Automation in Construction 140 (2022) 104333

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Structural digital twin framework: Formulation and technology integration

Manuel Chiachío [a,b,*], María Megía [a,b], Juan Chiachío [a,b], Juan Fernandez [a,b], María L. Jalón [a]

[a] *Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain*
[b] *Dept. Structural Mechanics & Hydraulics Engineering, University of Granada, 18071 Granada, Spain*

ARTICLE INFO

Keywords:
Digital twin
Petri nets
Bayesian learning
Internet of things
Structural health monitoring

ABSTRACT

This work presents a digital twin framework for structural engineering. The digital twin is conceptualised and mathematically idealised within the context of structural integrity, and includes the main attributes to behave as a functional digital twin, namely simulation, learning, and management. The manuscript makes special emphasis on the autonomous interactions between the physical and digital counterparts along with on the workflow modelling of the digital twin, which are both missing aspects in the majority of use cases found in the literature, specially within the civil and structural engineering domain. The proposed framework is demonstrated in a proof of concept using a laboratory scale test structure monitored using internet-of-things-based sensors and actuators. The results reveal that the virtual counterpart can respond in real-time with self-adaptability in liaison to the performance of the physical counterpart. Moreover, the tests show that the digital twin is able to provide automated decision making for structural integrity.

1. Introduction

Structures, regarded as physical infrastructural assets for public use, are continuously subjected to loads and environmental actions which cause cumulative deterioration along their lifespan. Nowadays, the increasing traffic demands on the communication networks along with the exposure to variations of environmental variables, make the condition assessment of structures a necessity to avoid an uncontrolled increase of system failures and unexpected downtimes, whilst keeping costs associated with maintenance and inspection under reasonably lower levels [1]. During decades, structural health monitoring (SHM) systems have produced large amounts of monitoring data to control the structural behaviour during operation [2]. With the rampant expand of applications of artificial intelligence (AI) to engineering problems, these data can be treated by intelligent algorithms capable of mining out information relevant to the actual and future health state of structures [3]. The latter, together with the maturity of the physics-based models of structural response reached during decades, have opened the door for new condition assessment paradigms which fuse the information from the data and the models under a more digital-focused trend.

The Digital Twin (DT) is one of the fast-evolving digital technologies that support the digital transformation of the structural engineering to enable optimal decision support for improving the management, reliability, and sustainability of structures [4,5]. The DT concept, originally conceived as a digital replica to a physical asset, was first coined in 2002 by Michael Grieves at the University of Michigan [6] although its first practical application to structures took place in 2012 in the aerospace sector [7,8].

Over the past decade, and coinciding with the *Industry 4.0* technological trend [9], the interest of the DT has rampantly growth across different industries with relevant use cases in sectors like the manufacturing [10,11], nuclear [12], and the naval [13], to cite but any. In the civil structural engineering domain, the DT technology is still in its infancy as compared to other industries, with earliest contributions found in the literature dating from 2018 [4]. However, nowadays this topic is gaining increasing attention, possibly as consequence of a *technology–push* by the irruption of the industrial Internet of Things (IoT) [14], and a *demand–pull*, due to modifications in the use of structures, climate change, and accumulated ageing [15]. During the last few years, the research about DT technology applied to civil or building engineering structures has significantly increased mostly within the areas of operation and maintenance (O&M). Table 1 provides a systematic overview of the relevant contributions of the DT in the structural engineering sector, whilst recent reviews and discussion about the broader application of the DT in civil engineering, are given in [16,17]. In Table 1, the revised papers are classified based on how the DT is conceptualised and used, whether the SHM data are used and integrated with the IoT, the use of AI algorithms, etc.

[*] Corresponding author at: Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
E-mail address: mchiachio@ugr.es (M. Chiachío).

https://doi.org/10.1016/j.autcon.2022.104333
Received 29 September 2021; Received in revised form 27 April 2022; Accepted 4 May 2022
Available online 19 May 2022

One of the main aspects that emerge when revising the referred literature provided in Table 1 is that there is no consolidated view on the concept of the DT within the structural engineering community, which coincides with the perspective of other authors about the same topic [16,31]. Besides, the majority of the studies only discuss how to build the digital representation of the virtual side of the DT for existing structural engineering assets. Some of them are mainly Building Information Modelling (BIM) models aiming at providing detailed geometrical implementation of the physical asset and enriched with SHM data [15,18,20,23]. Only few of them [25,26,31] provide methods and discussion about the updating and learning of the virtual models with respect to the data from the physical counterpart, or provide automated decision-making [26,27]. They do not incorporate a management model acting as connection system between the physical and virtual counterparts to autonomously coordinate the DT tasks like data gathering, model updating, or the internal *hardware-in-the-loop* to control sensors and actuators, etc. In general, the referred approaches found in the literature contribute with one or some of the functionalities required for a DT, however they do not consider all the essential elements of a functional DT together, namely: simulation, learning, and management [25], as depicted in Fig. 1.

