D_t, the digital state at time t;
- U_t, collecting the actions and the decisions which influence the physical state;
- Q_t, encoding the quantities of interest (QoI) computed from the updated digital state;
- R_t, representing the reward quantifying the performance of the asset-twin system.

Treating all these quantities as random variables, it is possible to build a probabilistic graphical model (PGM) adding a set of decision nodes that allow to take control actions based on estimated quantities. Such a graph aims at representing the interactions between the asset-twin system and its evolution over time. The PGM we will rely on, inspired by [14], is a dynamic Bayesian network. Dynamic Bayesian networks are probabilistic graphical models that represent a set of variables and their conditional dependencies using a

directed acyclic graph. These kind of graphs also relate variables to each other over adjacent time steps, [3]. In the proposed PGM, the nodes of the graph represent the random variables of interest at a given time, and the arcs between the nodes represent the dependence relationship among the random variables, so the nodes not connected represent variables which are conditionally independent. It is important to note that the variables referred to a time step are also related with the ones referring to adjacent time-steps. To construct this graph, first the simulation time interval $(0, T)$ needs to be discretized; when a new observation o_t from the physical state is collected, a node, representing the observation $O_t = o_t$, the digital state D_t, the QoI Q_t and the reward R_t, are added to the graph. A decision node, which represents the control action U_t, is also added. As said before, the link between these new nodes represents a conditional probability; in particular, based on the structure of the graph, we can define these probabilities:

$$\phi_t^{dynamics} = P(D_t|D_{t-1}, U_{t-1}), \tag{1.2}$$

$$\phi_t^{QoI} = P(Q_t|D_t), \tag{1.3}$$

$$\phi_t^{evaluation} = P(R_t|Q_t, D_t, O_t = o_t, U_t = u_t), \tag{1.4}$$

$$\phi_t^{assimilation} = P(O_t|D_t), \tag{1.5}$$

$$\phi_t^{control} = P(U_t|D_t, Q_t). \tag{1.6}$$

In these equations, uppercase letters are used to define the random variables represented in the graph; corresponding lowercase letters are used to represent the values that these random variables can take. Therefore, at each timestep of the simulation, we are going to update the pre-existing graph in the way described above. In figure 1.1, the Bayesian dynamic network obtained from the initial time of the simulation , $t = 0$, to the current time, $t = t_c$, and to future to the prediction horizon, $t = t_p$, is represented. From the figure, it can be seen that the resulting graph is acyclic and directed, in fact it consists of vertices and edges, with each edge directed from a vertex to another vertex, such that following these directions will never form a closed loop.

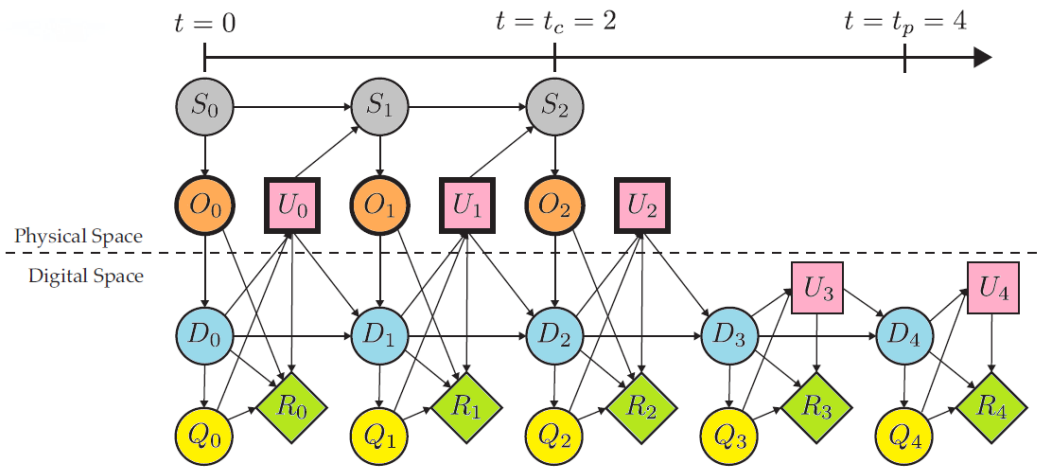


Figure 1.1: Adopted DBN (figure taken from [14]): nodes with bold outline represent the observed quantities (sensed data and enacted actions), while nodes with thin outline represent estimated quantities. Edges represent the conditional dependence between random variables.

Once the graph topology has been established, the belief about unobserved variables is updated and propagated using the loopy belief propagation algorithm, see e.g. [17]. The loopy belief propagation algorithm, also know as sum-product algorithm, consists of two steps: first, we need to compute the joint distributions of the random variables in the graph, then, we have to compute their marginals. In this way, once a new observation o_t is collected, the posterior distribution for each random variable can be updated. In this algorithm, the joint distribution is computed as

$$p(D_0, ..., D_{t_p}, Q_0, ..., Q_{t_p}, R_0, ..., R_{t_p}, U_{t_c+1}, ..., U_{t_p} | o_0, ..., o_{t_p}, u_0, ..., u_{t_c}) = \prod_{t=0}^{t_c} [\phi_t^{update} \phi_t^{QoI} \phi_t^{evaluation}], \tag{1.7}$$

where t_c is the current time-step and ϕ_t^{update} are defined:

$$\phi_t^{update} = P(D_t | D_{t-1}, U_{t-1} = u_{t-1}, O_t = o_t) \propto \phi_t^{dynamics} \phi_t^{assimilation}. \tag{1.8}$$

Using the properties of the dynamic Bayesian probabilistic graphical model and the sum-

product algorithm, it is possible to predict the value of the characteristic random variables in future time-step. In this case, the joint distribution is defined as

$$p(D_0, ..., D_{t_p}, Q_0, ..., Q_{t_p}, R_0, ..., R_{t_p}, U_{t_c+1}, ..., U_{t_p} | o_0, ..., o_{t_p}, u_0, ..., u_{t_c}) = \prod_{t=0}^{t_p} [\phi_t^{dynamics} \phi_t^{QoI} \phi_t^{evaluation}] \prod_{t=0}^{t_c} \phi_t^{assimilation} \prod_{t=t_c+1}^{t_p} \phi_t^{control} \tag{1.9}$$

where t_c is the current time step, t_p is the prediction horizon. Computing as before the marginal probabilities, it is possible to predict the future digital states, quantities of interest and control actions; this prediction is very important when this graph is used as a digital twin model. We remark that to compute the joint distribution we need to compute the decision output u_t; in section 1.2.1 we will discuss on how compute it. More details on loopy belief propagation algorithm are given in Appendix A.

1.2.1. Decision Node

Starting from the Figure 1.1, between the current time-step and the prediction one, we can look to our model as a partially observable Markov decision process ($POMDP$). This kind of processes provides an extension of the fully observable Markov decision processes ($MDPs$). In a MDP, the state is fully observable, while in a $POMDP$ the state is hidden, it is observable only in an indirect way. In particular, the only thing that we know of a state s is its emission probability $P(o|s)$ where o is a possible observation of the state s. As said before, at every time step, a control-input u_t is chosen in such a way that it maximises an expected future reward. This problem is a reinforcement learning problem, which has as solution a policy π defined as:

$$u_t = \pi(p(D_0, ..., D_t, Q_0, ..., Q_t | o_0, ..., o_t, u_0, ..., u_{t-1})). \tag{1.10}$$

Among all these possible policies, the agent wants to find the optimal one which allows him to maximize a reward function, R_t. The optimal policy, π^*, can be found solving an optimization problem of the form

$$\pi^* = \arg\min_{\pi} \sum_{t=t_c+1}^{t_p} \gamma^{t-t_c-1} E_\pi[R_t], \tag{1.11}$$

where $E_\pi[\cdot]$ represents the expetation when the policy π is followed and $\gamma \in [0, 1]$ is a discount factor. In particular, if $\gamma = 0$ the agent is only interested on the immediate reward, while if $\gamma \to 1$ the agent only cares about the maximization of future time reward.

Typically, it is very hard to solve this kind of problems; for this reason we approximate it to a fully observable MDP. In particular, every state is approximated with its maximum a posteriori estimate. In this way, the policy is now defined as $\pi(d^*, q^*)$, where d^*, q^* are the maximum a posteriori estimates for the digital twin state and for the quantity of interest state respectively. This approximation also allows to use the value-iteration algorithm to find the π^*; for more details about this algorithm, see [6]. Once the optimal policy is found, it is possible to choose at each time-step the control action u_t to take; in particular, this also allows to compute $\phi_t^{control}$ in an easy way:

$$\phi_t^{control} = \begin{cases} 1 & \text{if } \pi^*(d_t, q_t) = u_t, \\ 0 & \text{otherwise.} \end{cases} \tag{1.12}$$

1.2.2. Digital twin at scale

As discussed in [14], the strength of probabilistic graphical networks lies in the fact that they make possible to move from one-off digital twin to robust digital twin implementations at scale. As said before the crucial idea of all digital twin models is to provide virtual replicas of physical systems. This kind of models consider all the different variables which characterize the physical state and all the interactions between the physical and the digital assets. The digital twin simulation is carried out in two different phases the first one is the calibration phase, in which the parameters describing the digital state are calibrated, to closely reflect the physical asset. The calibration phase is included in the structure of the graph; this allows us to incorporate the uncertainty on the calibrated parameters in the rest of the simulation. The calibration phase allows the digital twin to be used at scale provided that each system analyzed is in principle calibrated; this is the real advantage of digital twin based on DBN, [14]. Once the calibration phase is performed starts the second phase, referred to as the operational phase, starts, during which the digital state is continuously updated through the assimilation of observational data, and adopted to compute quantities of interest and to choose the most appropriate control input.

1.3. Reduced order models

To perform the digital state updating, dynamic Bayesian networks require an input dataset. In this thesis, I investigate how this digital twin model behaves if instead of creating the input dataset using observations from real instruments, as was done in [14], the input dataset is generated using numerical simulations. Usually, obtaining a numerical

solution of a given system is a computationally expensive process, so it is important to rely on new techniques capable to overcome this drawback. Reduced order models (ROMs) encompass all those techniques used to reduce the computational complexity of a given numerical method. These methods are are very popular when dealing with parameterized partial differential equation (PDEs). In particular, the final goal of these methods is to find an approximate solution for any new parameter instance, however entailing a much smaller computational cost with respect to the one required by a full order model (FOM), guaranteeing that the error between the high fidelity solution and the ROM one is smaller than a given threshold. The most important class of ROMs in the contest of parametrised PDEs is provided by the reduced basis methods. Their basic idea is to construct a small finite-dimension subspace where to seek the solution of the PDE, and to represent the approximated solution as a linear combination of the basis functions spanning this subspace.
Another possible application of ROMs is to augment the data required as input by the Bayesian network; a process that usually goes under the name of data augmentation used to refer to all the techniques used for increasing the amount of data by creating synthetic data and adding them to existing data. Usually, it is expensive to build out a dataset, since the data usually come from sensors or high-fidelity numerical simulations. Data augmentation is used when the input data set is small and one would increase its dimension to improve the generalization performances of a model which receives in input this dataset. In this contest, data augmentation consists of adding to the starting data some data obtained easily from already existing ones, to increase the dataset used by the dynamic Bayesian network needs so that more accurate results can be obtained with a low computational cost. In particular the dataset needed by the DBN contains sensor measurement for each possible value of the digital space D_t. I will explore the idea to increase this dataset by adding sensor measurements for different health state parameters.

1.3.1. Parameterized PDE

The reduced order models are typically used to find the solution of partial differential equations depending on a set of parameters $\mathcal{P} \subset \mathbb{R}^P$, with $P \geq 1$. These parameters affect the behavior of the solution and allow this kind of equations to describe a wide range of physical and operational conditions. For this reason, it is important to be able to solve efficiently PDEs for different parameter values. An example of parameterized PDE formulation is the linear elasticity equation, which is used to describe the deformation of a body with density ρ and with Lamé coefficients equal to λ, μ. The static version of the linear elasticity equation reads as: given the parameters ν, λ, μ find $\mathbf{d}(\mathbf{x}) : \Omega \to \mathbb{R}^d$ such

that:

$$\begin{cases} -\text{div}P(\mathbf{d}) = \mathbf{f} \ \text{ in } \Omega, \\ \mathbf{d} = 0 \quad \text{on } \Gamma_D, \\ P(\mathbf{d})\mathbf{n} = \mathbf{h}(\nu) \ \text{ on } \Gamma_N, \end{cases} \tag{1.13}$$

where $P(\mathbf{d}) = 2\mu\epsilon + \lambda \text{tr}(\epsilon)$ and $\epsilon = \frac{1}{2}(\nabla\mathbf{d} + \nabla^T\mathbf{d})$. The time-dependent version of the linear elasticity equation reads as follows: given the parameters ν, λ, μ, ρ find $\mathbf{d}(\mathbf{x}, t)$: $\Omega \times (0, T) \to \mathbb{R}^d$ such that:

$$\begin{cases} \rho\frac{\partial^2\mathbf{d}}{\partial t^2} - \text{div}P(\mathbf{d}) = \mathbf{f} \ \text{ in } \Omega \times (0, T), \\ \mathbf{d} = 0 \ \text{ on } \Gamma_D \times (0, T), \\ P(\mathbf{d})\mathbf{n} = \mathbf{h}(\nu) \ \text{ on } \Gamma_N \times (0, T), \\ \mathbf{d} = \mathbf{d}_0 \ \text{ in } \Omega \ \text{ when } T = 0. \end{cases} \tag{1.14}$$

Also in this latter equation we have that $P(\mathbf{d}) = 2\mu\epsilon + \lambda \text{tr}(\epsilon)$ and $\epsilon = \frac{1}{2}(\nabla\mathbf{d} + \nabla^T\mathbf{d})$. In both equations, we have an external force equal to $\mathbf{f}$, homogeneous Dirichlet conditions are imposed on Γ_D, while a Numeann condition, which depends on the parameter ν, is imposed on Γ_N. In all my further analysis, I will solve the equations (1.13), (1.14) for different parameter values.
The general formulation of a stationary parametrized PDE reads as: given the Hilbert space V and its dual $V^{'}$, given $\mu \in \mathcal{P}$, find the solution $u(\mu) \in V$ such that

$$L(\mu)u(\mu) = f(\mu) \ \text{ in } V^{'}, \tag{1.15}$$

where L is a partial differential operator, while f is a suitable functional.

Starting from equation (1.15), we can find its weak formulation, which reads as follows: given $\mu \in \mathcal{P}$, find $u(\mu) \in V$ such that

$$a(u(\mu), v; \mu) = f(v; \mu) \ \ \forall v \in V, \tag{1.16}$$

where

$$a(u(\mu), v; \mu) =< L(\mu)u, v >_* \tag{1.17}$$

$$f(v; \mu) =< f(\mu), v >_* . \tag{1.18}$$

Starting from the weak formulation (1.16) and considering a finite dimensional subspace $V_h \subset V$, $dim(V_h) = N_h$, we can find the algebraic Galerkin formulation of problem (1.15).

This formulation reads as follows: given $\mu \in \mathcal{P}$ find the solution $\mathbf{u}_h(\mu) \in V_h$ such that

$$A_h(\mu)\mathbf{u}_h(\mu) = \mathbf{f}_h(\mu). \tag{1.19}$$

By solving the system (1.19) we can find an accurate numerical solution for the starting problem, so this system represent our high-fidelity FOM. The matrix associated with this system is a $N_h \times N_h$ matrix, where N_h is the number of degrees of freedom (Dof) associated with the trial space V_h, which is usually very high; for this reason, solving the system (1.19) could be computationally expensive.

1.3.2. Reduced basis strategy

The reduced basis method is a ROM based on the idea of building a low dimensional subspace $V_N \subset V_h$ of dimension $N \ll N_h$ and finding an approximate solution as a linear combination of the basis functions spanning the space V_N. To set up the reduced basis problem, we have to follow these three steps:

- Step 1: construct a basis $\{\zeta_1, ..., \zeta_N\}$ for the reduced space such that $V_N = \text{span}\{\zeta_1, ..., \zeta_N\}$. This basis is called reduced basis (RB).
- Step 2: express the approximate RB solutions as $u_N(\mu) = \sum_{i=0}^{N} u_N^{(i)}(\mu)\zeta_i$, guaranteeing that the error between this solution and the FOM one is smaller than a fixed threshold.
- Step 3: set up a problem to find the RB coefficients $u_N^{(i)}$ with $i = 1, ..., N$; this problem will be found using a projection approach, and imposing a set of N constraints.

Proper orthogonal decomposition

As said before, the first step of the RB method is to find an orthonormal basis of the subspace V_N. To do this, given a set of n_s parameters properly sampled from the parameter space $\mathcal{P}$, we compute n_s high-fidelity solutions $\{u_h(\mu_1), ..., u_h(\mu_{n_s})\}$, called snapshots. In order to have a better representation of the parameters space $\mathcal{P}$ as sampling method we use the Latin hypercube sampling. Given these snapshots, we define the matrix $S \in \mathbb{R}^{N_h \times n_s}$ as

$$S = [u_1|...|u_{n_s}],$$

where the vector $u_i \in \mathbb{R}^{N_h}, 1 \leq i \leq n_s$, represents the degrees of freedom of the high fidelity solution $u_h(\mu_i)$. Given this matrix, we use the proper orthogonal decomposition (POD) method to find an orthonormal basis for our subspace V_N. POD is based on

the singular value decomposition (SVD) of the snapshot matrix S, Indeed, given a real matrix $A \in \mathbb{R}^{m\times n}$, there exist two orthogonal matrices $U = [\zeta_1|...|\zeta_m] \in \mathbb{R}^{m\times m}$ and $\mathbf{Z} = [\psi_1|...|\psi_n] \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma Z \quad \text{with } \Sigma = \text{diag}(\sigma_1, ..., \sigma_p) \in \mathbb{R}^{m\times n} \tag{1.20}$$

and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$, with $p = \min(m, n)$. In the decomposition (1.20) the numbers σ_i are the singular values of the matrix A and they are defined as the square root of the eigenvalues of the matrix AA^T. The columns of the matrices U, Z are respectively the left and right singular vectors of A and are defined respectively as the eigenvectors of the matrices AA^T and A^TA. A remarkable result concerning the SVD decomposition of a matrix is the Schmidt-Eckart-Young theorem which states that given a matrix $A \in \mathbb{R}^{m\times n}$ of rank r the matrix

$$A_k = \sum_{i=1}^{k} \sigma_i \zeta_i \psi_i^T, \quad 0 \leq k \leq r, \tag{1.21}$$

satisfy the optimality property

$$\|A - A_k\|_F = \min_{\substack{B\in\mathbb{R}^{m\times n} \\ \text{rank(B)}\leq k}} \|A - B\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}, \tag{1.22}$$

where, the norm $\|\cdot\|_F$ denotes the Frobenius norm. The idea of proper orthogonal decomposition for the RB method consists of applying the decomposition defined in (1.20) to the snapshot matrix $S = U\Sigma Z$; in this way, we can define a basis for the subspace V_N being formed by the first $N < n_s$ columns of the matrix U. If we define the correlation matrix $C = S^TS$, we can define the POD basis matrix $V \in \mathbb{R}^{N_h\times N}$ as a matrix whose columns are equal to :

$$\zeta_i = \frac{1}{\sigma_i} S\psi_i, \quad i = 1, ..., N \tag{1.23}$$

where σ, ψ are the eigenvalues and eigenvectors of the correlation matrix C. The dimension N of the subspace V_N can be selected so that the sum of squared errors between each snapshot and tits projection on the subspace generated by V_N is minimum. As reported in [23], it is possible to prove that:

$$\sum_{i=1}^{n_s} \|u_i - VV^Tu_i\|_2^2 = \sum_{i=N+1}^{r} \sigma_i^2, \tag{1.24}$$

so that to select the minimal POD dimension N such that the projection error is lower

than a fixed tolerance ϵ_{POD} we have simply to select N so that:

$$I(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=N+1}^{r} \sigma_i^2} \geq 1 - \epsilon_{POD}^2. \tag{1.25}$$

Note that condition (1.25) is equivalent to state that the N-rank approximation S_N to S satisfies

$$\frac{\|S - S_N\|_F}{\|S\|_F} \leq \epsilon_{POD}. \tag{1.26}$$

In Appendix B I reported the algorithm I used in my tests to find a suitable POD basis.

1.3.3. Galerkin Reduced Basis Method for stationary problems

Once we have found a basis $\{\zeta_1, ..., \zeta_N\}$ using the POD method, we have to set up a problem which allows us to write the approximated solution as $u_N(\mu) = \sum_{i=0}^{N} u_N^{(i)}(\mu)\zeta_i$. We build a reduced problem by projection; and in such a way we can find N equations whose solution is equal to $u_N^{(i)}$, $i = 1, ..., N$. In particular, we can impose that the residual of the FOM is orthogonal to the subspace $V_N \subset V_h$ of dimension N. Starting from the weak formulation (1.16) we obtain the Galerkin reduced basis (G-RB) formulation, which reads as: given $\mu \in \mathcal{P}$, find $u_N(\mu) \in V_N$ such that:

$$a(u_N(\mu), v_N; \mu) = f(v_N; \mu) \quad \forall v_N \in V_N. \tag{1.27}$$

Inserting $u_N(\mu) = \sum_{i=0}^{N} u_N^{(i)}(\mu)\zeta_i$ in 1.27 and choosing $v_N = \zeta_j$ we obtain

$$\sum_{i=1}^{N} a(\zeta_i, \zeta_j; \mu) u_N^{(i)}(\mu) = f(\zeta_j; \mu) \quad , i = 1, ..., N. \tag{1.28}$$

The problem above is equivalent to the following $N \times N$ linear system:

$$A_N(\mu)\mathbf{u}_N(\mu) = \mathbf{f}_N(\mu) \tag{1.29}$$

where the matrix $A_N \in \mathbb{R}^{N \times N}$ has components $(A_N(\mu))_{ij} = a(\xi_i, \xi_j; \mu)$, the vector $\mathbf{f}_N \in \mathbb{R}^N$ has components $(\mathbf{f}_N(\mu))_i = f(\xi_j; \mu)$, and the solution $\mathbf{u}_N(\mu) \in \mathbb{R}^N$ is a vector whose components are equal to $u_N^{(i)}(\mu)$. Usually, the matrix A_N is full while the matrix A_h is in general sparse. However, we usually have that $N \ll N_h$, so the system (1.29) is is much faster and less expensive to solve than the one related to the high-fidelity problem (1.19). Unfortunately, the assembling of the matrix A_N and of the vector $\mathbf{f}_N$ still involves a computational complexity of order N_h. To overcame this drawback, it is possible to

introduce the affine (or separable) parametric dependence assumption. The bilinear form a and the functional f are said to be affine with respect to the parameter μ if the following relations are satisfied:

$$a(w, v; \mu) = \sum_{q=1}^{Q_a} \theta_a^q(\mu) a^q(w, v) \quad \forall v, w \in V, \mu \in \mathcal{P}, \tag{1.30}$$

$$f(v; \mu) = \sum_{q=1}^{Q_f} \theta_f^q(\mu) f^q(v) \quad \forall v \in V, \mu \in \mathcal{P}, \tag{1.31}$$

where $\theta_a^q(\mu) : \mathcal{P} \to \mathbb{R}, q = 1, ..., Q_a$ and $\theta_f^q(\mu) : \mathcal{P} \to \mathbb{R}, q = 1, ..., Q_f$ are μ-dependent functions, and $a^q : V \times V \to \mathbb{R}, q = 1, ..., Q_a$ and $f^q : V \to \mathbb{R}, q = 1, ..., Q_f$ are μ-independent forms. The affine assumption is also inherited by the algebraic problem; indeed, we can redefine A_N and $\mathbf{f}_N$ as:

$$A_N(\mu) = \sum_{q=1}^{Q_a} \theta_a^q(\mu) A_N^q, \tag{1.32}$$

$$\mathbf{f}_N(\mu) = \sum_{q=1}^{Q_f} \theta_f^q(\mu) \mathbf{f}_N^q, \tag{1.33}$$

where we have that $(A_N^q)_{ij} = a^q(\xi_i, \xi_j)$ and $(\mathbf{f}_N^q)_i = f^q(\xi_i)$, we can obtain these matrices and vectors using the POD decomposition. Indeed, $A_N^q = V^T A_h^q V \in \mathbb{R}^{N \times N}$ and $\mathbf{f}_N^q = V^T \mathbf{f}_h^q \in \mathbb{R}^N$, where V is the matrix whose columns are the POD basis. The real advantage of this affine transformation is that we can compute and store all the quantities which are μ-independent in an offline phase. Then, for any given parameter value $\mu \in \mathcal{P}$, we assemble and solve the RB system (1.29) using the sums reported in 1.32, 1.33 with a cost depending only on N. For further details on the RB models, see [23].

1.3.4. Galerkin RB Method for time dependent problems

The Galerkin RB method can be extended to time dependent problems as well. Suppose that we want to solve a first order time dependent problem whose weak formulation reads as follows: given $\mu \in \mathcal{P}$, $\forall t > 0$ find $u(t; \mu) \in V$ such that $u(\mu) = u_0$ when $t = 0$ and

$$(\frac{\partial u(\mu)}{\partial t}, v) + a(u(\mu), v; \mu) = f(v; \mu) \quad \forall v \in V, t \in (0, T) \tag{1.34}$$

Proceeding as in [22], the space discretization formulation of (1.34) reads as:

$$M_h(\mu)\frac{\partial \mathbf{u}_h(t;\mu)}{\partial t} + A_h(\mu)\mathbf{u}_h(t;\mu) = \mathbf{f}_h(t;\mu), \quad t \in (0,T). \tag{1.35}$$

with $u_h = u_{h,0}$ when $t = 0$. In the equation (1.35), $M_h(\mu)$ is the mass matrix and both M_h and A_h are of dimension $N_h \times N_h$ and parameter dependent. If now we choose $V = V_N$, where V_N is the subspace spanned by the POD basis functions $\{\zeta_1, ..., \zeta_N\}$ and we define $u_N(t;\mu) = \sum_{i=0}^{N} u_N^{(i)}(t;\mu)\zeta_i$, we obtain a weak formulation of the corresponding RB problem which reads as follows: given $\mu \in \mathcal{P}$, find $u_N(\mu) \in V_N$ such that $u_N(\mu) = u_{0,N}$ when $t = 0$ and:

$$(\frac{\partial u_N(\mu)}{\partial t}, v_N) + a(u_N(\mu), v_N; \mu) = f(v_N;\mu) \quad \forall v_N \in V_N, t \in (0,T). \tag{1.36}$$

Proceeding as before, we obtain the following system to solve:

$$M_N(\mu)\frac{\partial \mathbf{u}_N(t;\mu)}{\partial t} + A_N(\mu)\mathbf{u}_N(t;\mu) = \mathbf{f}_N(t;\mu), \quad t \in (0,T) \tag{1.37}$$

where the matrices are defined as $(M_N(\mu))_{ij} = m(\xi_i, \xi_j;\mu)$, being $m(u,v;\mu) = (u,v)_\mu$, $(A_N(\mu))_{ij} = a(\xi_i, \xi_j;\mu)$, the vector $(\mathbf{f}_N(\mu))_i = f(\xi_j;\mu)$. These matrices and vectors have dimension N, so that problem (1.37) has a low computational cost than the one in (1.35). Once we have found this space discretization, we can simply solve (1.36) using a temporal discretization scheme like the one used in the FOM [22]. It is important to notice that the considered time-dependent problem 1.14 is a second order problem; in order to apply the techniques described in this section is important to rewrite it as a first order problem, see [22].

2 | An application: structural health monitoring of a concrete beam

In this chapter I will describe the application of the Bayesian network to monitor the health state of a bridge like structure. Starting from the unmanned aerial vehicle (UAV)test case reported in [14], I introduce a new test case involving a concrete doubly clamped beam, that is stressed by applying a weight at its midspan. As in [14], the aim is to monitor the health state of the beam by estimating the length of two different cracks which can form on the upper face of the beam after the application of the weight.. The dynamic Bayesian network receives information from the physical state thought 24 different sensor which can measure both the von Mises stress and the vertical displacement of the beam. In the absence of experimental data, these quantities were simulated through high fidelity numerical simulations, using a finite element method to solve the equation (1.13). The numerical simulations carried out to describe the potential behaviors of the beam are performed on the redbKIT Matlab finite element library [19].The aim of this test case is to verify that the methodology proposed in [14] can be also applied to more general test cases and therefore analyze its robustness.

2.1. Problem description

This test case involves the monitoring of the health state of a structure describing an idealized bridge model, during a time interval $[0, T]$. To model this kind of structure, I considered a concrete doubly clamped beam. This beam has height $h = 1\,\mathrm{m}$, length $l = 6\,\mathrm{m}$ and width $w = 1\,\mathrm{m}$. At every timestep of the simulation, the beam can be loaded by two different weights placed on its midspan; this kind of loading scenario is assumed to represent the load induced by traffic over the bridge. In particular, the severe load is equal to $300\,\mathrm{kg}$, and represents a situation in which there is a lot of traffic over the bridge; the less severe load, equal to $200\,\mathrm{kg}$, represents a low traffic situation. Similar to [14],

time is discretized in such a way that the transient period between the two admissible loading conditions is neglected, and only the related static configuration is accounted for. In this way, we only need to solve the static version of equation (1.13), with homogeneous Dirichlet boundary conditions on the two clamped sides and imposing a non-homogeneous Neumann boundary condition on the top surface of the beam, representing the external surface forces. The Lamè coefficients result from a Poisson ratio equal to 0.1, density equal to $2400\,\mathrm{kg/m^3}$ and Young modulus $E = 47.25\,\mathrm{Gpa}$. In figure 2.1 we can see how a beam without cracks deforms when a weight equal to $200\,\mathrm{kg}$ is applied on its middle section.

At each time step of the simulation, a weight is applied and the beam health state may worsen. I model the health state of the beam with a vector $\mathbf{z}$ whose components represent the length of a crack in a specific area of the beam, in percentage terms with respect to the total height of the beam. In particular, I assume that the cracks cannot change their position during the simulation and that they can only be stretched, worsening the health situation of the beam, with their width kept fixed to $0.02\,\mathrm{m}$. These assumptions allow to describe the health state of the beam using a limited number of parameters. To simulate the evolution of the structural degradation, similarly to [14] I prescribe a ground truth model which describes how the components of $\mathbf{z}$ evolve. This model is unknown to the digital twin and it is simply used to simulate sensor data. In particular, at every time-step we generate observation data starting from the current state, described by the ground truth model, and from the most recent input control; then these data are then passed to the probabilistic graphical model. As described in section 1.2, we have to model the probability $\phi_t^{dynamics}$; to this aim, I suppose that the probability of damage progression in each defect region is known, fixed, and conditionally independent given the load on the middle of the beam.

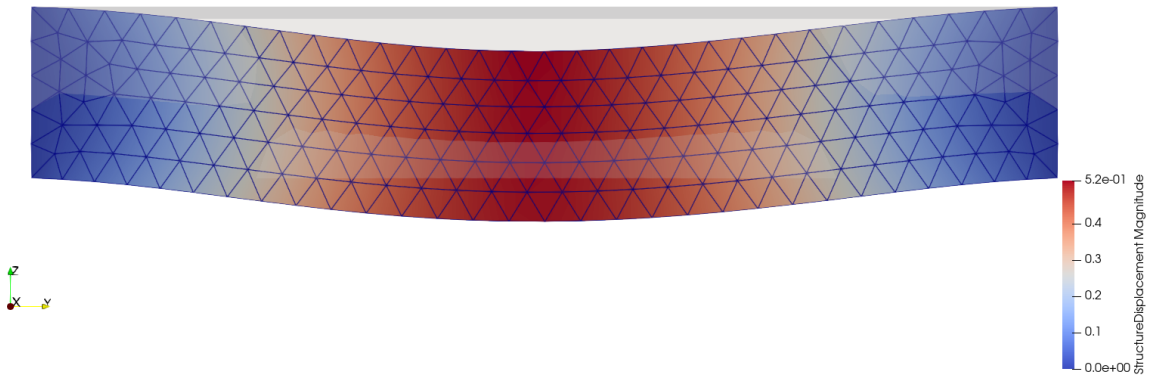


Figure 2.1: Beam displacement when $\mathbf{z} = 0$ and a 200 kg weight applied at its midspan.

The digital D_t state needs to include geometrical parameters and material parameters which have to be calibrated before the operational phase of the digital twin, as well as the parameters describing the current health situation of the beam. The digital state is described by the following vector of parameters:

$$\mathbf{d} = \left[\; l, \quad h, \quad w, \quad e, \quad \mathbf{z} \; \right]^T. \tag{2.1}$$

Where l, h, w are geometrical parameters describing the sizes of the beam, e is a material parameter adopted to rescale the concrete Young's modulus, and $\mathbf{z}$ is a vector of structural health parameters describing the health state of the structure as explained above. It is important to notice that the parameters which describe the digital state are similar to the one used in [14]; this is done on purpose, with the aim of testing the framework proposed in [14] in a new different setting. The principal difference between the two set of parameters is the fact that the parameters used in [14] in relation to the mass of the sensors are here neglected, because the observations used as input to the digital twin do not come from a real sensor but from numerical simulations. Similarly to [14], to estimate the digital state I calibrate all the parameters as described in the next section but the health state $\mathbf{z}$ which will be estimated in the operational phase.

2.2. Calibration phase

In this section I describe the calibration process of the parameters, $\mathbf{d}$, describing the digital state. This phase is crucial since it is very important that a digital twin describes in a persistent way the physical asset. The calibration phase is done at the first time-step of the simulation, $t = 0$, by adapting the code [13] to my test case. To do this, I consider a beam without crack so both the components of $\mathbf{z}$ are equal to zero, and by assimilating the data coming from the physical beam I calibrate the other parameters of $\mathbf{d}$, except for $\mathbf{z}$. For the geometrical parameters, instead of a proper calibration I simply created a mesh with height, length and width fixed to $l = 6\,\mathrm{m}, h = 1\,\mathrm{m}, w = 1\,\mathrm{m}$. The Young's modulus scaling factor e is an important parameter related with the material properties of the beam, which takes into account differences between beam asset due to variability in materials, manufacturing, or operational history. As in [14], the prior information of Young's modulus scale factor is modeled as a Gaussian distribution with mean equal to one and variance chosen in such a way the 95% credible interval covers the 5% of variability. To calibrate the Young's modulus scaling factor e I a particle filter algorithm, as in [14]. To apply this algorithm, I have to find a relationship between the scaling factor and the elastic modulus k, then I can proceed to calibrate e and update my prior belief on this parameter. To find a relationship between k and e a load displacement test was simulated. First of all, I have to verify that there is a linear relationship between the force applied in the middle of the beam f and the maximum measured displacement x. To find the relationship between f and x I put on the middle of the beam some weights between $0.2\,\mathrm{kg}$ and $20\,\mathrm{kg}$ keeping fixed the Young's modulus to $E = 47.25\,\mathrm{Gpa}$.

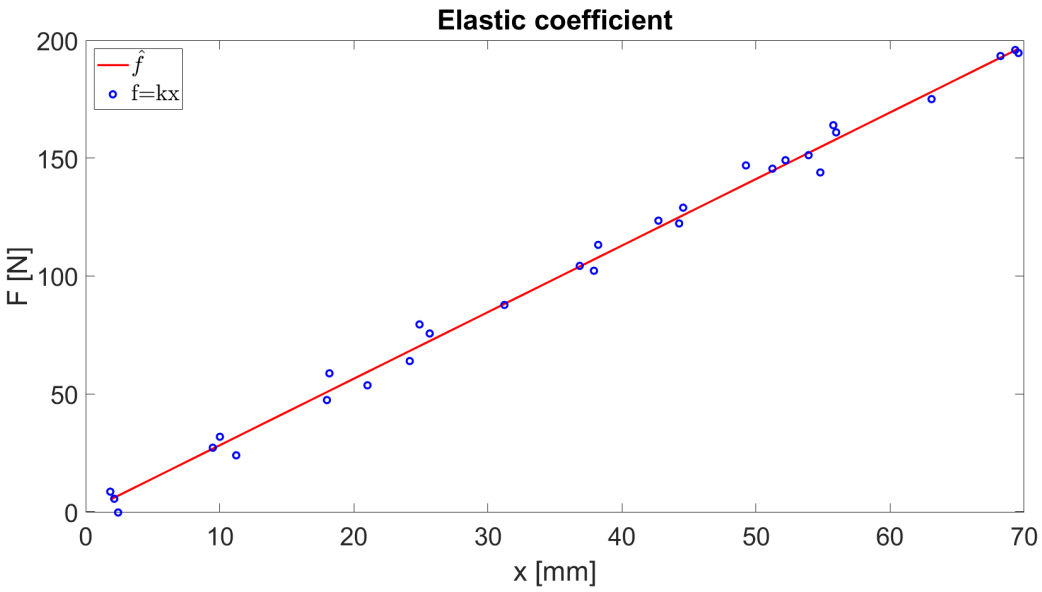


Figure 2.2: Linear relationship between the maximum displacement of the beam and a force applied in the middle of a beam with Young modulus equal to $E = 47.25\,\text{Gpa}$.

As shown in figure 2.2, we have a linear relationship between x, f, and so the elastic coefficient can be computed as $k = f/x$. To find a relationship between the Young's modulus scaling factor and the linear elastic coefficient, I have first drawn 30 samples from the prior distribution of e, and for each of them, the corresponding displacement at midspan was computed under the action of a 10 Kg mass applied at midspan, from which the elastic coefficient k was finally computed. Using linear regression, I found the results in Figure 2.3 and the following relationship:

$$k = 2.8229e. \tag{2.2}$$

Once this relationship is found, I can use the particle filter algorithm implemented in [13] to calibrate the Young's modulus scaling factor e by updating the prior information on this parameter. In particular, I tuned the number of particles used by the particle filter algorithm so that the variance of the posterior distribution was as small as possible.

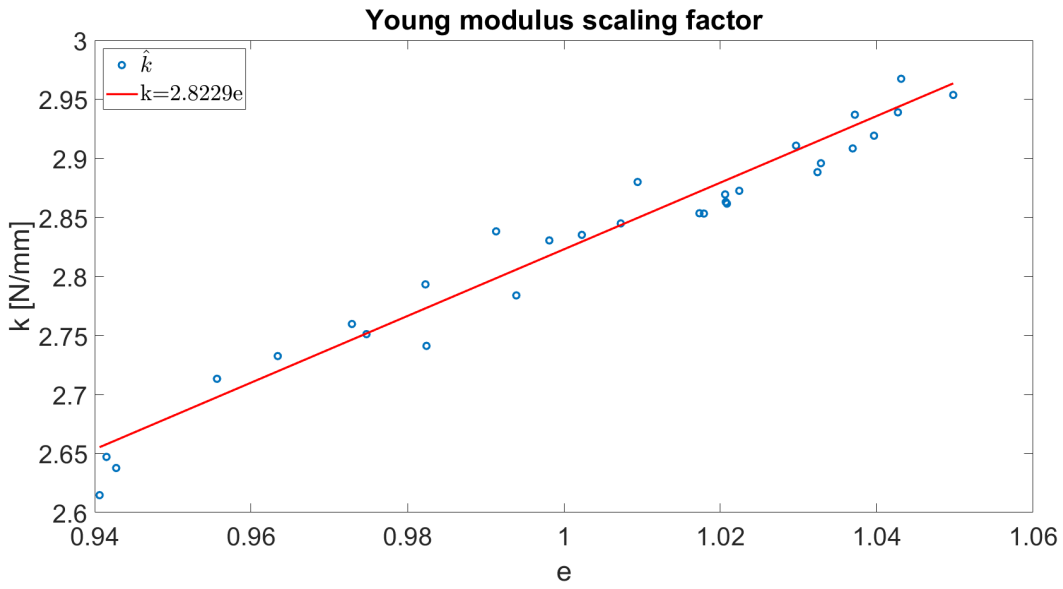


Figure 2.3: Linear relationship between the measured young coefficient $\hat{k}$ varing the Young's modulus scaling factor e.