This paper gives response to the aforementioned missing aspects from the literature by providing a technology integration framework of a functional DT within the structural O&M context. After a first step of conceptualization and mathematical idealization, the DT framework is characterized for simulation, learning, and management, which are all of them integrated within a web-based platform as modules. The model learning module is conceived here under the physics-based Bayesian approach for its appropriateness and rigour to quantify the uncertainty and for its robustness when dealing with *ill-conditioned* solutions in structural model identification [38,39]. On the other hand, the management of the entire DT is proposed as a dynamical event-based system model, which in this case has been formulated using the Petri net paradigm [40]. Petri nets (PNs) are typically regarded as powerful system-level workflow modelling tools due to their ability to account for resource availability, concurrency, and synchronisation within complex processes, which are common aspects that underline the majority of workflow models. From a mathematical perspective, PNs are bipartite directed graphs (digraphs) used to model and analyse event-based systems. The basic concepts relative to the theory of PNs are summarised in [41], whereas a tutorial for practical engineering applications of PNs can be found in [42]. In particular, High-Level Petri nets (HLPNs) [43] are used in this work as variant of the PNs to allow for higher complexity in the modelling of DT workflows.

Besides, this paper also gives technology integration of the DT framework, as a response to a generic claim by recent authors about the need for examples of practical implementations of DT technology integration [25,31,44]. To illustrate the proposed framework, a proof of concept is presented using a two-story laboratory scale test structure monitored using real-time IoT-based sensors and actuators. The proposed DT is applied within a structural integrity context to provide online decision support towards excessive deformation under the action of unknown loads. The results show that the proposed DT is able to virtually deal with the time-evolving behaviour of the physical twin under different load cases to aid and adopt automated decision-making actions.

The remainder of the paper is organised as follows. Section 2 presents the fundamentals and conceptualization of the DT within the structural engineering context, after which Section 3 provides mathematical idealization and computational aspects. In Section 4, the technology integration details are provided and the main attributes of the architecture of the proposed DT, are described. Section 5 illustrates and discusses the approach over an engineering proof of concept, and finally Section 6 gives concluding remarks.

2. The digital twin concept in structural engineering

After an analysis of the current development and research gap, the previous section indicated that there is a need for a consolidated view on the concept of DT for structural engineering, to which this section is devoted.

Following the definition by [44], a DT is a technology that enables a virtual representation of a physical system (and its associated environment and processes), which is continuously updated through the exchange of information between the physical and virtual world. Indeed, a structural DT interactively combines virtual representation models, referred to as the *virtual entity* (VE), with information from its physical counterpart, namely the *physical entity* (PE), within an on-line

Table 1
Bibliography synoptic table about DT in civil & building structural engineering applications.

Ref.	Year	Field	SHM	IoT	BIM	AI	MOD	INF	DIAG	PHM	UQ	WF	DS	ADM
[18]	2018	Bridge O&M	✓	–	✓	✓	✓	–	✓	–	–	–	–	–
[15]	2019	Bridge assessment	✓	–	✓	✓	✓	✓	–	–	✓	–	–	–
[19]	2019	Offshore Str. monitoring	✓	✓	✓	–	✓	–	–	–	–	–	–	–
[20]	2019	Bridge assessment	✓	–	✓	✓	–	–	–	–	–	✓	–	–
[21]	2019	Bridge O&M	–	–	✓	–	✓	–	✓	–	–	–	–	–
[22]	2019	Building monitoring	–	–	✓	–	✓	–	✓	✓	–	–	–	–
[23]	2019	Bridge monitoring	✓	–	✓	–	✓	–	✓	✓	–	–	–	–
[24]	2020	Structural assessment	✓	–	✓	✓	✓	–	–	–	–	–	–	–
[25]	2020	Structural assessment	✓	–	–	–	✓	✓	✓	–	✓	–	–	–
[26]	2020	Structural assessment	✓	–	–	✓	✓	✓	✓	✓	✓	✓	–	✓
[27]	2020	Eng. Systems degradation	✓	✓	–	✓	✓	–	51	✓	–	✓	✓	✓
[28]	2020	Building assessment	–	–	✓	–	✓	–	✓	✓	–	–	–	–
[29]	2020	Bridge assessment	✓	–	✓	–	✓	✓	✓	–	–	–	–	–
[30]	2020	Building O&M	✓	–	✓	✓	✓	✓	✓	–	✓	✓	✓	✓
[31]	2021	Structural assessment	–	–	–	✓	✓	–	✓	–	✓	–	–	–
[32]	2021	Structural assessment	✓	–	–	–	✓	–	✓	✓	✓	–	–	–
[33]	2021	Bridge assessment	✓	–	✓	✓	✓	–	✓	–	–	–	✓	–
[34]	2021	Offshore Str. assessment	✓	✓	–	✓	✓	–	✓	–	✓	–	✓	–
[35]	2021	Bridge assessment	✓	✓	✓	✓	✓	–	✓	–	–	–	✓	–
[36]	2021	Bridge assessment	✓	✓	✓	✓	✓	–	✓	–	–	–	✓	–
[37]	2022	Structural assessment	✓	–	–	✓	✓	✓	✓	–	✓	–	–	–
Author's work	2022	Structural O&M	✓	✓	–	✓	✓	✓	✓	–	✓	✓	✓	✓

Ref.: Reference; SHM: Use of SHM data; IoT: Use the IoT technology; BIM: Use of Building Information Modelling;
AI: Use of Artificial Intelligence; MOD: Use model as VE; INF: Makes Bayesian learning (Inference) between VE and PE;
DIAG: Application in diagnostics; PHM: Application in Prognostics; UQ: Provides uncertainty quantification; WF: Includes a workflow model;
DS:Provides decision support; AMD: Autonomous decision making.

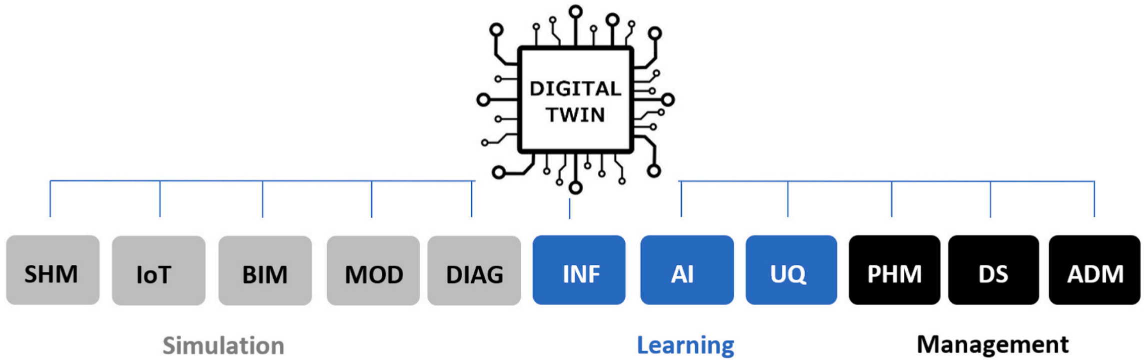


Fig. 1. Conceptualization of the Digital Twin by features and pillars. Refer to the caption of Table 1 for definition of the acronyms and abbreviations used in this figure. Observe that the SHM feature has been depicted into the *simulation* pillar, which applies when it is used within data-based approaches, although its role is also central within the *learning* pillar (in blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

connection system (CS) which includes sensors and actuators.

Within a structural engineering context, the PE comprises the structural components and sub-systems of the physical asset in question, which interacts with the surrounding environment through physical or chemical processes. The surrounding environment accounts for the external actions on the structure, like temperature, humidity, and loads, to name a few. On the other hand, the VE provides an idealization of the PE under a specific level of abstraction through mathematical models based on laws of physics, data, or both. The structural performance and its evolution over time is described by the models using states and parameters. The VE is adaptively updated based on the performance of the PE measured from the sensors throughout the whole structural life cycle. At the same time, it provides a tool to infer the structural health state of the structure, perform reliability and risk assessment, predict its remaining useful life (RUL), and autonomously obtain scenario planning and decision making actions [45].

Finally, the CS carries out the physical-to-virtual, yet virtual-to-physical, interaction through IoT-based sensors and actuators which collect and process data about the structural behaviour and the environmental conditions. Through a proper model updating method, the CS uses the data to update the VE from the measured PE states and environmental variables. Such updated information from the VE can be used in the opposite direction to autonomously provide levels of decisions and apply actions over the PE that might result in changes of the physical states through the actuators. Alternatively, *human-in-the-loop* actions, for example, visual inspection or structural repair actions, can be considered within the possible actions, although it implies off-line interaction [44]. These bi-directional interactions between the physical and virtual entities are governed by a management model, which also coordinates other lower-level actions such as data collection and processing (when needed) or the internal *hardware-in-the-loop* to control sensors and actuators.