Figure 2.4 reports the obtained posterior probability distribution, which is centered at 0.9919 (corresponding to a 0.81% reduction) and features a reduced standard deviation. Once the calibration phase is completed, the input dataset for the DBN is assembled by simulating a set of potential damage scenarios affecting the structure. This is done for each possible damage state $\mathbf{z}$ with reference to the updated distribution of parameter e, by taking 30 samples from the posterior and computing the relevant quantities of interest for each sample, for instance in terms of displacements or strains.

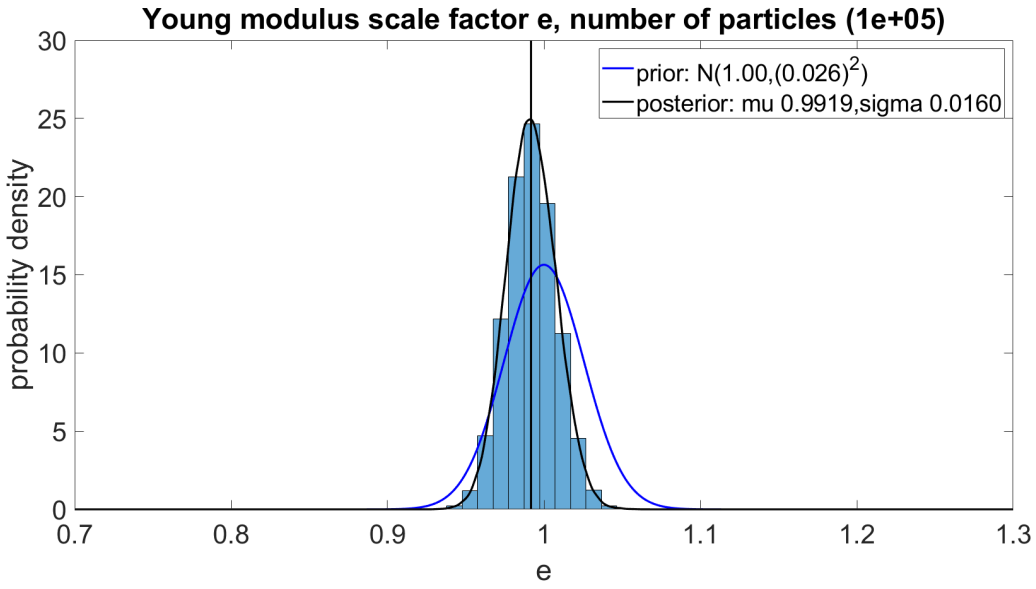


Figure 2.4: Prior and posterior distributions of the Young's modulus scaling factor e for the beam test case.

2.3. Operational phase

In this section I describe the prediction of the health state $\mathbf{z}$, by adapting the code [12] to my test case. Is important to notice that all the parameters characterizing the digital state d, except $\mathbf{z}$, are fixed to their values or distribution found during the calibration phase. The variables that characterize the probabilistic graphical model for all $t > 0$ are:

- O_t, the observation data measured by the sensors placed on the lower face of the beam;
- D_t, the digital state characterized by the parameters collected in d;
- Q_t, the quantity of interest, approximating the sensors measurements, as computed from the updated digital state;
- U_t, represents the decision node.

Is important to notice that in this case R_t is not defined; indeed, the adopted control policy is prescribed without solving a planning problem. Here we note that, with respect to [14], we have a different quantity of interest, a different number of actions and a greater dimension of the vector $\mathbf{z}$ which describes the health state of the beam. In particular, I focused on four different tests:

- Test 1: using the stress as QoI;
- Test 2: using the displacement as QoI;
- Test 3: three control actions;
- Test 4: the dimension of the vector $\mathbf{z}$ equal to three.

In all these tests, to measure the performances I used an error function similar to the one used in [14], which is defined as:

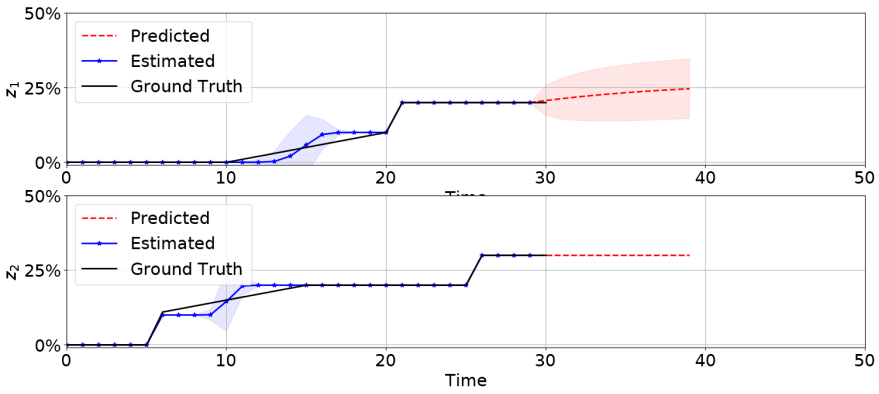
$$E = -\frac{1}{24}\sum_{i=1}^{24}\frac{|\epsilon_i - \hat{\epsilon}_i|}{\sigma_{sensor}}, \tag{2.3}$$

where ϵ is the considered QoI while the $\hat{\epsilon}$ indicates the real measured data and σ_{sensor} is equal to 125 microstrains in the case in which I used the stress as quantity of interest and to 12.5×10^{-4}m in the case the displacement is used as QoI. For all these tests I prescribed a precise policy which I will illustrate in details in the following. As time interval I consider $[0, 40]$ and as prediction horizon I set a time equal to $t_p = t_c + 10$, where t_c is the current time-step. The figures reporting the result include both the ground truth true value of each quantity and the estimated ones, as well as the 95% confidence interval for the prediction. To validate the prediction capacity of the digital twin, the results are reported for two different time-steps, $t_c = 30$ and $t_c = 40$.

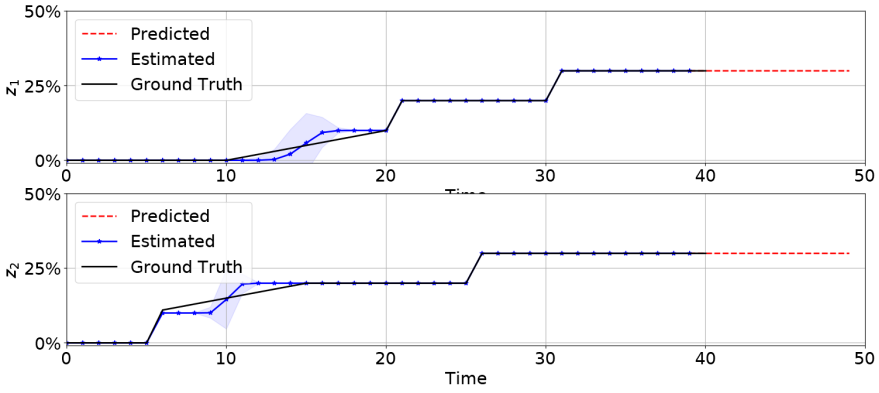
2.3.1. Test 1: using the stress as QoI

First of all, I validate the code using a test case similar to the one in [12], by defining $\phi_t^{dynamics}$ and the parameter $\mathbf{z}$ as in [14], thus yielding $\mathbf{z} = \{0\%, 20\%, 40\%, 60\%, 80\%\} \times \{0\%, 20\%, 40\%, 60\%, 80\%\}$. I do not report the results for this validation test case as the values of possible crack lengths are unrealistically too high. Once I have validated the code, I considered a new test case with reasonable values for the lengths of the cracks $\mathbf{z} = \{0\%, 10\%, 20\%, 30\%\} \times \{0\%, 10\%, 20\%, 30\%\}$. For this test case, I consider a new definition of $\phi_t^{dynamics}$. In particular, I suppose that the structural health in each defect region has a probability equal to 0.05 to worsen by 10% when the 200 kg load is applied, and a probability equal to 0.05 to worsen by 10% when the heaviest weight is applied. For this test case I defined a policy Π which consists in two actions: if a component of $\mathbf{z}$ is greater then 10% the beam is excited with smallest weight; in the other cases, the heaviest weight is placed on the beam. The von Mises stress is considered as quantity of interest. In Figure 2.5, I reported the results regarding the prediction of the components of $\mathbf{z}$. In these figures, we can see that both components are well predicted and that in

the region in which the ground truth model visits the states not belonging to the range $\{0\%, 10\%, 20\%, 30\%\} \times \{0\%, 10\%, 20\%, 30\%\}$ there is a greater uncertainty.



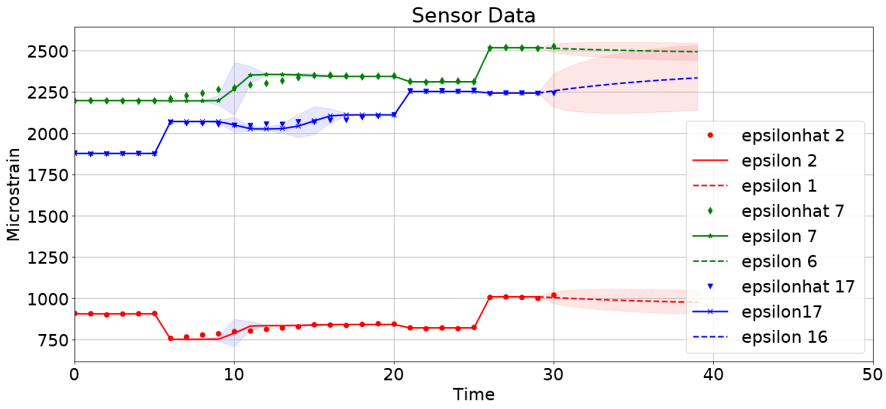
(a) Prediction of the state at $t_c = 30$.



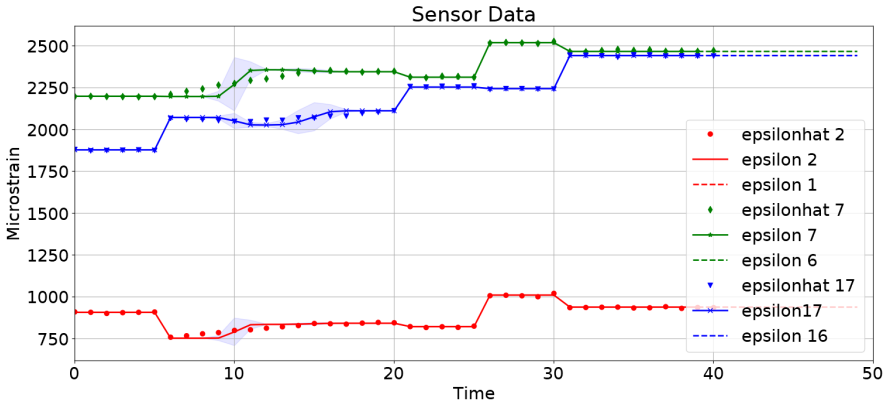
(b) Prediction of the state at $t_c = 40$.

Figure 2.5: Test case 1, using the stress as QoI: estimated and predicted evolution of the health state and the prescribed ground truth model for two different current time steps.

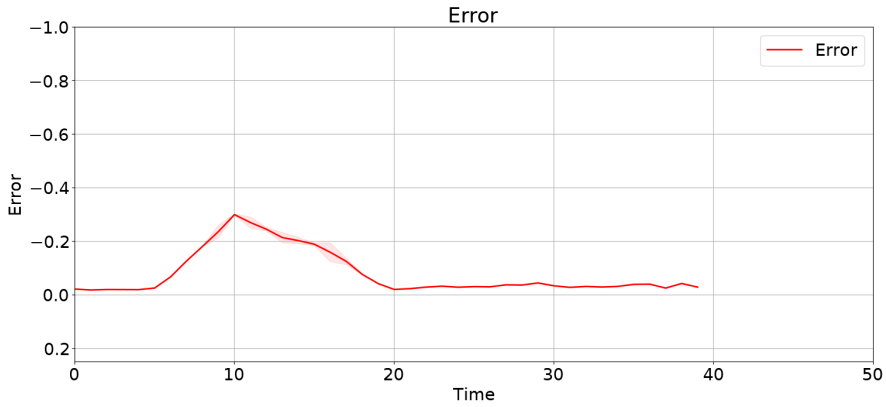
The results regarding the prediction of the quantity of interest are reported in Figure 2.6, highlighting that also the prediction of the stress measured by the sensors is good, and the relative uncertainty is attributed to the uncertainty affectingly the prediction of $\mathbf{z}$.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



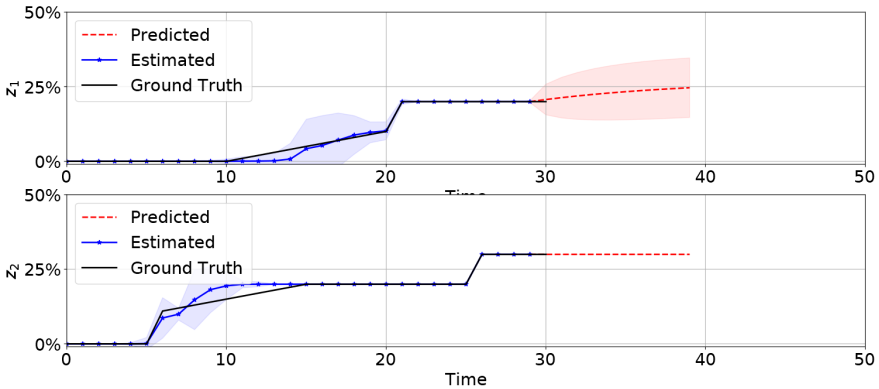
(c) Error during simulation time interval

Figure 2.6: Test case 1, using the stress as *QoI*: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

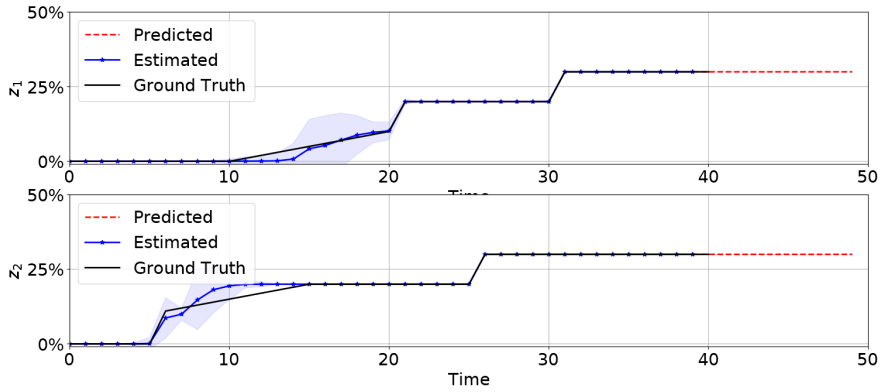
From these results, we can conclude that the method described in [14] can be used in different test cases where the health state is no longer described through the percentage reduction in material stiffness, but by the length of the cracks that form on the upper face of the beam.

2.3.2. Test 2: using the displacement as QoI.

As in test 1, I have considered $\mathbf{z} = \{0\%, 10\%, 20\%, 30\%\} \times \{0\%, 10\%, 20\%, 30\%\}$ and the same definition of both $\phi_t^{dynamics}$ and Π. Through this test, I want to verify that the digital twin model also works in the case in which I do not consider the stress as QoI. Indeed, the only difference with respect to the previous test is that now I consider the displacement as quantity of interest. In particular, I consider the vertical displacement of the beam, measured by 24 sensors located uniformly on the bottom surface of the beam. In figure 2.7, I reported the results concerning the prediction of the components of $\mathbf{z}$. In particular, we can see that the uncertainty bounds are greater then before; this can be due to the fact that the displacement is less informative about the components of $\mathbf{z}$, and for this reason in the further tests I keep using the stress as quantity of interest.



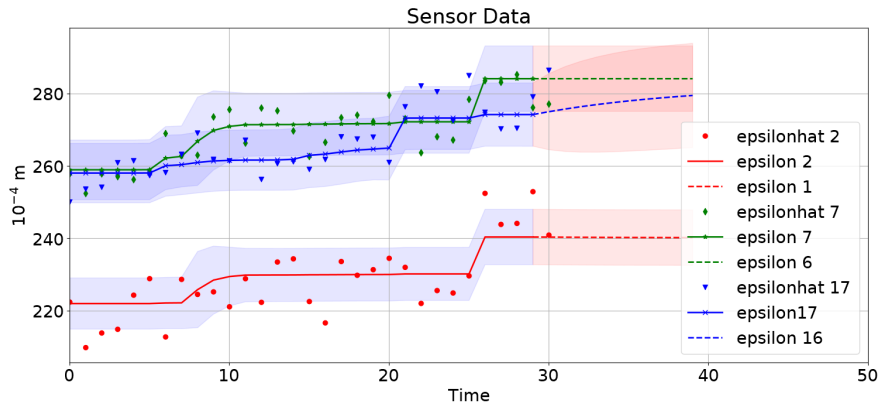
(a) Prediction of the state at $t_c = 30$.



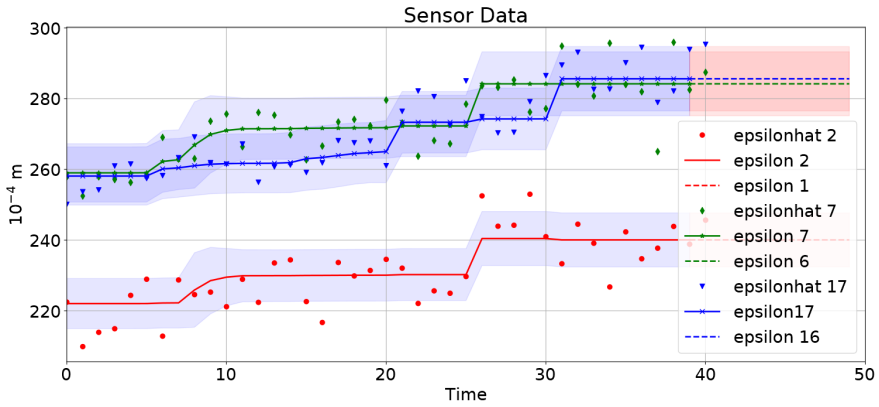
(b) Prediction of the state at $t_c = 40$.

Figure 2.7: Test case 2, using the displacement as QoI: estimated and predicted evolution of the health state and the prescribed ground truth model.

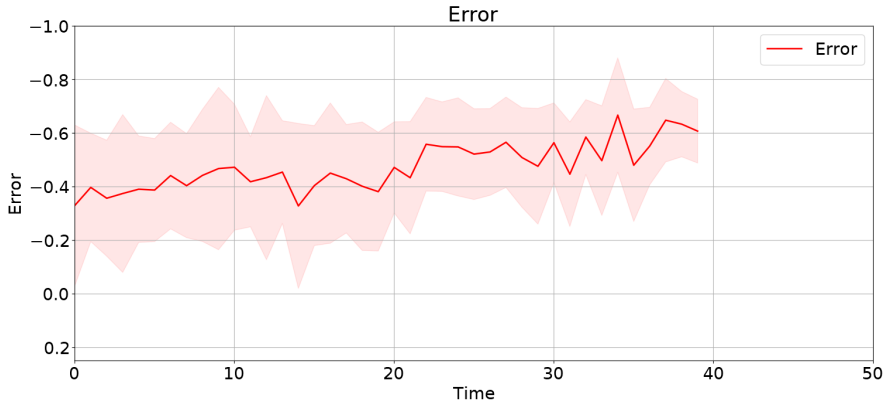
In figure 2.8, I report the results concerning the estimate and the prediction of the displacement measured by the sensors.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



(c) Error during simulation time interval.