As a final remark of the DT characterization, it is important to note that a structural DT cannot perfectly model the absolute real performance of the structure, either virtual or physical. This is because not all structural states and environmental variables can be directly measured with the available sensing methods, nor the whole structural performance can be idealised through all possible levels of abstraction within a modelling framework due to limitation of either knowledge, technology, or economics. Thus, a structural DT framework should be defined by using *contextual twins*, where the *context* [46] is given by a subset of structural parameters, health or damage states, and environmental variables of interest, whether certain or uncertain, that can be measured, modelled, or both.

3. Formulation of a structural digital twin framework

3.1. Mathematical idealization

From a mathematical perspective, let us consider that the structural performance can be represented by a n_s-dimensional state vector $\mathbf{s}(t) \in \mathscr{D} \subset \mathbb{R}^{n_s}$ at time t, and also that a vector of environmental variables $\mathbf{e}(t) \in \mathbb{R}^{n_e}$, is available. Let us also assume that the states $\mathbf{s}(t)$ can be measured during operation and that, at a certain time t, these states can be manifested through sensors' measurements $\mathbf{s}(t) = \mathbf{d}(\mathbf{e},\mathbf{w})$, where $\mathbf{d} : \mathbb{R}^{n_e} \times \mathbb{R}^{n_w} \rightarrow \mathscr{D}$ is a measurement equation and $\mathbf{w}$ is a n_w-dimensional measurement input vector, which accounts for the measurement error. Lastly, let us also consider that the virtual representation of the structural states, referred to as $\widehat{\mathbf{s}} \in \mathscr{M} \subset \mathbb{R}^{n_s}$, is given through a model $\mathbf{m}$ so that $\widehat{\mathbf{s}} = \mathbf{m}(\boldsymbol{\theta},\mathbf{u},\mathbf{e})$, $\mathbf{m} : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_s}$, which depends on a set of n_θ uncertain model parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ along with a set of model input parameters $\mathbf{u} \in \mathbb{R}^{n_u}$ and the environmental vector $\mathbf{e}$. Model $\mathbf{m}$ can be based on physical (i.e., structural) laws, can be built using data taken from the PE as a data-based model (e.g., through an Artificial Neural Network),[1] or can come from a combination of both sources of information.

Following this approach, the structural DT can be mathematically described under a specific context $\mathfrak{C}$, defined as $\mathfrak{C} \subset \mathscr{M} \times \mathscr{D}$, as follows:

$$\underbrace{\mathbf{m}(\boldsymbol{\theta},\mathbf{u},\mathbf{e})}_{\widehat{\mathbf{s}}} \overset{\text{CS}}{\underset{\mathfrak{C}}{\leftrightarrow}} \underbrace{\mathbf{d}(\mathbf{e},\mathbf{w})}_{\mathbf{s}} \tag{1}$$

where the double-arrow indicates that the correspondence between the virtual and physical representation of the structural states $\mathbf{s}, \widehat{\mathbf{s}} \in \mathfrak{C}$ is materialized through the CS.

3.2. Probabilistic Bayesian approach for model learning

The model representation of the VE, namely $\mathbf{m}(\boldsymbol{\theta},\mathbf{u},\mathbf{e})$, needs periodic updating based on the actual structural states $\mathbf{s}$ observed in the PE, through the SHM system. Since the models can be either data-based or physics-based, their updating will vary depending on such nature. As stated in Section 1,this work assumes that the structural response can be modelled using a physics-based approach (i.e., the stiffness method, or a Finite Element method), thus the model learning proposed here is based on a Bayesian approach.

The updating of a structural model using information gathered from sensors can be understood as the *inverse problem* [47], where there exist

[1] Note that in such a case, the SHM of the DT serves for monitoring the structural states through a measurement equation $\mathbf{d}$, but also for creating the model $\mathbf{m}$ through a training set of data.

certain parameters to be inferred [48]. If the Bayes' theorem is followed, the *prior* information $p(\boldsymbol{\theta}|\mathscr{M})$ about those uncertain parameters θ in the model class $\mathscr{M}$, is updated using the observed data $\mathscr{D}$, as follows:

$$p(\boldsymbol{\theta}|\mathscr{D},\mathscr{M}) = \frac{p(\mathscr{D}|\boldsymbol{\theta},\mathscr{M})p(\boldsymbol{\theta}|\mathscr{M})}{p(\mathscr{D}|\mathscr{M})} \tag{2}$$

where $p(\boldsymbol{\theta}|\mathscr{D},\mathscr{M})$ is the *posterior* PDF of the uncertain parameters, $p(\mathscr{D}|\boldsymbol{\theta},\mathscr{M})$ is the *likelihood function*, and $p(\mathscr{D}|\mathscr{M})$ is known as the *evidence*. In the proposed framework, the model class $\mathscr{M}$ is given by $\mathbf{m}(\boldsymbol{\theta},\mathbf{u},\mathbf{e})$ and the likelihood function is defined by the probability model chosen for the error $\mathbf{w}$.

At this standpoint it is important to note that within a DT context and a model-based approach, the posterior PDF not only provides information about the uncertain parameters but also helps us to evaluate the degree of agreement between the VE and PE. Indeed, the posterior $p(\boldsymbol{\theta}|\mathscr{D},\mathscr{M})$ indicates how plausible are the modelled states $\widehat{\mathbf{s}} = \mathbf{m}(\boldsymbol{\theta},\mathbf{u},\mathbf{e}) \in \mathscr{M}$ adopted by the VE to predict the observed outputs $\mathbf{s} \in \mathscr{D}$ in the PE, so that the higher the plausibility, the better the agreement.

3.3. Management modelling through the Petri net paradigm

This section briefly presents the method adopted to model the autonomous workflow of the DT using Petri nets. Petri nets (PNs) are bipartite directed graphs (digraph) used for modelling the dynamics of systems [40]. The graphical structure of a PN consists of two types of nodes, *transitions* and *places*, which are connected by *arcs*, which indicate the direction of the connection. A place, which is graphically symbolised using a circle, represents a particular state of the system or activity being modelled (e.g. the current damage state of a structural component or an inspection activity in progress). Places are temporarily visited by *tokens*, the abstract moving units of a PN. The distribution of tokens over the PN at a specific execution time is referred to as *marking*, which is expressed as a vector indicating the state of the PN. The transitions, represented by boxes, are responsible of the dynamic behaviour of the PN, and enable the system to move from one state to another [49]. Arcs are labeled with their corresponding *weights* (1 by default), and the *firing rule* dictate when tokens are created and destroyed to represent changes in the marking [50].

Mathematically, a PN can be defined as an ordered 5-tuple $\mathfrak{N}$ as follows [41]:

$$\mathfrak{N} \triangleq \langle \mathbf{P},\mathbf{T},\mathbf{E},\mathbf{M}_0,\mathbf{W} \rangle \tag{3}$$

where $\mathbf{P}$ and $\mathbf{T}$ denote the set of n_p places and n_t transitions of the PN, whose connections are expressed through the set of weighted edges $\mathbf{E} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$, with weights given in $\mathbf{W}$. Given an initial marking of the net, namely $\mathbf{M}_0$, the PN dynamics can be described through the change of the marking vector, which can be obtained by the following equation:

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{A}^T \mathbf{u}_k \tag{4}$$

where $\mathbf{u}_k$ is the *firing vector* at execution time k, a n_t-dimensional vector of Boolean values. $\mathbf{A} \in \mathbb{N}^{n_t \times n_p}$ is the *incidence matrix* of the graph, whose elements are the result of subtracting the *backward incidence matrix* from the *forward incidence matrix*, thus $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^- = a_{ij}^+ - a_{ij}^-$, where $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$. If transition t_i is activated at state k, then $u_{i,k} = \mathbb{I}_i$ according to the *firing rule*, where $\mathbb{I}_i$ takes the value of 1 if $M_k(j) \geqslant a_{ij}^- \ \forall p_j \in {}^\bullet\mathbf{P}_{t_i}$, or 0 otherwise. In this last equation, $M_k(j) \in \mathbb{N}$ is the marking at state k for place p_j, and ${}^\bullet\mathbf{P}_{t_i}$ denotes the set of places that belong to the preset of transition t_i, $i = 1, \ldots, n_t$.

More complex Petri net configurations can be found in the literature, like those with time synchronisation, where time delays are typically assigned to transitions. The resulting PNs are called Timed Petri Nets if the delays are deterministic, and Stochastic Petri Nets if the delays are specified by a probability model [41,51]. There also exist other PN variants that incorporate logical and mathematical predicates within the net elements (nodes and arcs), using a more flexible definition of the tokens and arcs types, along with the rules for transition firing. Those variants are referred to as high-level Petri nets (HLPN) [43] and allow for higher complexity in the modelling of systems dynamics as well as analysing logic flows in a more versatile manner.