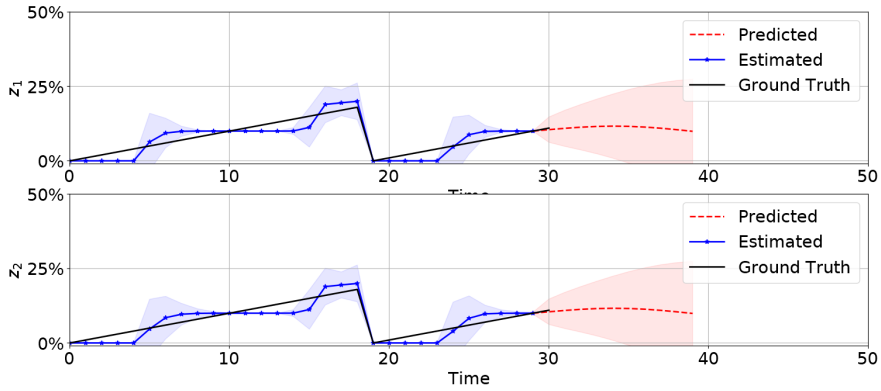
Figure 2.8: Test case 2, using the displacement as QoI: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

Looking at these results, it is evident that some quantities of interest might be more significant than others in describing the state of health of the physical space. For example, in this case the uncertainty bands of the second test case turn out to be wider than those of the first test case, where the stress is considered as QoI.

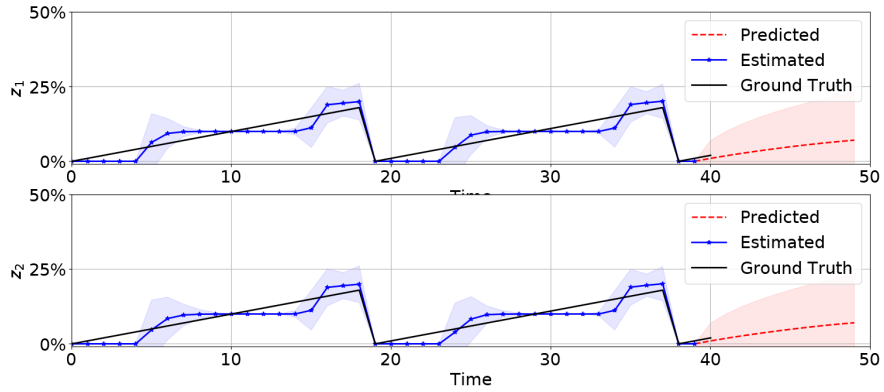
2.3.3. Test 3: three control actions

In this third test I keep the same definition of both $\mathbf{z}$, and $\phi_t^{dynamics}$ and I used the stress as quantity of interest; the novelty is that a new action $\mathbf{r}$, is now introduced, consisting in the

reparation of the beam. Indeed, it is important to notice that another difference between this test case and the UAV test cases presented in [14] is the fact that during a mission the UAV cannot be repaired, but only a less aggressive manoeuvre can be taken, while a beam during its life cycle can be repaired. The new policy $\hat{\Pi}$ consists in performing the action `r` in the case in which there is at least one component of $\mathbf{z}$ greater than 20%, doing the action `SW` in the case in which there is at least one component greater than 10% and apply the heaviest weight in all the other cases. To describe the evolution of the component of $\mathbf{z}$, I do not use anymore a custom model but a linear one. In figure 2.9, I report the results regarding the prediction of the health state, despite of the good estimation capabilities of the digital twin, the relative uncertainty is increased with respect to the previous cases; this is likely due to the introduction of a new action and the higher complexity of the ground truth model. By looking at the prediction interval, it is interesting to note that the digital twin is also able to predict when the repair action `r` should be performed.



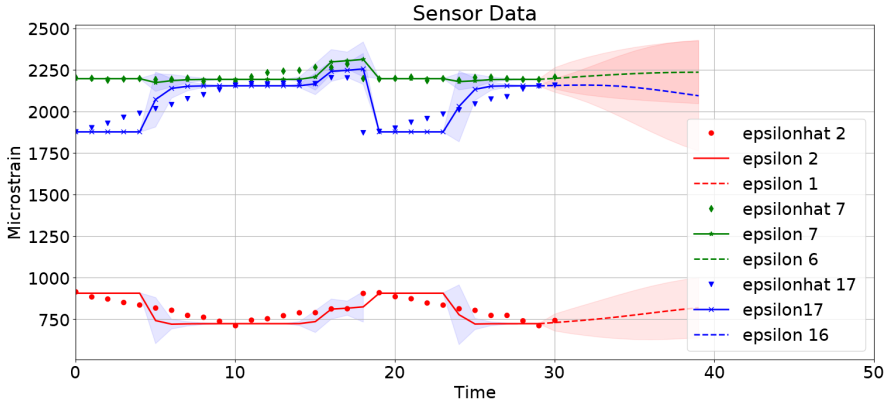
(a) Prediction of the state at $t_c = 30$.



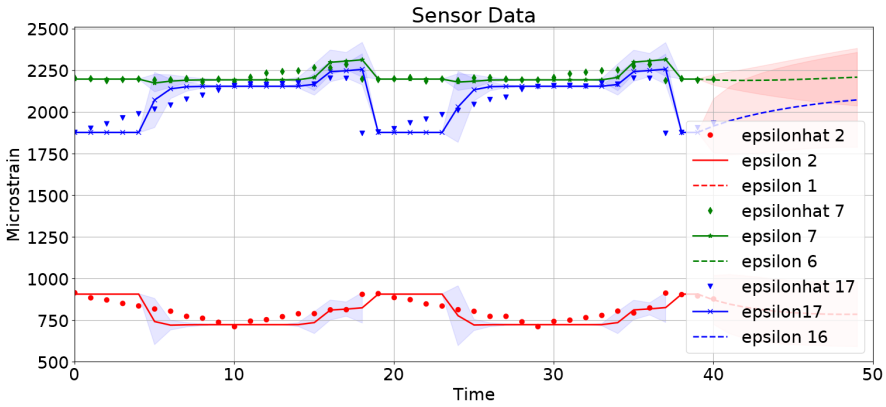
(b) Prediction of the state at $t_c = 40$.

Figure 2.9: Test case 3, three control actions: estimated and predicted evolution of the health state and the prescribed ground truth model.

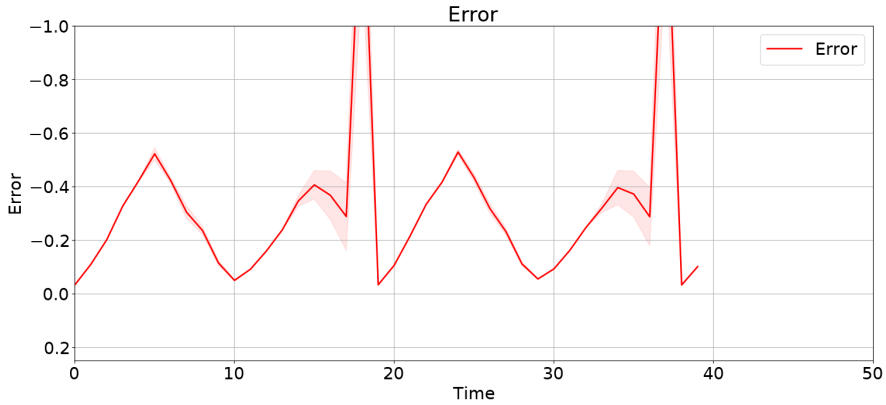
In Figure 2.10, I report the results regarding the estimate of the data measured by the sensor and the relative error between measured and estimated quantities, E. In particular, we can notice that this estimate is good in every time step, but the ones in which the action `r` is applied.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



(c) Error during simulation time interval.

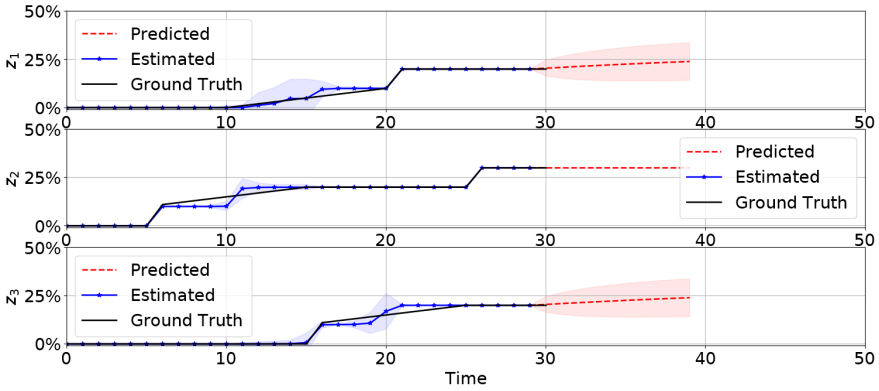
Figure 2.10: Test case 3, three control actions: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

This test case allows me to state that the Bayesian network can also be used in complex cases similar to reality. In fact, in this test case, due to the presence of the `r` action, the health state of the beam can change suddenly, however the digital twin strategy is still able to correctly estimate both the crack length and the stress measurements.

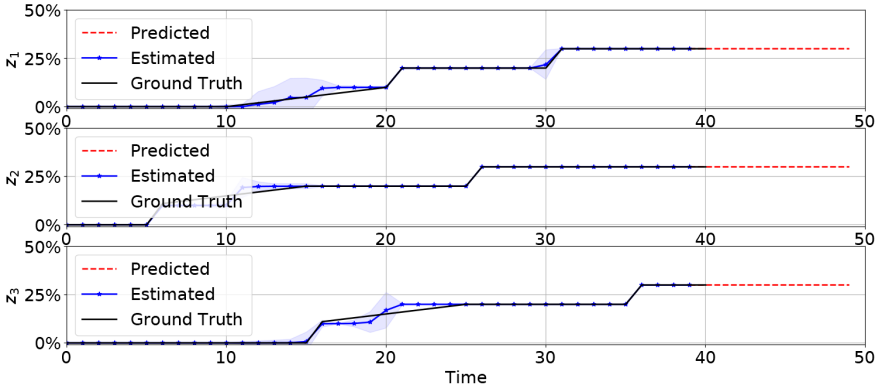
2.3.4. Test 4: dimension of the vector z equal to three

In all the previous test cases, the health state of the beam $\mathbf{z}$ is modeled by two components representing the length of two different cracks. In this new test case, I want to analyze a

more complex situation with an health state vector $\mathbf{z}$ characterized by three components. In particular, the i-th component of this vector represents the length of a crack (in percentage terms) in the i-th zone. The three different zones are located at $y = 2, 4, 5\,\text{m}$, yielding $\mathbf{z} = \{0\%, 10\%, 20\%, 30\%\} \times \{0\%, 10\%, 20\%, 30\%\} \times \{0\%, 10\%, 20\%, 30\%\}$ and respectively $\phi_t^{dynamics}$ is as before. In this test case, for the sake of simplicity, I only consider the actions `SW` and `HW`, prescribing the same policy of tests 1 and 2. In figure 2.11, I reported the results regarding the prediction of the health state and, from which it can be observed that the digital twin seems to work better in the case in which it has to estimate more than two parameters.



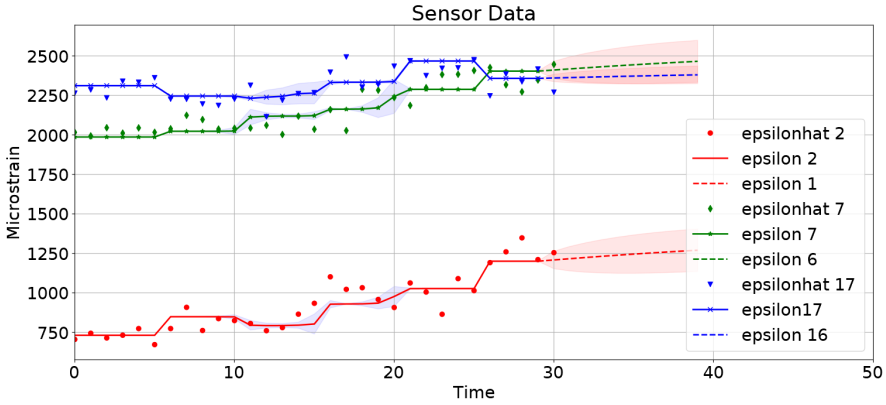
(a) Prediction of the state at $t_c = 30$.



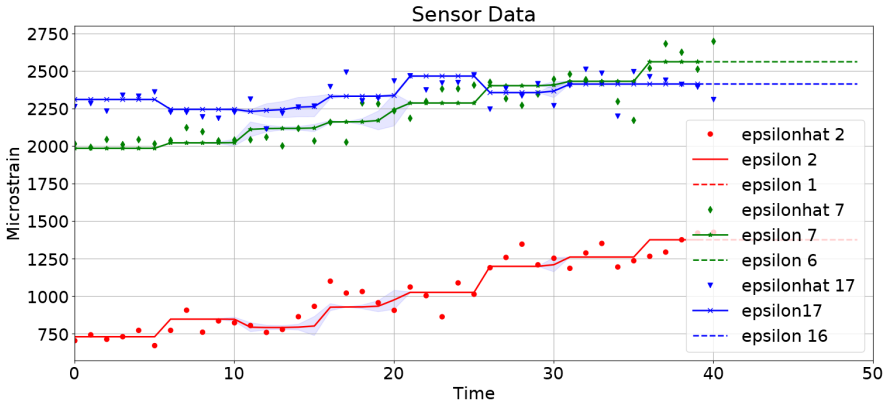
(b) Prediction of the state at $t_c = 40$.

Figure 2.11: Test case 4, dimension of the vector $\mathbf{z}$ equal to three: estimated and predicted evolution of the health state and the prescribed ground truth model.

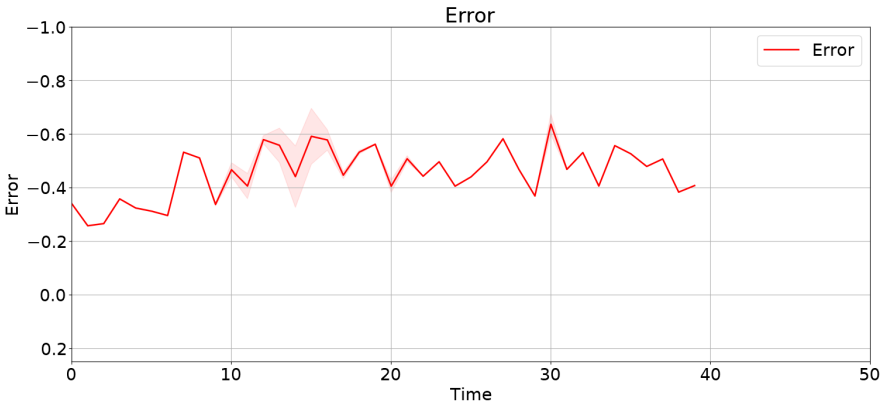
In Figure 2.12, I report the results regarding the estimate of the stress measured by the sensors. Also in this case, we have good prediction and estimate results, as confirmed also by the plot of the error in Figure 2.12c.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



(c) Error during simulation time interval.

Figure 2.12: Test case 4, dimension of the vector **z** equal to three: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

In light of these results, I can conclude that the probabilistic graphical digital twin model can be used also to manage situations characterized by more than two structural health parameters. These four test cases allow us to conclude that the digital twin metodology proposed in [14] is a very robust and flexible tool. This model works in the case where a different definition of the health of the system is used, it is not dependent on the quantity of interest considered, and it can be also used in more complex cases where there are more than two actions or more than two parameters involved.

3 | Combined use of a ROM and of a Dynamic Bayesian Network for structural monitoring

In this chapter, I consider a new test case to assess the performance of the dynamic Bayesian network. This new test case consists of a beam which is composed by ten different subregions, where the Young modulus E is parametrized to account for different damage scenarios. I use this test case to fully exploit the framework described in Figure 1, see Chaper 1. The two main results presented deal with the use of data augmentation to improve the accuracy performance of the DBN and the use of a ROM to build the input dataset and to model the sensor observations. In this way we can simulate many scenarios at a very low computational cost: for example, we can load the beam with different weights or take into account different damage scenarios. Then we use the digital twin to predict the health of a beam model in all these cases. The reminder of the chapter is organized as follows. In Section 3.1 we describe the problem at hand; the adopted strategy for model order reduction is detailed in Section 3.2, and the obtained results in terms of estimation and prediction of the health state of the beam are finally reported in Section 3.3.

3.1. Problem description

Once again, the objective is to estimate the health state of an idealized bridge-like structure, in this new test case, the physical asset to be monitored is a doubly clamped concrete beam. The lenght of the beam is 6 m while its height and width are fixed to 1 m. The mechanical properties are those of concrete: Poisson ratio equal to 0.1, Young's modulus $E = 47.25\,\text{GPa}$, density equal to $2400\,\text{kg/m}^3$. At every time step of the simulation, the beam can be loaded by a weight located at its midspan that can be equal to 150 kg or to 100 kg. As done in [14], the time interval of the simulation is discretized in such a way that we can consider only what happen when the weight is applied and ignoring the transient period between two load applications. In this new test case, the health state is a vector

$\mathbf{z} = [z_1, z_2]$ and its components represent the percentage of material stiffness reduction in two of the ten subregions in which the beam is partitioned in order to model possible damage scenarios. An undamaged beam feature a Young modulus $E = 47.25\,\text{GPa}$ in all the ten subregions; when one of the ten subregions features a reduced value of the Young modulus, the beam is damaged. As before, this physical state can be observed only through numerical simulations, in particular, we assume to observe numerically generated pseudo-expirimental recordings, in terms of displacements or stresses from 24 sensors placed on the lower face of the beam. Also in this case, before the operational phase in which the digital twin estimates the health state of the beam, we need to calibrate the parameters which characterize the digital state. In order to use the same calibration carried out in Section 2.2, I use the typical lengths of the beam and the Young modulus scaling factor as the parameters which describe the digital state and that need to be calibrated. Note that the calibration phase is done assuming that the beam is in a undamaged health state so that the definition of the health state vector $\mathbf{z}$ does not affect the calibration phase.

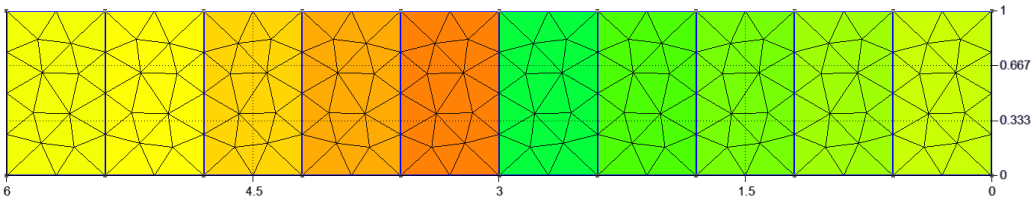


Figure 3.1: Mesh of the beam used for numerical simulations. The different colors underline the subregions of the beam in which the Young modulus can take different values.

3.2. Reduced order modeling

In this section, I will present the mathematical formulation of the problem which I need to solve and all the properties of the ROM used. The high fidelity FOM that I use for computing the stress measurements solve the linear elasticity equation (1.13) which

describes how an elastic material moves when is subjected to external forces. The unknown of this equation is the displacement of the beam $\vec{d}$; Ω is the domain occupied by the undeformed beam and in particular we have that $\Omega = \cup_{i=1}^{10}\Omega_i$ being Ω_i are the subdomains of the mesh, see Figure 3.1. In the linear elasticity equation solved in this test case $f = 0$, indeed there are no volumetric forces which act on the beam, Γ_N is the upper face of the beam, while Γ_D are the ends of the beam, the function $\mathbf{h}(l) = l\mathbb{1}(\Gamma_F)$, where Γ_F is the subregion of the upper face of the beam for which $2 \leq y \leq 3$. This equation is characterized by three different parameters, l, λ, μ. The parameters ν, λ are dependent on the Poisson ratio ν and on the Young modulus E, indeed we have that $\mu = \frac{E}{2(1+\nu)}$ and $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$. Note that the Young modulus could be different in each zone of the beam; the parameter l represents instead the weight of the load that is applied on the midspan of the beam. Starting from the equation (1.13) we can find its associated weak formulation that read as:

$$\int_\Omega P(\mathbf{d}) : \nabla \mathbf{w}\, d\Omega = \int_{\Gamma_N} \mathbf{h}(l)\mathbf{w}\, d\Gamma \qquad \forall \mathbf{w} \in V. \tag{3.1}$$

Considering the partitioning of the domain Ω and using the linear property of the integral we have that (3.1) is equivalent to:

$$\sum_{i=1}^{10} \int_{\Omega_i} P(\mathbf{d}) : \nabla \mathbf{w}\, d\Omega_i = \int_{\Gamma_N} \mathbf{h}(l)\mathbf{w}\, d\Gamma \qquad \forall \mathbf{w} \in V. \tag{3.2}$$

Substituting the linear constitutive law, we have that:

$$\sum_{i=1}^{10} \mu_i \int_{\Omega_i} (\nabla \mathbf{d} + \nabla \mathbf{d}^T) : \nabla \mathbf{w}\, d\Omega_i + \sum_{i=1}^{10} \lambda_i \int_{\Omega_i} \mathrm{tr}(\nabla \mathbf{d}) : \nabla \mathbf{w}\, d\Omega_i = l \int_{\Gamma_F} \mathbf{w}\, d\Gamma \qquad \forall \mathbf{w} \in V, \tag{3.3}$$

where μ_i, λ_i are the coefficient computed using the Young modulus of the subdomain Ω_i. Introducing the operators $c_i(\mathbf{u}, \mathbf{v}) = \int_{\Omega_i} (\nabla \mathbf{u} + \nabla \mathbf{u}^T) : \nabla \mathbf{v}\, d\Omega_i$, $\mathbf{b}_i(\mathbf{u}, \mathbf{v}) = \int_{\Omega_i} \mathrm{tr}(\nabla \mathbf{u}) : \nabla \mathbf{v}\, d\Omega_i$, and $\mathbf{f}(\mathbf{v}) = \int_{\Gamma_F} \mathbf{v}\, d\Gamma$, we can rewrite equation (3.1) as:

$$\sum_{i=1}^{10} \mu_i c_i(\mathbf{d}, \mathbf{w}) + \sum_{i=1}^{10} \lambda_i b_i(\mathbf{d}, \mathbf{w}) = l f(\mathbf{w}) \qquad \forall \mathbf{w} \in V. \tag{3.4}$$

We can note that the bilinear forms that we have introduced are affine with respect to the parameters λ, μ and l. Proceeding as in [22], the algebraic formulation of the Galerkin

finite element FOM reads as:

$$A_h(\mu)\mathbf{u}_h = \mathbf{b}_h. \tag{3.5}$$

The affine property is also inherited by this algebraic problem so we can decompose $A_h(\mu)$ as

$$A_h(\mu) = \sum_{i=1}^{10} \theta_c^i(\mu) C_h^i + \theta_b^i(\mu) B_h^i \tag{3.6}$$

where the entrance of the matrices are defined as $(C_h^i)_{jk} = c_i(\phi_k, \phi_j)$, $(B_h^i)_{jk} = b_i(\phi_k, \phi_j)$ and the element of the vector are defined as $(b_h)_k = lf(\phi_k)$. The matrices C_h^i, B_h^i are independent by the parameters while the functions $\theta_c^i(\mu), \theta_b^i(\mu)$ are defined as:

$$\theta_c^i(\mu) = \frac{E_i}{2+2\nu}, \qquad \theta_b^i(\mu) = \frac{E_i \nu}{(1+\nu)(1-2\nu)} \tag{3.7}$$

where E_i, ν are the Young modulus and the Poisson ratio referred to the subdomain Ω_i.

3.2.1. Reduced order model

The ROM I used to carry out our further analyses is a Galerkin reduced basis method. To train this method, first of all I computed $n_s = 100$ snapshots for different values of the parameters E_i and l, suitably sampled with the Latin hypercube sampling technique. In particular, I sampled each E_i in a range of values [4.725 Gpa , 51.975 Gpa] and I sampled l in a range of values [98 N ,1783.6 N]. Then, I computed the RB using the POD technique reported in appendix B with $\epsilon_{POD} = 10^{-3}$. As we can see in figure 3.2, under these assumptions the resulting reduced order model only features $N = 43$ degrees of freedom.

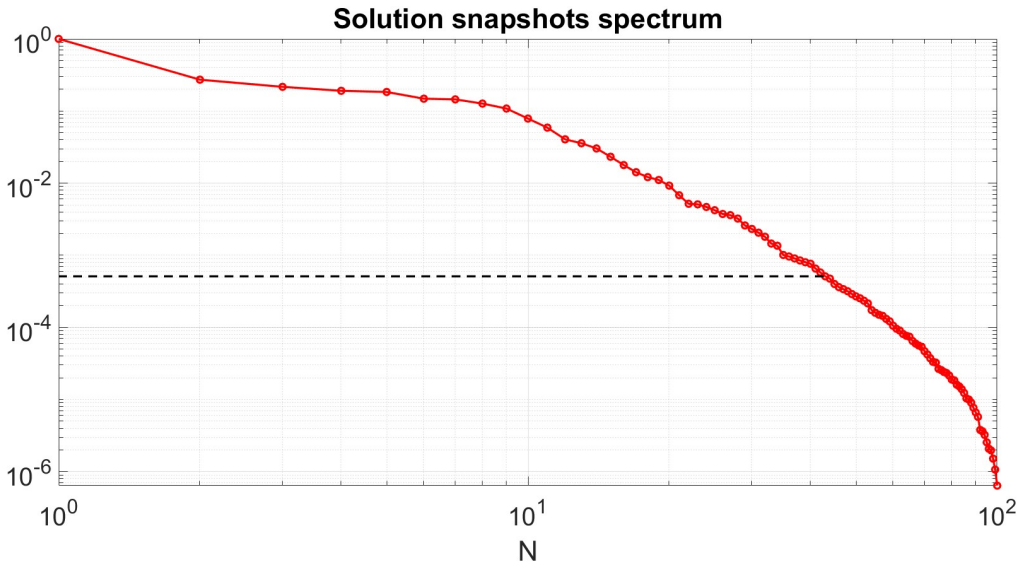


Figure 3.2: Solution snapshots spectrum obtained with a $n_s = 100$ snapshots and a tolerance $\epsilon_{POD} = 10^{-3}$.

Once I have obtained the desired ROM, I test its accuracy and performance on a testing set of dimension 10 and I compare its performance and accuracy with the one of the FOM used to compute the snapshots. The obtained results are reported in 3.1.

	FOM	**ROM**
DOF	4547	43
Time	4.47×10^{-1}s	1.27×10^{-2}s
Error	-	0.268 %

Table 3.1: Comparison between the performances and the accuracy of the FOM and the Galerkin RB method with $N = 43$ basis.

In particular, the time is referred to the mean time of assembling and solving both the FOM and the ROM, while the error is the relative error between the solution obtained using the ROM and the FOM. By analyzing the results reported in Table 3.1, we can notice that despite a relative error equal to 0.268% the ROM is more or less 35 faster

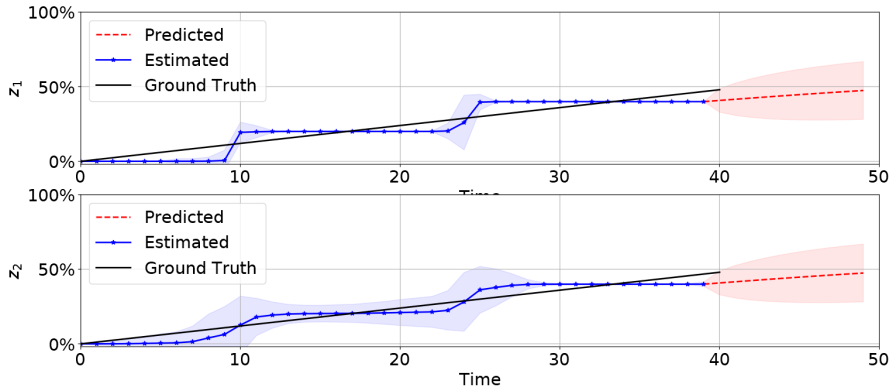
then the FOM. So, the ROM allows us to obtain accurate numerical results in a very rapid time.

3.3. Operational phase

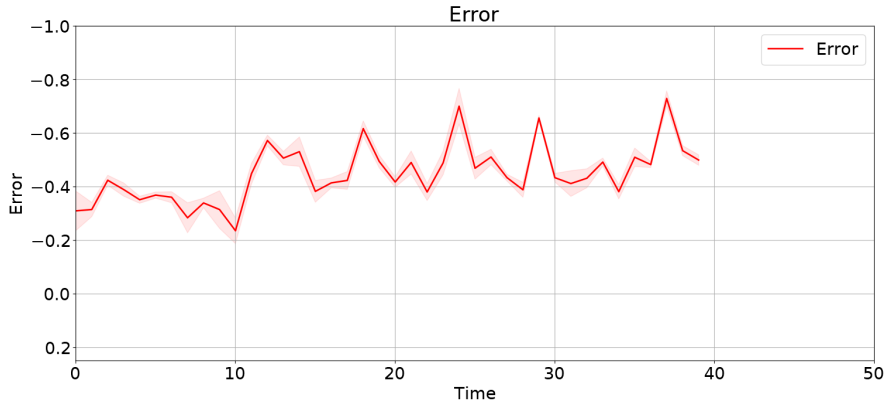
In this section, I report all the results concerning with the prediction of the health state $\mathbf{z}$. In particular, I want to improve what proposed in [14] by using the ROM presented in subsection 3.2.1 by allowing for a data augmentation on the input dataset to improve the accuracy of the dynamic Bayesian network; the relevant results are reported in subsection 3.3.1. In a subsequent analysis, I use the ROM to create an input dataset in which the user can choose wherein the material stiffness is reduced and also choose the two weights to be applied at the midspan; the relevant results are reported in subsection 3.3.2. Then, I consider a test case in which the location of the region wherein the stiffness is reduced is supposed to be unknown and the digital twin to find the position of this zone; the relevant results are reported in subsection 3.3.3. The last two test cases in subsections 3.3.4, 3.3.5 instead focus on the importance of the ROM accuracy and of the calibration phase; in particular, I want to analyze how the results of the Dynamic Bayesian network are influenced by an inaccurate ROM or by a poor calibration phase.

3.3.1. Test 1: Data Augmentation using a ROM

In the present test case, the health state of the beam is modeled using a vector of two components $\mathbf{z} = \{0\%, 20\%, 40\%, 60\%\} \times \{0\%, 20\%, 40\%, 60\%\}$; the first component represents the percentage reduction of the material stiffness of the fourth subregion of the beam while the second one is referred to the seventh subregion of the beam. As said in the previous chapters, the input dataset for the dynamic Bayesian network contains the measurements of the 24 sensors for each possible health state situation and for each possible value of the load applied on the midspan of the beam. The sensors measurement are obtained with an high fidelity FOM providing the numerical solution of equation (1.13) and the stress value at the locations where the sensors are positioned. To evaluate the accuracy performances of the digital twin prediction, I used the error function described by equation 2.3. In Figure 3.3, I report the estimation and prediction results for the health state $\mathbf{z}$ and the behaviour of the error function.



(a) Prediction of the health state at $t_c = 40$.

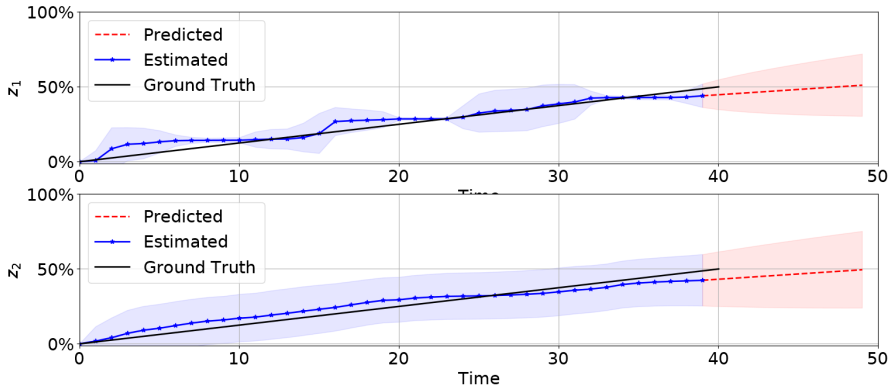


(b) Error at $t_c = 40$.

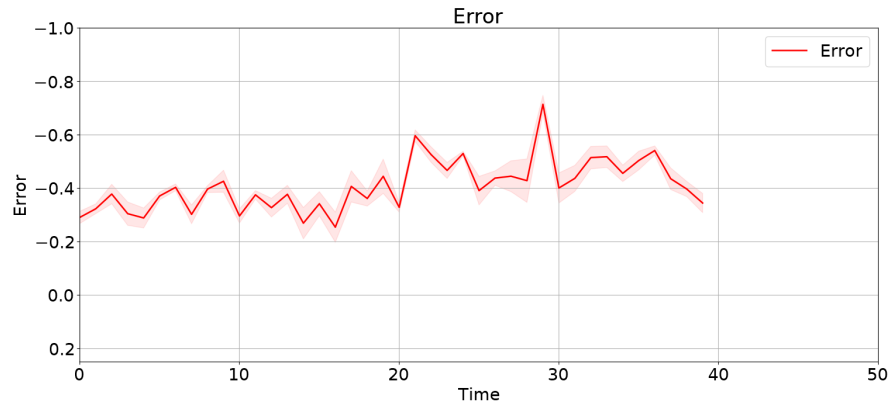
Figure 3.3: Data Augmentation using a ROM: estimated and predicted evolution of the health state and prescribed ground truth model; below is reported the trend of the relative error between estimated quantities of interest and their measured counterparts.

Figure 3.3 shows that when a linear model is introduced to describe the evolution of the structural health, the accuracy of the digital twin worsens when the health state z takes values far from those contained in the input dataset. To improve the estimation and prediction capabilities of the digital twin, I chose to perform a data augmentation of the dataset provided to the DBN. This is done by refining the possible states that can be assumed by the structural health parameters as $\{0\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%\}$. For each of the new possible configuration, I compute the value of the stress values using the Galerkin Reduced Basis method described in 3.2.1, to obtain an augmented input dataset for the digital twin. Once the data augmentation is completed, I use the digital

twin to estimate and to predict the health state $\mathbf{z}$; the corresponding results are reported in Figure 3.4.



(a) Prediction of the health state at $t_c = 40$ using data augmentation.



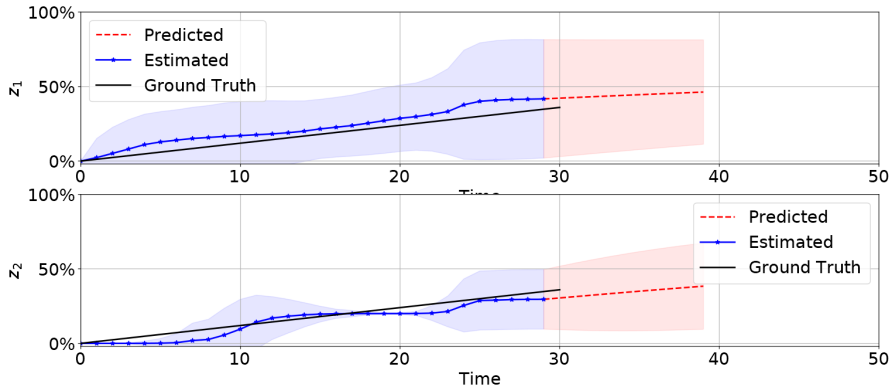
(b) Error at $t_c = 40$ using data augmentation.

Figure 3.4: Data Augmentation using a ROM: estimated and predicted evolution of the health state and the prescribed ground truth model; below is reported the trend of the relative error between estimated quantities of interest and their measured counterparts. The size of the input dataset is increased using a ROM.

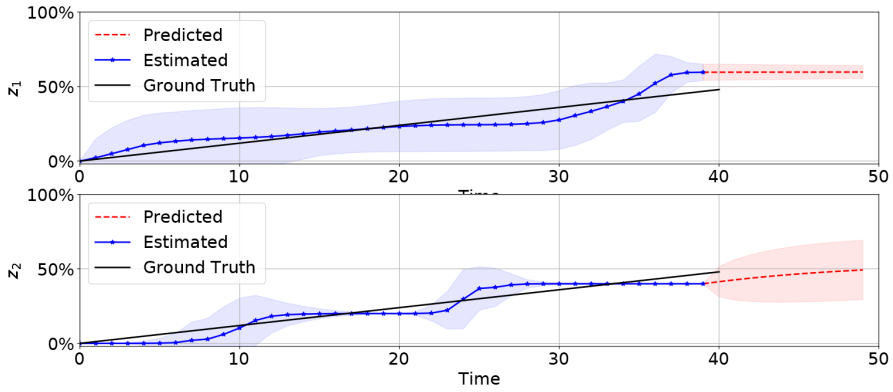
Looking at the error reported in figure 3.4b, I can see that this is lower in modulus then the one reported in figure 3.3b, and also that the estimation and prediction of the health state improved in this new test case. Therfore, data augmentation on the input dataset can helps the dynamic Bayesian network to improve its performances at the price of mild increase of computational time.

3.3.2. Test 2: Online input dataset creation using a ROM

In this test case, I consider as health state a vector $\mathbf{z} = \{0\%, 20\%, 40\%, 60\%\} \times \{0\%, 20\%, 40\%, 60\%\}$; the two components quantify the percentage reduction in material stiffness of two zones of interest. In this test case I use the ROM described in section 3.2.1 to build the input dataset for the dynamic Bayesian network; in particular the user, before using the digital twin to make prediction on the health state of the beam, can select the the two zones of interest which the component of $\mathbf{z}$ are referred to, as well as the two weights which correspond to the actions `SW` and `HW`. The creation of the input dataset in an online phase allows to use the dynamic Bayesian network to make predictions while a ROM allows to obtain this dataset in a short time. In particular, I choose a test case in which I consider the third and the sixth subregion of the beam as regions of interest and I defined the action `SW` and `HW` as placing a weight equal to 100 kg and 150 kg on the midspan of the beam, respectively. In this new test case, I consider as quantity of interest the beam vertical displacement at the sensors position, instead of the stress. In figure 3.5, I report the results regarding the estimate and prediction of the health state of the beam:



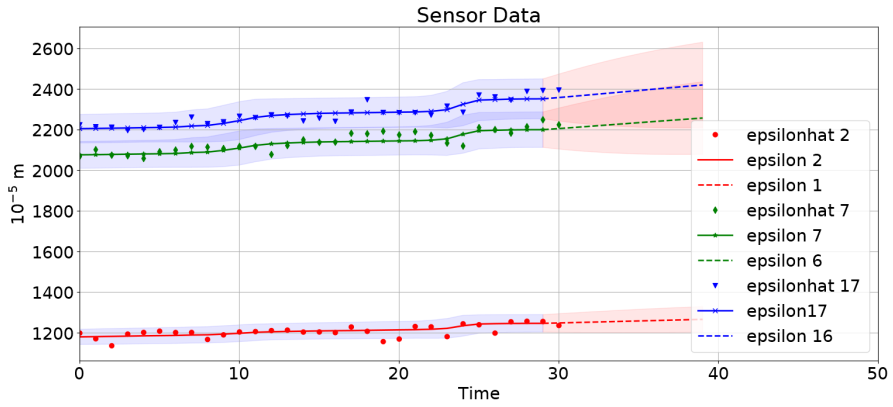
(a) Prediction of the state at $t_c = 30$.