A HLPN is used in this work by adding an extra *transition condition* to the firing rule, which uses algebraic predicates based on certain DT variables. These variables include model parameters, input parameters, environmental variables, and data. In mathematical terms, the resulting firing rule can be expressed as $u_{i,k} = \mathbb{I}_i \cdot \mathbb{I}_{\mathscr{C}_i}$, where:

$$\mathbb{I}_{\mathscr{C}_i} = \begin{cases} 1 & \text{if} \quad \mathscr{C}_i = \text{True} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In our framework, $\mathscr{C}_i = \mathscr{C}_i(\boldsymbol{\theta},\mathscr{D})$ is a boolean variable ($Bool \rightarrow \{\text{True},\text{False}\}$) and therefore, a transition t_i is fired only if it is enabled ($\mathbb{I}_i = 1$) and the transition condition $\mathscr{C}_i$ is true. The following rules summarize the algebra of the HLPN used in this work:

1. Transitions always consume tokens from all the input arcs at the same time, and produce the same number of tokens to all out-coming arcs;
2. Transition t_i is enabled if every input place p_j from its preset ${}^\bullet\mathbf{P}_{t_i}$ is marked with at least a_{ij}^- tokens;
3. An enabled transition t_i will fire if the transition condition $\mathscr{C}_i$ is true;
4. After firing, transition t_i removes a_{ij}^- tokens from p_j, and adds a_{ij}^+ tokens to each j-th output place of t_i.

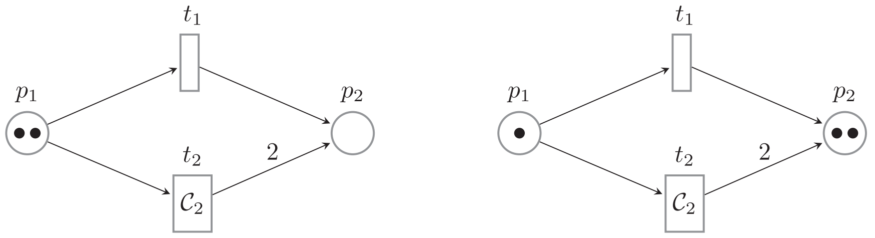
When representing HPNL graphically, the transition conditions $\mathscr{C}_i$ are shown within the transition box. If no transition condition exists, then $\mathbb{I}_{\mathscr{C}_i} = 1$.

Example 1. Let us consider the HLPN from Fig. 2 which consists of $n_p = 2$ places, and $n_t = 2$ transitions. In this example, the transition condition $\mathscr{C}_2$ is based on a sensor measurement $s \in \mathscr{D}$, such that $\mathscr{C}_2 = True$ if $\{s < 1\}$, expressed in dimensionless units. Let us now assume that the actual measurement of the sensor is $s = 0.5$ and the initial marking $\mathbf{M}_0 = (2,0)^T$, as depicted in Fig. 2. Then, by the HLPN firing rule described above, $\mathbb{I}_1 = 1$ since $M_0(1) \geqslant \underbrace{a_{11}^-}_{1}$, and $\mathbb{I}_{\mathscr{C}_1} = 1$. Regarding t_2, $\mathbb{I}_2 = 1$ since again, the marking of p_1 is higher than its input arc to t_2, and $\mathbb{I}_{\mathscr{C}_2} = 1$ because the transition condition $\mathscr{C}_2 = \{s < 1\}$ is true for the given transition variable $s = 0.5$. Thus, the marking can evolve from $\mathbf{M}_0$ to $\mathbf{M}_1$ as depicted in Fig. 2 (a) and (b).

4. Technology integration

As mentioned before, DT should not be seen as a single technology but as a set of devices, communication tools, and software working together, and most importantly, autonomously. Such autonomy implies machine to machine (M2M) communication without any direct human intervention, effectively achieved by the IoT. In this context, the physical world is perceived through a sensor network spatially distributed around the monitored structure, which means that a large amount of data is collected. At the same time, the devices can operate on the physical world through actuators when instructed. And that is one of the fundamental pillars of an IoT system, the seamless exchange of data between heterogeneous devices and their interoperability, achieved through a variety of protocols and standards. However, there also exist computational and energy constraints, hence the need for careful planning during the design stage. The best performance of the system is therefore achieved only when an efficient interoperability is assured.

The proposed DT of this work has been devised following these principles. Its overall architecture can be divided into three main blocks, as indicated in Fig. 3: the *smart devices* (sensors and actuators with micro-controllers and/or micro-processors), the *network connection*



(a) State of the HLPN at $k = 0$ (b) State of the HLPN at $k = 1$

Fig. 2. Sample illustration of a HLPN of two places and two transitions, one of which (t_2) is provided with a transition condition named as $\mathscr{C}_2$.