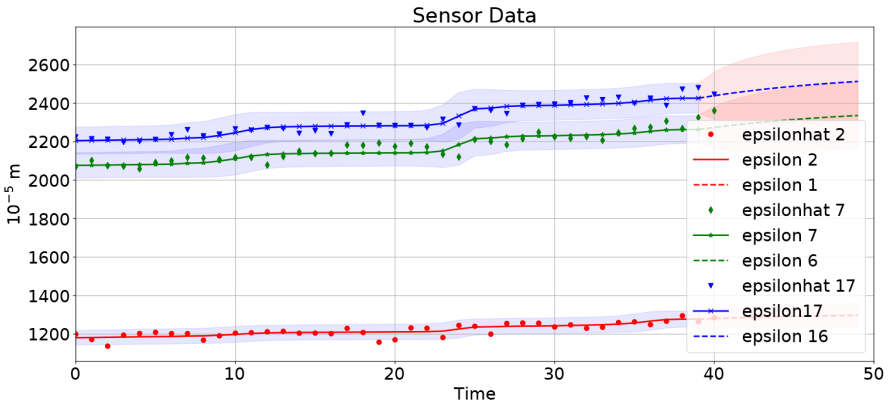
(b) Prediction of the state at $t_c = 40$

Figure 3.5: Online input dataset creation using a ROM: estimated and predicted evolution of the health state and prescribed ground truth model.

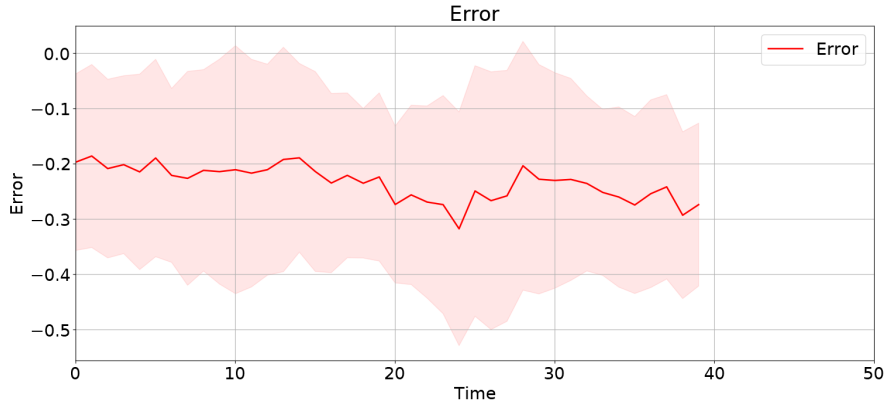
The estimate of the health state is rather good. Also in this case, I reported the results at two different time steps of the simulation in order to check that the prediction in future is reliable. In figure 3.6, I report the estimate and prediction results for the quantity of interest. The estimate and prediction of the displacement of the beam is rather accurate also in this case.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



(c) Error during simulation time interval

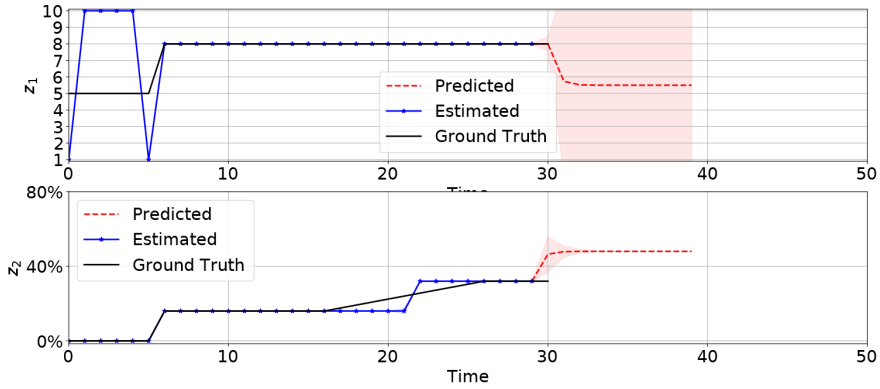
Figure 3.6: Online input dataset creation using a ROM: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

These results allow us to validate the framework described in Figure 1 and the online creation of the input dataset enables the user to study a wide range of test cases using few computational resources.

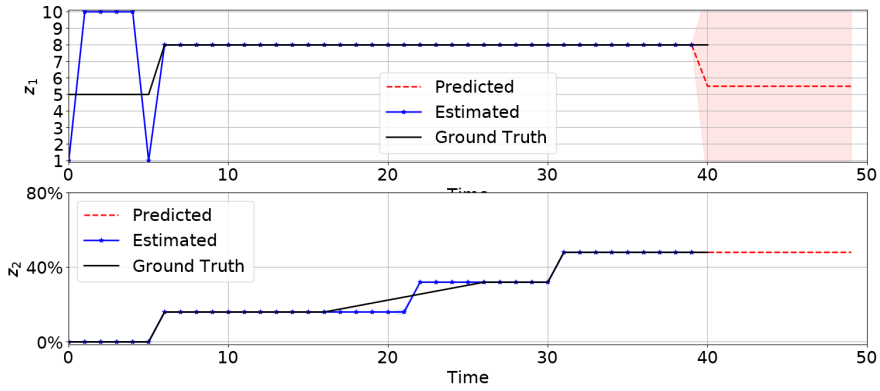
3.3.3. Test 3: Localization of damaged subregions using the DBN

In this third test case, I only consider one region of interest where a reduction in material stiffness can take place. Unlike the previous cases, I suppose that the position of the

region of interest is not known and the dynamic Bayesian network has to be used to estimate the position of this region. To this aim, I have to redefine the health state vector $\mathbf{z}$, with its first component representing the position of the region of interest while the second one representing the percentage of material stiffness reduction in this zone. So we have that $\mathbf{z} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \times \{0\%, 20\%, 40\%, 60\%\}$. It is important to notice that this test case represents a real case scenario, indeed usually the position of a structural damage is not known a priori. Once again, as quantity of interest measured by the sensors I consider the stress of the beam and the weights which correspond to the actions `SW` and `HW` are respectively equal to 100 kg and 150 kg. In Figures 3.7, I report the results concerning with the health state estimate and prediction and the behavior of the error.



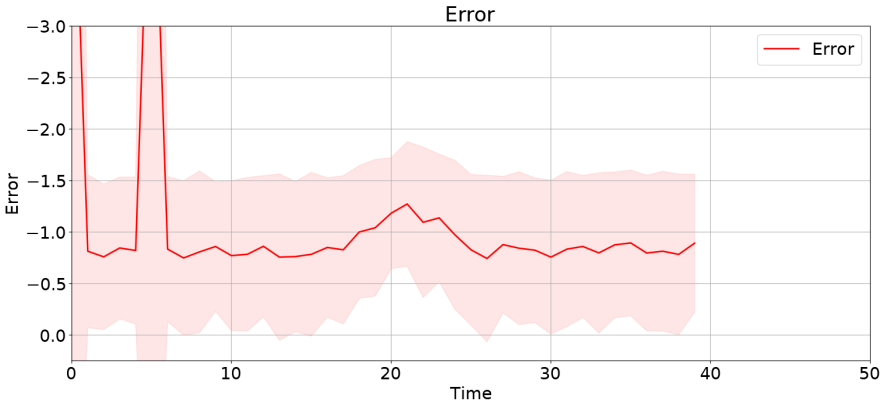
(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.

Figure 3.7: Localization of the damage location: estimated and predicted evolution of the health state and prescribed ground truth model.

Looking at figure 4.5a and 4.5b, we can see that the estimate of the percentage reduction in material stiffness is more or less accurate in all the simulation intervals. The situation is different for the estimation of the location of the zone of interest, i.e. the component $\mathbf{z}_1$, indeed we have a precise estimate only for $t > 5$ when the component $\mathbf{z}_2$ starts increasing. This is a quite expected outcome, since there is no sensitivity with respect to $\mathbf{z}_2$ if $\mathbf{z}_1$ is equal to 0%. The prediction for future time step is characterized by a lot of uncertainly, indeed in the above Figures we report a very big 95% confidence interval. The behaviour of the error is reported in figure 3.8, showing that it is very large in modulus for the time steps $t < 5$ when the prediction of the first component of $\mathbf{z}$ is not accurate.



(a) Error during simulation time interval

Figure 3.8: Localization of the damage location: relative error between estimated quantities of interest and their measured counterparts.

3.3.4. Test 4: inaccurate ROM effects on DBN predictions

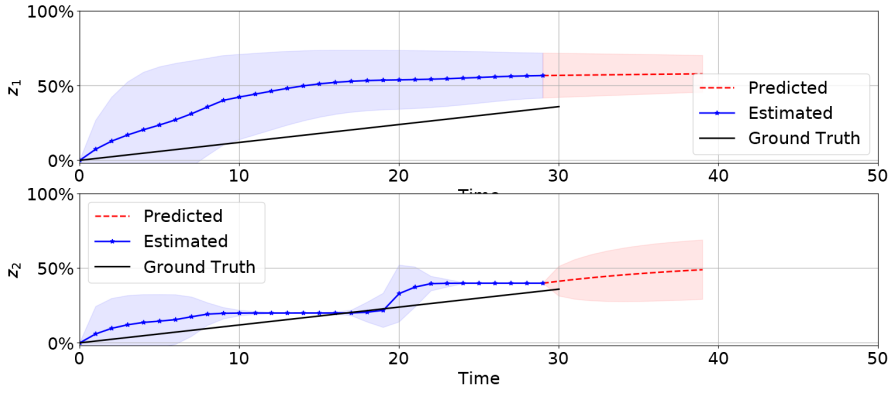
In this test case, I want to analyze how the accuracy of the ROM affects the results of the digital twin. In particular as explained in the flowchart reported in figure 1, I assume that the observation O_t are obtained thought a FOM while the input dataset is obtained with a ROM. The input dataset for the DBN requires a large amount of numerical simulations. If these simulations are characterized by a high number of degrees of freedom, the creation of the input dataset requires a lot of computational time. For this reason, is important to rely on a method that enables to reduce the computational complexity of this process. In particular, in this test case, I want to analyze the behaviour of the digital twin in the case in which an inaccurate ROM is used to create the input dataset. I obtained this ROM following the same procedure described in subsection 3.2.1, but selecting a tolerance equal

to $\epsilon_{POD} = 3 \times 10^{-1}$; the resulting Galerkin RB model only features $N = 5$ basis functions. The accuracy and computational time entailed by the ROM are reported in table 3.2.

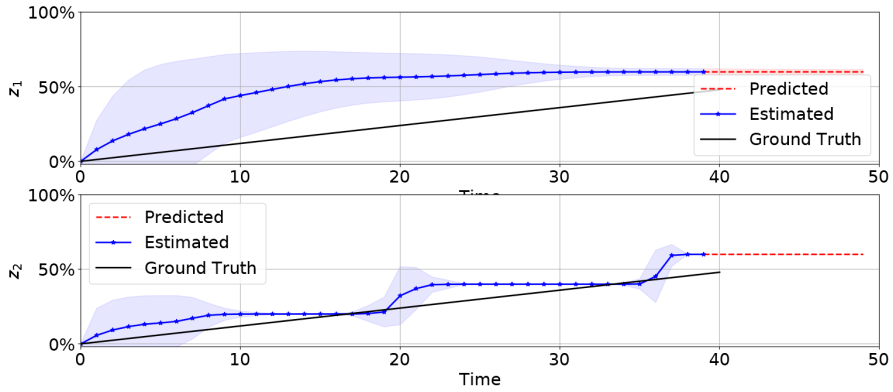
	FOM	ROM
DOF	4547	5
Time	4.47×10^{-1}s	6.24×10^{-3}s
Error	-	19.887 %

Table 3.2: Comparison between the performances and the accuracy of the FOM and the Galerkin RB method with $N = 5$ basis.

As in the previous case, the computational times refer to the mean assembling and solving time while the error is the relative error between the ROM and FOM solutions. This new reduced order method is not accurate but features a very little computational cost indeed it is two time faster than the ROM adopted in subsection 3.2.1 and 70 time faster than the high fidelity FOM. Once I obtained this new reduced order model, I want to use it to create the input dataset of the Dynamic Bayesian Network; in particular, I consider the same physical asset described in 3.3.2, to compare how the ROM accuracy impacts on the prediction results of the digital twin. In figure 3.9, I report the prediction results for the health state of the beam in this new.



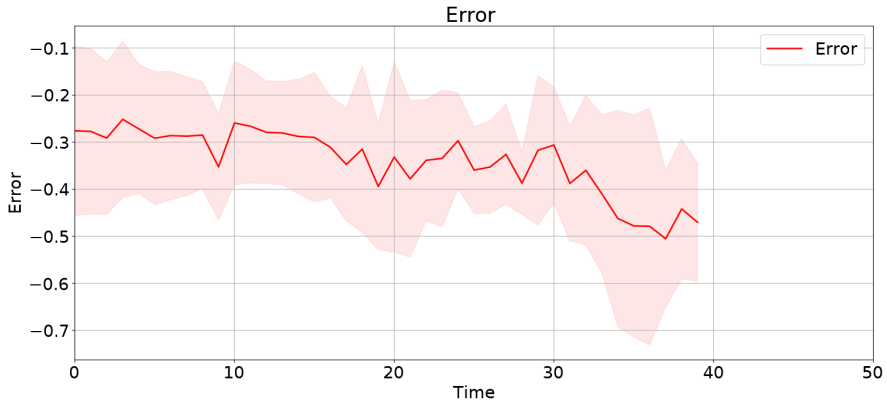
(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.

Figure 3.9: Effects of an inaccurate ROM: estimated and predicted evolution of the health state and prescribed ground truth model.

We can notice that the predictions and estimates are worse than those obtained in section 3.3.2. In particular, the first component of the health state vector $\mathbf{z}$ is overestimated for all the simulations. In figure 3.10, I report the error on the estimation of the quantity of interest.



(a) Error during simulation time interval

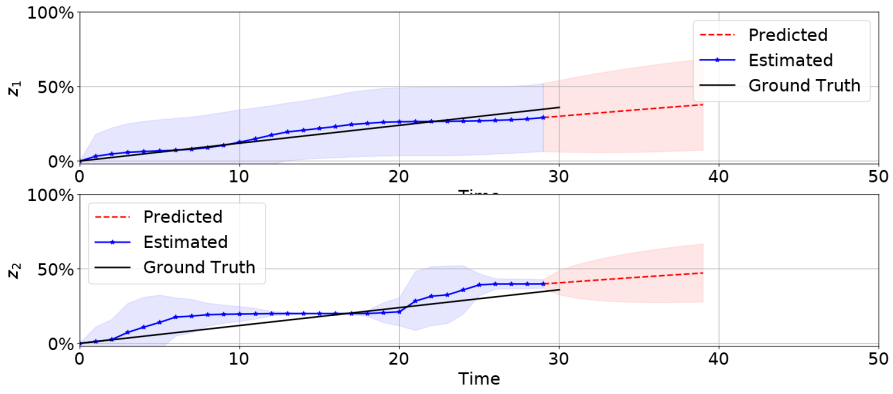
Figure 3.10: Effects of an inaccurate ROM: relative error between estimated quantities of interest and their measured counterparts.

Therefore using a reduce order method to create the DBN input dataset is useful to reduce the computational cost of the simulation; however, an accurate ROM is required

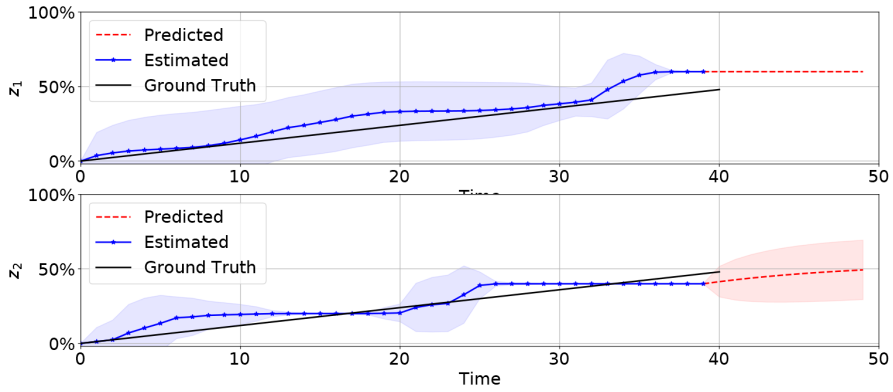
to guarantee the performance of the digital twin in terms of estimation and evolution prediction of the parameters affecting the digital state and the quantity of interest.

3.3.5. Test 5: poor calibration effects on DBN predictions

In this test case, I want to show how a bad calibration can affect the prediction and the estimate results of the digital twin. This test case is useful because sometimes is not possible to calibrate the parameters of the digital state. I suppose to have the same physical asset described in section 3.3.2 so that I can compare the two results. In this test case, the only parameter that has to be calibrated is the Young modulus scaling factor. I suppose that the only information that I have on e is that its distribution is a Gaussian distribution with mean equal to one and variance chosen in such a way the 95% credible interval is equal to the 5% of variability. I create the input dataset by drawing 30 samples from the prior distribution and using the RB Galerkin method described in section 3.2.1. In figure 3.11, I report the results regarding the estimation and prediction of the health state vector $\mathbf{z}$.



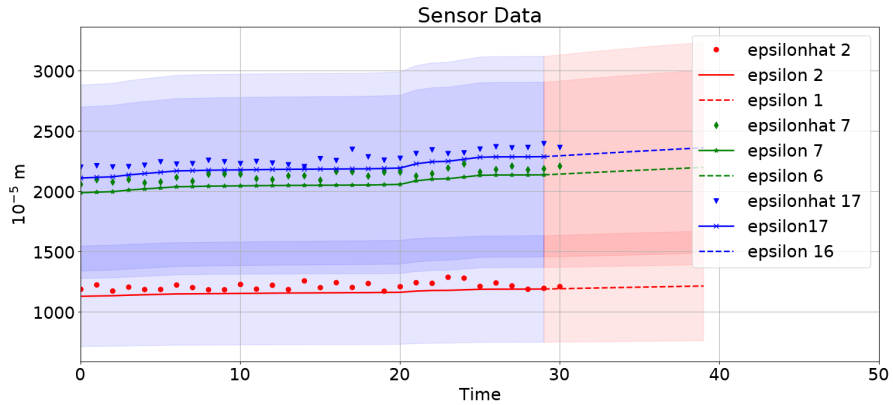
(a) Prediction of the state at $t_c = 30$.



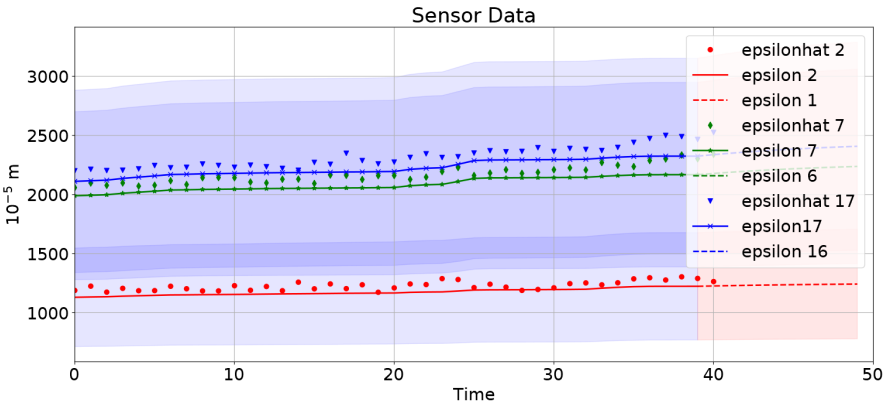
(b) Prediction of the state at $t_c = 40$

Figure 3.11: Poor calibration effects: estimated and predicted evolution of the health state and prescribed ground truth model.

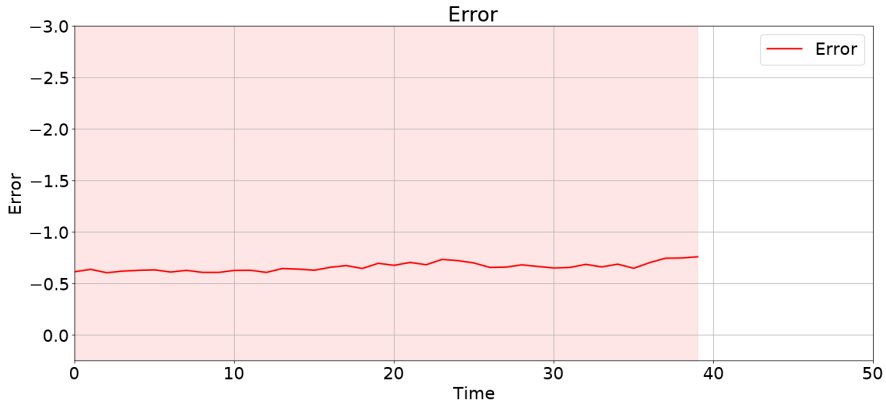
We can see that a bad calibration does not influence on these results, indeed the prediction obtained is very similar to the one shown in figure 3.5. In figure 3.12, I report the results regarding the prediction of the displacement measured by the sensors.