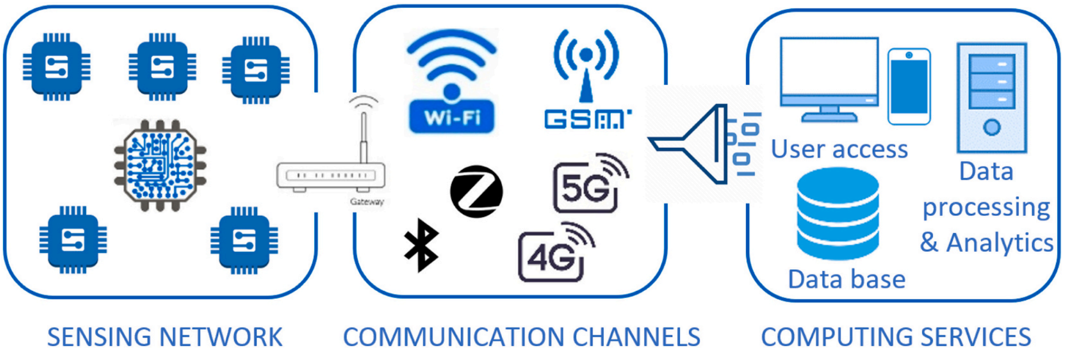


Fig. 3. Schematic view of the three main blocks comprising the technology integration of the proposed DT.

(including the physical media -wired or wireless, gateways, communication channels, protocols and standards for message transmission), and the *integration platform*, which comprises a *front-end* (the user interface through an *API web*) and a *back-end* (performing the data storage, data analytic, IoT applications and security, among others), acting as a binder of the whole system. The following subsection provides brief description of the main aspects of the technology integration through each of the referred blocks. To help the reader better understand such integration, a more detailed scheme is provided in Fig. 4.

4.1. Smart devices

The smart devices obtain data from the physical asset via sensors, which convert a physical state or stimulus into a signal. Actuators are also found in smart devices, turning signals into physical effects and triggering an action when a command is received. Connected to them are the micro-controllers and micro-processors, which process the digital signals for the sensors and actuators respectively, thus converting them into information. The choice, number, and location of the sensing appliances are critical to capture the essence of the physical asset and its environment [52,53], and conform the basis of well-known SHM

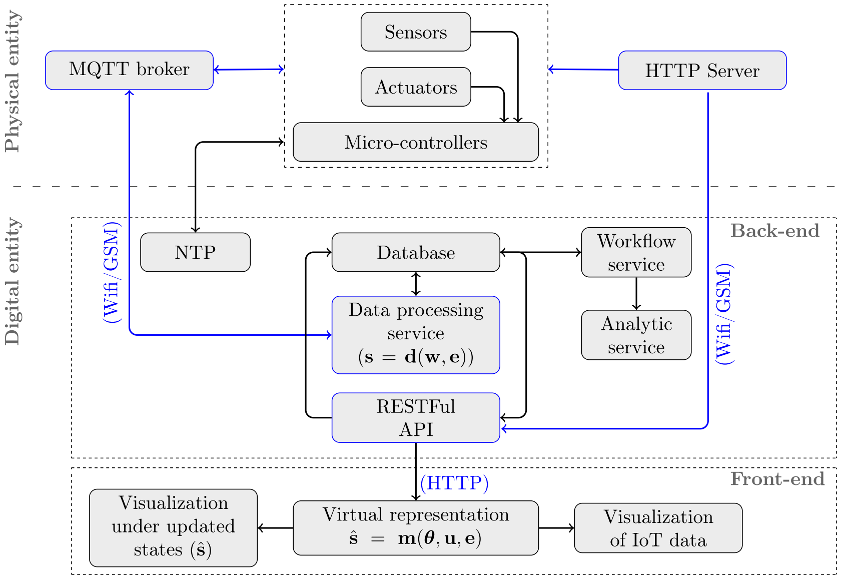


Fig. 4. Conceptual scheme of the proposed DT technology integration. The data communication features are drawn using blue colour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

techniques. In general, a wide range of sensors are available, from smart materials to fibre-optic sensors, and their use are case-specific. However the most suitable for wireless network communications within an IoT environment are the micro-electro-mechanical-systems (also referred to as MEMS), as they provide reliability, flexibility and a high digital capacity while they are low-cost, small, and have a low power consumption [14,54]. Moreover, the recent use of citizen sensors (i.e., through smartphones [55]) opens a new door within the DT context since it allows the integration of the sensors, the micro-controller, and communications within one device [56,57].