(a) Prediction of the state at $t_c = 30$.



(b) Prediction of the state at $t_c = 40$.



(c) Error during simulation time interval

Figure 3.12: Poor calibration effects: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps and their relative error.

We can therefore concludethat poor calibration does not negatively affect the estimate and prediction of the **z** health state, but it only has a negative influence on the estimate and prediction of the quantity of interest. In fact we can see that the uncertainty bands in Figure 3.12 are much larger than those reported in Figure 3.6, where a more accurate calibration was performed.

4 | An application: vibration-based structural health monitoring

In this chapter, I analyze a new test case involving a portal frame structure that is excited by random forces, applied on its surface to mimic the action of low-intensity seismic loads, such as ambient vibrations. The main difference between this new test case and the previous ones is that I no longer consider a static case but a dynamic one. I will use this test case to verify the performance of the Bayesian network in this dynamic case where the forces corresponding to the different control actions are not known but are instead random. Another substantial difference between this test case and the previous ones lies in the choice of the quantity of interest considered: I no longer consider a directly measured quantity such as stress or displacement, but indirect quantities such as the first four eigenfrequencies.

4.1. Problem description

This test case consists in monitoring the health state of a four-story portal frame. This structure has a total height $h = 6\,\text{m}$, length $l = 0.3\,\text{m}$ and width $w = 0.4\,\text{m}$. I describe the the response of the structure under the applied loading by solving the equation (1.14) in a time interval $(0{,}5\,\text{s})$ with a finite element method implemented in [19], in particular for the time discretization I used the second order Newmark scheme choosing the coefficients $\beta = 0.25$ and $\gamma = 0.5$ yielding an unconditionally stable method; for further details see [20]. The equation (1.14) is solved homogeneous Dirichlet boundary condition on the two faces resting on the ground, while a pressure condition $h(y,t) = N(t)\mathbb{1}\{2.7 < y < 3\} + 2N(t)\mathbb{1}\{5.7 < y < 6\} + 3N(t)\mathbb{1}\{8.7 < y < 9\} + 4N(t)\mathbb{1}\{11.7 < y < 12\}$ is imposed on the right lateral face. The latter describes an activation function resulting in distributed forces applied in correspondence of the four stories, with a modulation coefficient that increases linearly with the height. $N(t)$ is a Gaussian signal of mean 0 and variance 1,

which has been attenuated with a low-pass filter featuring a cutoff frequency of 20 Hz. The Lamè coefficients are computed considering a Young modulus equal to 30 GPa and a Poisson ratio equal to 0.2, while the density of the structure is equal to 2500 $\mathrm{Kg/m^3}$. Each time-step of the in time interval $[0, T]$ considered by the dynamic Bayesian network corresponds to a 5 s simulation. To model the health state of this system I considered a vector $\mathbf{z}$ of two components which represent the percentage of material stiffness reduction of two of the ten zones located at the ends of the four stories, in proximity of the clamped sides. Figure 4.1 shows the four-story portal frame and specifically highlights the zones that could be damaged.

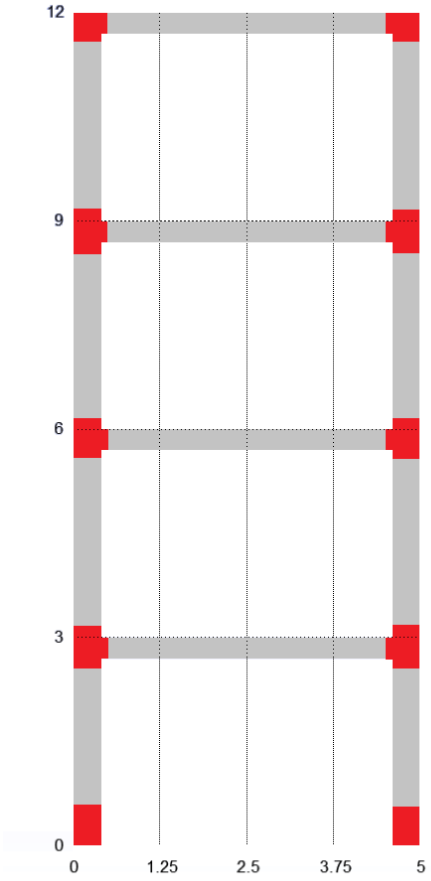


Figure 4.1: Four-story portal frame structure, the zones that could be damaged are highlighted in red.

The structure is monitored by considering as quantity of interest the first four eignefrequencies of the structure instead of stress or displacement measurements. To compute these quantities, I use the frequency domain decomposition method using the horizontal

and vertical displacement signals measured by eight different sensors, as detailed in subsection 4.1.1. It is important to notice that in this new test case the input dataset for the digital twin contains less data for each health state situation. Therefore, test case is also useful to show how the performance and accuracy of the Bayesian network vary as a function of the number of data it receives as input. In figure 4.2, I reported the four-story frame in a deformed configuration.

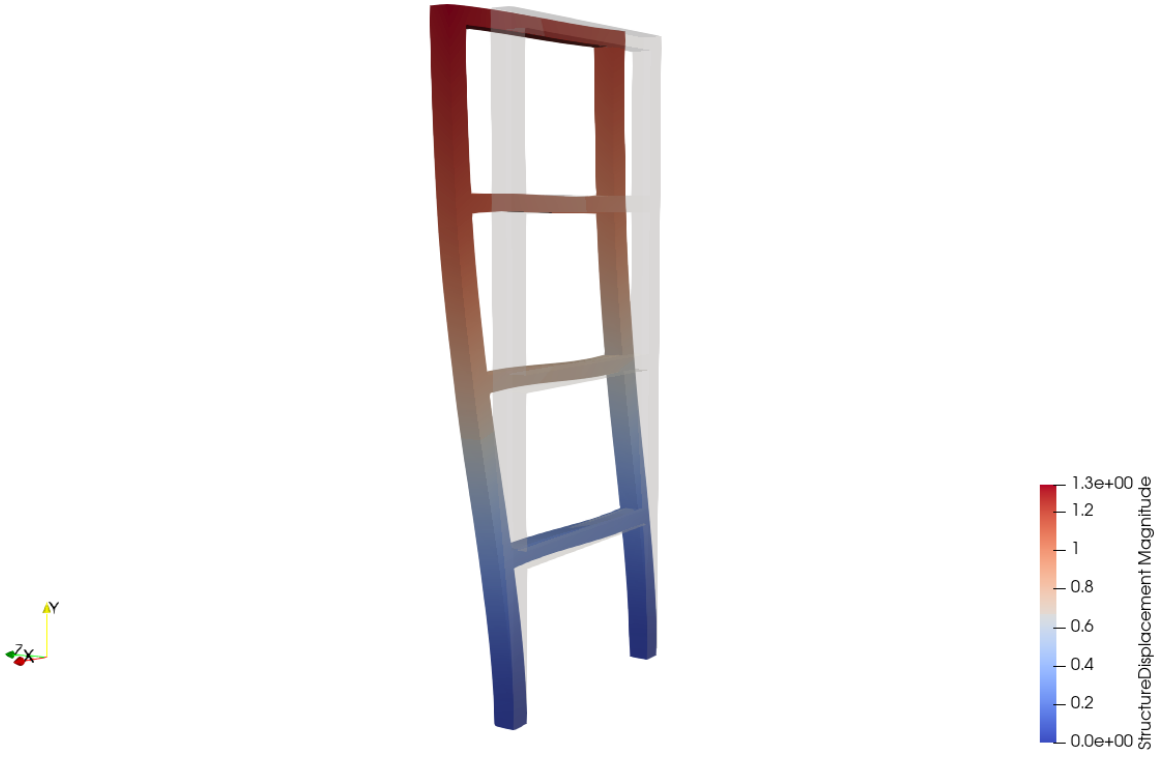


Figure 4.2: Four-story frame displacement contour plot at time $t = 5\,\mathrm{s}$.

The final aim of this test case is to monitor the health state of this kind of frame using the Bayesian network; to this aim, we need to parameterize the digital state D_t, which needs to include geometrical, material, and health parameters. The digital state is described by the following vector of parameters:

$$\mathbf{d} = \left[\; l, \quad h, \quad w, \quad n, \quad e, \quad \mathbf{z} \; \right]^T. \tag{4.1}$$

Where l, h, w are geometrical parameters describing the dimensions of the frame, n is the number of stories, e is a material parameter adopted to rescale the concrete Young's modulus, and $\mathbf{z}$ is a vector of structural health parameters describing the health state of the

structure as previously explained. Once again this vector of parameters is similar to the one used in [14] except that now sensor masses are not considered since all measurements are obtained from numerical experiments.

4.1.1. Frequency domain decomposition

In this new test case, as previously said, as quantity of interest I considered the firsts four eigenfrequencies of the structure. These are computed with the frequency domain decomposition method (FDD), a powerful method of modal analysis that enables to compute the eigenfrequencies of a system knowing only its output. As an output-only algorithm, it is useful when the input data is unknown like in this test case in which the input is a random signal. The frequency domain decomposition is based on the SVD decomposition. In particular, the output of the system is rewritten as $y(t) = \Phi q(t)$ where Φ is the mode shape matrix; starting from this definition, we can compute the covariance of the output and by taking its Fourier transform we can compute the spectral density matrix evaluated at the frequency f, as follows:

$$G_{yy}(f) = \Phi G_{qq}(f)\Phi^T. \tag{4.2}$$

The FDD is a method based on the singular value decomposition of the spectral density matrix, that reads as:

$$G_{yy}(f) = U(f)SV(f)^T, \text{ with } S = diag(s_1, ..., s_n), \tag{4.3}$$

S is the diagonal matrix of singular values while U is the matrix of singular vectors. It is important to remark that the singular vector matrices are a function of the frequency f. Once the SVD decomposition of $G_{yy}(f)$ is found, we can plot the singular values against their corresponding frequency lines, and by picking the peak values we can estimate the natural frequencies. Finally, the mode shape vectors are estimated from the singular vectors corresponding to each identified natural frequency; for example, the mode associated to the natural frequency f_0 is computed as $\phi = u_1(f_0)$. For further details about the frequency domain decomposition method see [4]. In figure 4.3, I report the plot of the first singular values against frequency underling the firsts four peaks selected to find the firsts four eigenfrequency.

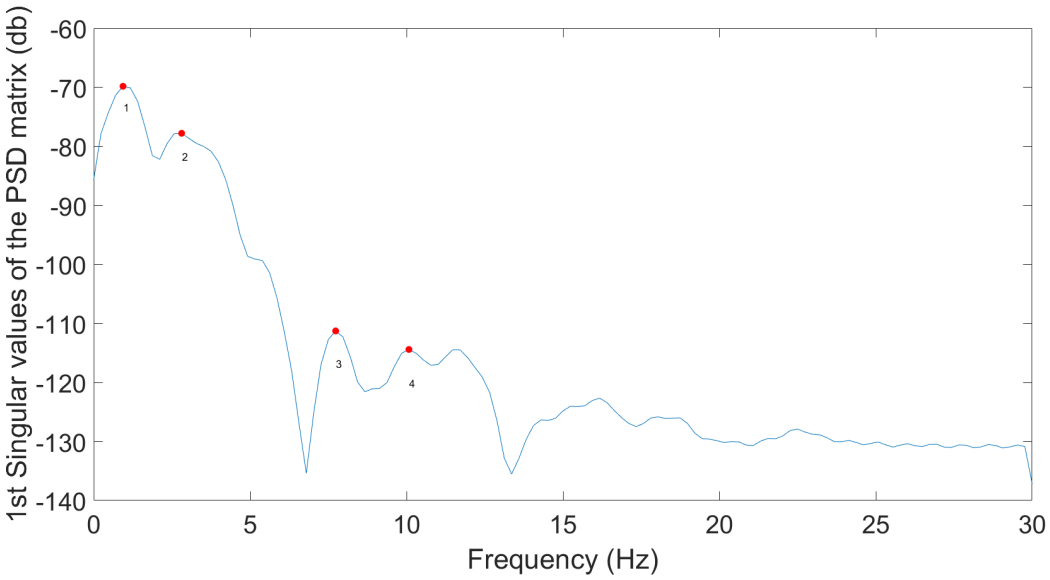


Figure 4.3: First singular value with red dots over the first four peaks selected.

As mentioned before, the output analyzed by the FDD method is a signal measured by eight different sensors placed on the frame and capable of measuring both the vertical and horizontal displacement of this structure. Specifically, I placed one sensor at the midspan of each story, and an additional sensor at the extremity of each story, and four more sensors near the areas where the force acts. It is also important to note that these frequencies are not a directly measured quantity such as the displacement or the stress, but are indirect quantities, so it is interesting to see how the Bayesian network will behaves when receiving them as input data.

4.2. Calibration phase

In this section, I will describe how the calibration process of the parameters contained in the vector $\mathbf{d}$ is carried out. As said before, this phase is crucial in order to have a precise description of the physical asset and to obtain a digital twin at-scale. The calibration phase is performed at the first time step of the digital twin simulation. As in chapter 2 I adapted the code already exploited and developed in [12] to the four-stories frame. This process is done at $t = 0$ when the frame has a good health state, $\mathbf{z} = 0$; by using observation coming from the physical state the other parameters contained in $\mathbf{d}$ are calibrated. For what concern the geometrical parameters l, h, w, n, without any measurement of these quantities I simply create the frame mesh fixing $h = 6\,\text{m}$, $l = 0.3\,\text{m}$,

$w = 0.4\,\mathrm{m}$ and $n = 4$, so that only the Young modulus scaling factor e, which is related to the material properties of the frame considered has to be calibrated. To this goal, as in [14], I used a particle filter algorithm. The prior information of this parameter is a Gaussian with unit mean and variance is chosen in such a way the 95% credible interval equal to 5% of variability. To use the particles filter algorithm is crucial to find a relationship between the Young modulus scaling factor and the elastic modulus k, proceeding as in Chapter 2, the relationship found in this case is

$$k = 0.0012478e \tag{4.4}$$

Then, the prior information update takes place using the particles filter algorithm with 10^5 particles, this number of particles is chosen in such a way the variance of the posterior is smallest as possible. In figure 4.4, I reported the result of this algorithm, showing with a blue line the prior and with a black line the posterior.

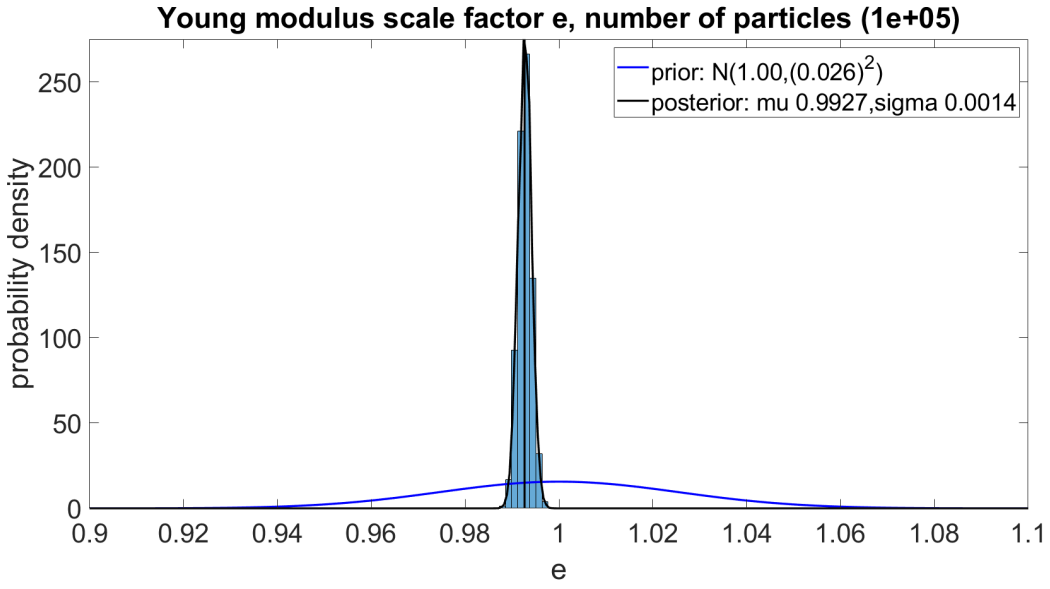


Figure 4.4: Prior and posterior distributions of the Young's modulus scaling factor for the frame test case.

Figure 4.4 reports the obtained posterior probability distribution, which is centered at 0.9927 (corresponding to a 0.73% reduction of e) and features a much reduced standard deviation compared to the prior distribution. Once this parameter is calibrated, we have to create the input dataset for the Bayesian network. In particular, we have to sample 30 values from the posterior distribution and for each of these samples the first four natural

frequencies are computed using the FDD method, for each possible damage state $\mathbf{z}$.

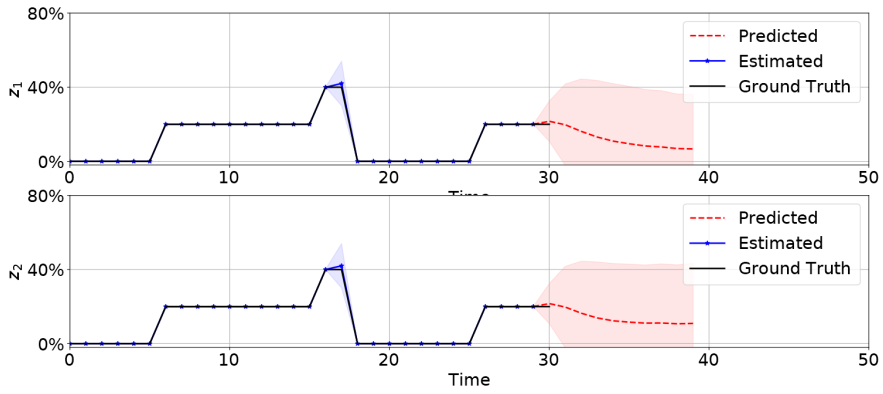
4.3. Operational phase

After having calibrated the parameters in $\mathbf{d}$, the operational phase, where the components of $\mathbf{z}$ are predicted, can start. As in the beam test case, the positions of the two regions that can be damaged by the random force applied to the lateral face of the frame are supposed to be known. These two zones are located in correspondence of ends of the first two inter-planes and I suppose that the Young modulus of the other regions of the beam remains unchanged. The nodes which characterize the Bayesian network for $t > 0$ are:

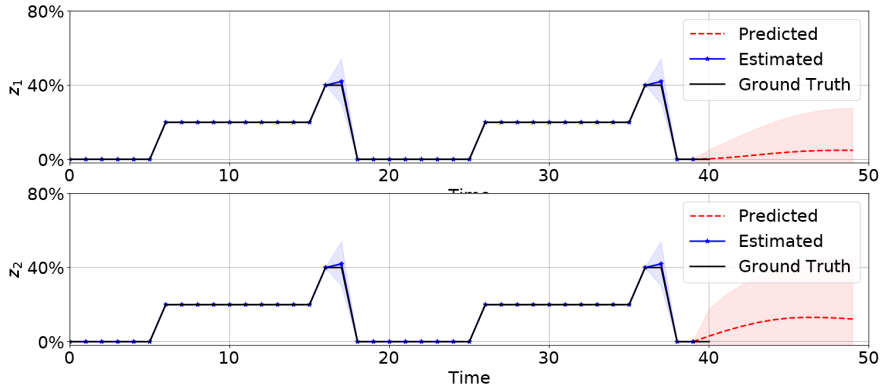
- O_t, the natural frequencies computed at time t using the FDD approach;
- D_t, the digital state at time t characterized by the parameters $\mathbf{d}$;
- Q_t, the quantity of interest which approximates the natural frequencies at time t;
- U_t, the action acted on the frame at time t.

At each timestep, U_t can be equal to `h`, which corresponds to the situation in which a random signal stresses the frame, or it can be `r`, which corresponds to the situation in which the frame is repaired and it returns to its original health state, $\mathbf{z} = 0$. As in Chapter 2, reward functions are not introduced and a policy is fixed. This policy consists of repairing the frame, by taking the action `r`, in the case where at least one of the two areas of interest has more than 30% damage; in all other cases, the random surface force is assumed to act on the lateral face of the frame, so that the action `h` is taken. It is important to remark that the test case considered is very different from those considered previously; in fact, now the weight corresponding to the control action `h` is not known and the quantity of interest considered is not measured directly by the sensors placed on the frame, but it is an indirect quantity. In this test case, the health state $\mathbf{z}$ represents the percentage of material stiffness reduction in two distinct zones, as $\mathbf{z} = \{0\%, 20\%, 40\%, 60\%\} \times \{0\%, 20\%, 40\%, 60\%\}$; where the first component is referred to the zone near the first story, while the second component is referred to the zone near the second story. $\Phi_t^{dynamics}$ is defined in such a way that when the action `h` is played we have a worsening probability equal to 0.05; instead when `r` is played, the frame returns to the state $\mathbf{z} = 0$ with probability one. Regarding the evolution of the health state $\mathbf{z}$, a step-wise ground truth model is considered, involving that when $t > 5$ the health state of the two areas of interest increases by 20% every ten time steps. In figure 4.5, I report the health state estimation and prediction results in a time simulation $(0, 40)$; in particular, I reported the estimated together with the prescribed health state dynamics

and the predicted health state in future timestep. As prediction interval, an interval $(0, 10)$ is considered. We can see the accuracy of the Bayesian network is good, as small uncertainties only arise when the action `r` is played. These results allow us to conclude that the digital twin considered succeeds in predicting well the health status of the frame even in the case where few data are observed, in fact only the first four natural frequencies are considered, even if they are not directly measured quantities from the frame but are indirect quantities.



(a) Prediction of the state at $t_c = 30$.

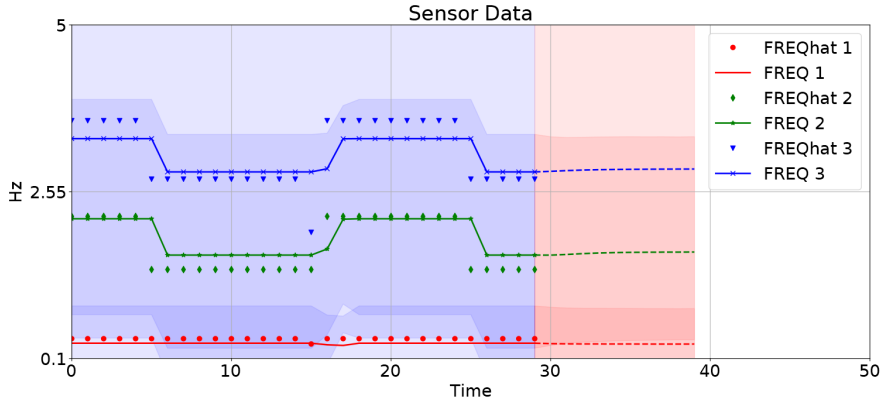


(b) Prediction of the state at $t_c = 40$.

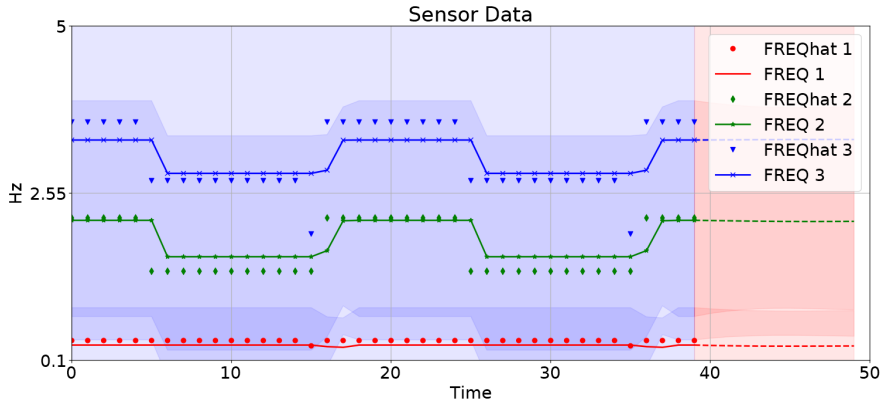
Figure 4.5: Vibration-based structural health monitoring: estimated and predicted evolution of the health state and prescribed ground truth model.

In figure 4.6, I report the results concerning the accuracy of the estimate and the prediction of the first four natural frequencies. In this case, the prediction of these quantities turns

out to be more affected by uncertainty; in fact, the confidence interval bands turn out to be very large.



(a) Prediction of the natural frequencies at $t_c = 30$.



(b) Prediction of the natural frequencies at $t_c = 40$.

Figure 4.6: Vibration-based structural health monitoring: estimated, predicted evolution and measured values of the quantity of interest with reference to two different current time steps.

From these results, we can conclude that the Bayesian network can be used to estimate the health state of a structure even in more complex cases where directly measured quantities are not available, the force acting on the structure is not known, and in situations where the transient period between the application of two forces is not neglected.

5 | Conclusions and future developments

Through the tests proposed in this thesis, I wanted to investigate the accuracy and performance of the Bayesian dynamical network proposed in [14] as a digital twin method. As proved from the collected results, this method can be reliably used in several contexts to estimate the health state of a structure. In particular, this strategy for building digital twins has been used to estimate structural health parameters that could correspond to specific causes of damage, for example, the length of a crack or the percentage reduction in material stiffness in a zone of interest. As seen by the tests described in Sections 2.3.1 and 3.3.1, the probabilistic graphical model allows estimating the health state in both these situations. Through these test cases, I also showed that the accuracy of this method is not affected by the observation O_t it receives as input; in these test cases, different quantities of interest such as Von Mises stresses, structural displacements, or the eigenfrequencies of the monitored structure were indeed used. This allowed me to conclude that the method works correctly whether we have directly observed quantities or indirectly observed quantities.

The other goal of my thesis was to extend the range of applications of the Bayesian network to more complex situations than the unmanned aerial vehicle (UAV) test described in [14]. I wanted to study the behavior of this method in the case where more than two control actions are applied. As reported in Section 2.3.3, the digital twin is accurate even in the case where there are three control actions. In the UAV test, strong hypothesis was stated, assuming to know the area of interest where the UAV health worsened. In the test described in Section 3.3.3, I investigated the use of the Bayesian network to locate this region of interest and to study the evolution of the parameters describing the worsening of the structural health. I also investigated what happens if the health state of the system is described by more than two parameters. In Section 2.3.4, I reported a case in which the health state is described by three parameters, finding that the method is still accurate. Among several attempts made, I also tried to consider more than three param-

eters; however, the belief propagation algorithm did not converge in these cases and the simulation of the digital twin crashed after a few timesteps. In the last test case reported in section 4.3, I also demonstrated how this method can be used in a dynamical test case, and when the control input applied on the structure is not known. The latter test case is much more complex than the UAV test case in which the forces characterizing the control action are known and the transient period between the application of two forces is ignored.

As we have seen in all these cases tests, before using the Bayesian network to estimate the health parameters of the system, we had to calibrate the parameters that describe the digital state using the particle filter algorithm. As we have seen in the test case reported in Section 3.3.5 the calibration phase is crucial to obtain good estimates of the quantity of interest. Once the calibration phase is carried out, we sample from the posterior distribution of the calibrated parameters, and for each of these samples, observations are computed for each possible damage state and for each control action. Creating this input dataset using numerical simulations can be very expensive from a computational point of view. For example, to create the input dataset of the test described in Section 2.3.1, 960 simulations with a full order model would be required. To reduce the relative computational burden, it is of key importance to introduce a suitable ROM. As we have seen in Chapter 3, the combined use of Bayesian networks and reduce order models allowed us to create the input dataset in a very short time, on the order of ten seconds, and to perform a data augmentation on the input dataset, useful to ensure a greater accuracy of the digital twin. It is important to remark that a fairly accurate ROM is required to get accurate predictions from the DBN as we have seen in the test reported in Section 3.3.4.

The main problem of this method is that it has been reported so far to be not suitable to estimate more than three parameters. It would therefore be interesting to investigate how to design the Bayesian network in these cases. A further problem with this method is that it is computational expensive to be used to estimate parameters that describe the health state of a system in simulations where there are many time steps. In fact, the time taken to calculate the joint probabilities increases with the increase of time steps. The latter fact is due to the increase of the dimension of the considered graph with the simulation time; indeed, at each time step new nodes representing the digital state, the control action, the quantity of interest and the observation are added to the graph. In figure 5.1, I report the behavior of the computational time required to compute the joint probabilities varying the number of time step of the simulations for the test case described in Section 3.3.1. In particular, the computational time at $t = 41$ is equal to 17.052 s, while the computational time at the first time step is equal to 0.006 s.

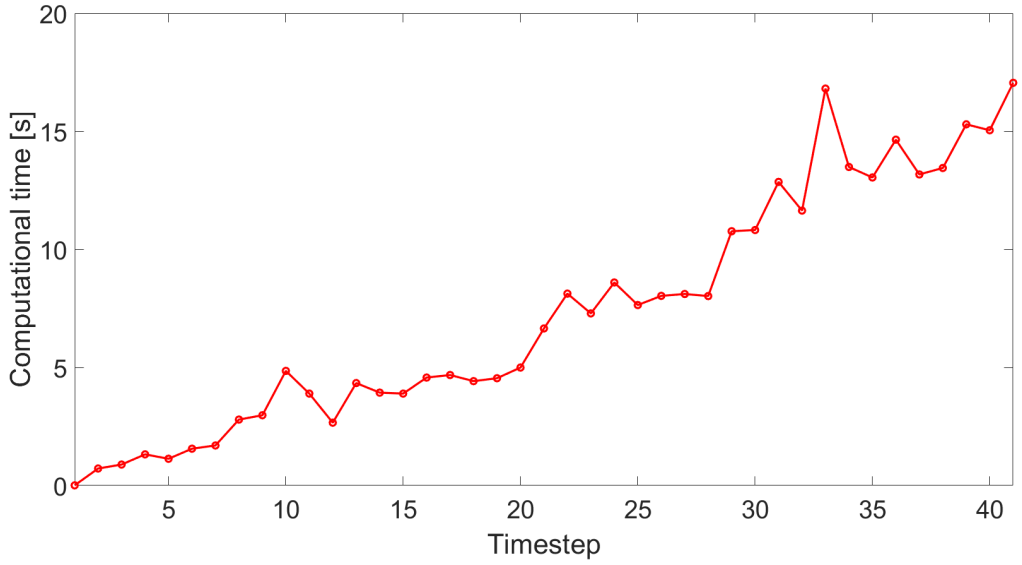


Figure 5.1: Computational time required by the Bayesian network to compute the joint probabilities at different times of the simulation.

The computational time behavior features this trend because no real dynamic Bayesian network is implemented in [12], but new nodes are added to the graph at each time step and the belief propagation algorithm is then applied to this new graph. One possible solution to improve the computational performance of the digital twin would be therefore to implement a true dynamic Bayesian network and update the information contained in the nodes referring to time t only by considering the nodes referring to time $t-1$.

In conclusion, all the tests described in the present thesis allowed to show the strength of dynamic Bayesian networks as enablers for digital twins. In the context of structural health monitoring, this method allows to build digital twins at-scale starting from a few input information such as the dimensions of the structure and its material properties.

Bibliography

[1] S. E. Azam and S. Mariani. Online damage detection in structural systems via dynamic inverse analysis: A recursive bayesian approach. *Engineering Structures*, 159:28–45, 2018.

[2] P. Bauer, B. Stevens, and W. Hazeleger. A digital twin of earth for the green transition. *Nature Climate Change*, 11(2):80–83, 2021.

[3] I. Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 1, 2008.

[4] R. Brincker, P. Andersen, and N.-J. Jacobsen. Automated frequency domain decomposition for operational modal analysis. In *Conference proceedings: IMAC-XXIV: A conference & exposition on structural dynamics*. Society for Experimental Mechanics, 2007.

[5] J. Corral-Acero, F. Margara, M. Marciniak, C. Rodero, F. Loncaric, Y. Feng, A. Gilbert, J. F. Fernandes, H. A. Bukhari, A. Wajdan, et al. The 'digital twin'to enable the vision of precision cardiology. *European heart journal*, 41(48):4556–4564, 2020.

[6] P. Dai and J. Goldsmith. Topological value iteration algorithm for markov decision processes. In *IJCAI*, pages 1860–1865, 2007.

[7] C. R. Farrar and K. Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[8] V. Giglioni, I. Venanzi, V. Poggioni, A. Milani, and F. Ubertini. Autoencoders for unsupervised real-time bridge health assessment. *Computer-Aided Civil and Infrastructure Engineering*, 2022.

[9] E. Glaessgen and D. Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd structures, structural dynamics and materials conference 20th adaptive structures conference 14th*, page 1818, 2012.

[10] G. Gobat, A. Opreni, S. Fresca, A. Manzoni, and A. Frangi. Reduced order modeling

of nonlinear microstructures through proper orthogonal decomposition. *Mechanical Systems and Signal Processing*, 171:108864, 2022.

[11] M. Jans-Singh, K. Leeming, R. Choudhary, and M. Girolami. Digital twin of an urban-integrated hydroponic farm. *Data-Centric Engineering*, 1:e20, 2020.

[12] M. Kapteyn. michaelkapteyn/UAV-Digital-Twin: Version 1.0 Release, Apr. 2021. URL `https://doi.org/10.5281/zenodo.4658878`.

[13] M. Kapteyn. michaelkapteyn/UAV-Experimental-Calibration: Version 1.0 Release, Apr. 2021. URL `https://doi.org/10.5281/zenodo.4658935`.

[14] M. G. Kapteyn, J. V. Pretorius, and K. E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *arXiv preprint arXiv:2012.05841*, 2020.

[15] M. G. Kapteyn, D. J. Knezevic, D. Huynh, M. Tran, and K. E. Willcox. Data-driven physics-based digital twins via a library of component-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 123(13):2986–3003, 2022.

[16] G. Kerschen, J.-c. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear dynamics*, 41:147–169, 2005.

[17] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[18] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang. Dynamic bayesian network for aircraft wing health monitoring digital twin. *Aiaa Journal*, 55(3):930–941, 2017.

[19] F. Negri. redbKIT Version 2.2. `http://redbkit.github.io/redbKIT/`, 2016.

[20] N. M. Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[21] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science*, 1(5):313–320, 2021.

[22] A. Quarteroni. *Modellistica numerica per problemi differenziali*, volume 2. Springer Science & Business Media, 2009.

[23] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[24] L. Rosafalco, M. Torzoni, A. Manzoni, S. Mariani, and A. Corigliano. Online structural health monitoring by model order reduction and deep learning algorithms. *Computers & Structures*, 255:106604, 2021.

[25] J. Schreiber. pomegranate. `https://github.com/jmschrei/pomegranate`, 2016.

[26] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu. A comprehensive review of digital twin–part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. *arXiv preprint arXiv:2208.12904*, 2022.

[27] M. Torzoni, A. Manzoni, and S. Mariani. Structural health monitoring of civil structures: A diagnostic framework powered by deep metric learning. *Computers & Structures*, 271:106858, 2022.

[28] Y. Ye, Q. Yang, F. Yang, Y. Huo, and S. Meng. Digital twin for the structural health management of reusable spacecraft: a case study. *Engineering Fracture Mechanics*, 234:107076, 2020.

A | Appendix A

The general pseudocode for the belief propagation algorithm for a factor graph $\mathcal{G}(X, F)$, where X are the variable vertices and F are the factor vertices which represent the conditional probabilities of the variable vertices, is as follows:

Algorithm A.1 Belief Propagation Algorithm

while not converged **do**
  for all $i \in X$ **do**
    for all $a \in \mathrm{N}(i)$ **do**
      if $\mathrm{N}(i) \setminus a = \emptyset$ **then**
        $\mu_{i \to a}(\mathbf{x}_i)$ is set to the uniform distribution.
      end if
      compute message update $\mu_{i \to a}(\mathbf{x}_i) = \prod_{k \in \mathrm{N}(i) \setminus a} \mu_{k \to i}(\mathbf{x}_i)$
      where $\mathrm{N}(i)$ is the set of neighboring factor nodes of i.
    end for
  end for
  for all $a \in F$ **do**
    for all $i \in \mathrm{N}(a)$ **do**
      if $\mathrm{N}(a) \setminus I = \emptyset$ **then**
        $\mu_{a \to i}(\mathbf{x}_i) = f_a(\mathbf{x}_i)$
      end if
      update message $\mu_{a \to i}(\mathbf{x}_i) = \sum_{\mathbf{x}'_a : \mathbf{x}'_i = \mathbf{x}_i} f_a(\mathbf{x}'_a) \prod_{v^* \in \mathrm{N}(a) \setminus i} \mu_{v^* \to a}(\mathbf{x}'_i)$
      where $\mathrm{N}(a)$ is the set of neighboring variable nodes of a.
    end for
  end for
end while
The marginal distribution of each variable nodes is $p_{\mathbf{x}_i}(\mathbf{x}_i) \propto \prod_{a \in \mathrm{N}(i)} \mu_{a \to i}(\mathbf{x}_i)$
The joint marginal distribution of the set of variables belonging to one factor is $p_{\mathbf{x}_a}(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{i \in \mathrm{N}(a)} \mu_{i \to a}(\mathbf{x}_i)$

This algorithm is implemented in [25] in method `predict_proba`.

B | Appendix B

In this appendix I report the algorithm used for compute the reduced basis using the proper orthogonal decomposition presented in section 1.3.2.

Algorithm B.1 POD Algorithm

1: Compute n_s snapshot using a FOM.
2: Fix a tolerance ϵ_{POD}
3: **if** $n_s \leq N_h$ **then**
4: Compute the correlation matrix $C = S^T S$.
5: Solve the eigenvalue problem $C\psi_i = \sigma_i^2 \psi_i$.
6: Set $\zeta_i = \frac{1}{\sigma_i} S\psi_i$
7: **else**
8: Compute the matrix $K = SS^T$.
9: Solve the eigenvalue problem $K\zeta_i = \sigma_i^2 \zeta_i$.
10: **end if**
11: Define N as the minimum integer such that $I(N) \geq 1 - \epsilon_{POD}^2$.
12: Compute the matrix $V = [\zeta_1 | ... | \zeta_N]$

This algorithm is implemented in [19], in particular in my test cases I used the function `VPOD_basis_computation`.

List of Figures

Acknowledgements

Innanzitutto, ringrazio il mio relatore Prof. Andrea Manzoni e il mio correlatore Dott. Matteo Torzoni, che mi hanno seguito con disponibilità e sono sempre stati pronti a darmi le giuste indicazioni in ogni fase della realizzazione dell'elaborato.

Ringrazio le due persone a cui voglio più bene, i miei genitori per aver sempre creduto in me e per avermi sostenuto in ogni occasione.

Ringrazio i miei nonni per l'amore che mi hanno saputo donare e per l'appoggio che non mi hanno mai fatto mancare.

Ringrazio i miei amici Samuele, Riccardo, Guglielmo, Carola, Matteo, Francesco, Lorenzo, Emanuele e Gloria che hanno alleggerito i momenti più pesanti e mi hanno sempre spronato a dare di più.

Ringrazio i miei compagni delle superiori che mi hanno accompagnato dal debito in matematica fino ad oggi.

Ringrazio i miei amici dell' università Riccardo, Mirko, Matteo, Roberto, Lorenzo e Laura per questi anni passati insieme.