4.2. Communication and data transmission

In this work, the exchange of data between the devices interconnected within the DT network is enabled by IoT. Currently, there are a number of technologies which can be used for connectivity infrastructure (e.g. 6LowPAN, IPV6/RPL, IPv6), transport and communication (Wi-Fi, Bluetooth, Satellite -GSM, 3G, 4G, 5G, Radio Frequency, NFC, RFID), data transmission (MQTT, CoAP, HTTP), security (DTLS, TLS, MTLS, SSL) and device management (OMA-LwM2M, OMA-DM), to name a few use cases. Two main standards are used in the implementation of these communication technologies, namely the OSI model (Open Systems Interconnection, ISO/IEC 7498–1) and the TCP/IP model (Transmission Control Protocol/Internet Protocol, https://ietf.org/standards/), which describe the functions and processes deployed in each of them.

In the proposed DT approach two alternative wireless communication channels, namely wi-fi and and cellular network, are used to ensure a fluent communication of encrypted information (of type SSL) between the PE and VE. These channels use the MQTT and HTTP transmission protocols, which are both of type TCP/IP standard, resulting in low power consumption, embedded security, and scalability. MQTT is set as the primary communication channel due to its lightweight and fast message transmission capability [58]. Through MQTT, bidirectional connections can be established with a callback option for the reception of events and reconfiguration of the smart devices, if required. Also, it allows the data to be collected by a *MQTT broker Mosquitto*, which makes the data available to subscribers depending on the policy applied.

HTTP is used as the secondary channel, dedicated to the reception of events and to support the API web along with the real-time monitoring status. Through HTTP, each sensor is assigned a unique IP address to facilitate its identification and communication. Data are encapsulated into IPv6 packets and forwarded to an *Apache server*, operated in Linux, when a request is made. Here, the Apache server has been devised as an open-source server which implements the HTTP/1.1 protocol and works as a virtual site according to the RFC 2616 standard [59]. This protocol is not as efficient as MQTT for IoT but allows the transmission of heavier information, and allows the communication with the API web. Further details about the communication and data-transmission aspects are shown in Table 2.

By these communication protocols, data can be transmitted from the sensing nodes through the communication network to the digital twin API web, described next, where data are stored in a database for further processing. The data-interchange formats adopted here are lightweight, such as XML or JSON, and considered Open Data Formats by the Open Group IoT standard [60].

4.3. Web-based integration platform

The DT requires an inclusive platform to support the interaction of devices, networks, and software in an effective, reliable, and secure way. To this end, a representational state transfer (REST) software architecture [61], namely RESTful, is adopted to both manage the data coming from the IoT devices through HTTP, and provide real-time data access to the users [14,27]. By this means, the platform provides a bi-directional connection of the DT components, using a web based framework with decentralized control [62]. In our DT approach, a *Service Oriented Architecture* (SOA) is adopted for the API, thus the applications are conceived as *services* provided between components through their respective APIs and communication protocols, which can be accessed remotely, acted upon and updated independently.

The *back-end* of the platform is the edge of the IoT deployment, supporting the management of the interconnected devices and networks, the data storage and processing, along with the connection with the workflow model, as further described in Section 4.3.1. Through the back-end, the platform and the DT smart devices synchronise their clocks using an *network time protocol* (NTP) to allow real-time response and event-synchronisation. In this work, the platform has been hosted in a permanent server, although it can be alternatively hosted in a cloud-infrastructure should the application requires pervasive monitoring.

The *front-end* of the platform offers a user-friendly environment, helping end-users to easily manage all the available information. It also eases the end-user interaction with the DT nodes and devices, managing operations like real-time measurements requests or device reconfiguration. Indeed, since IoT-based monitoring implies supervising the operation of the sensors, including communications and data conveyed, if a fault is registered in any node of the physical entity, they can be remotely reconfigured or rebooted directly from the API.

4.3.1. Computing at the back-end

As stated in the previous section, the core computing modules of the DT are hosted in the back-end as services. There are three main services in the proposed DT, referred to as analytic, workflow, and data-processing. These services are hosted by a Python-based kernel, which communicates with the database, as depicted in Fig. 4.

The data processing service manages the DT data and allows these

Table 2
Summary of properties of the data transmission technologies and communication channels used in the proposed DT.

		MQTT	HTTP			Wi-fi	Cellular network
Message Transmitting Protocols	Specification	ISO/IEC 20922:2016	IEEE 802.15.4	Communication channel	Specification	Based on IEEE 802.11a/b/g	Based on GSM and GPRS
	Application	M2M & IoT devices	Wireless LAN, broadband Internet access		Application	Wireless LAN connectivity, broadband Internet	Mobile radio connectivity, voice and data services
	Standard	TCP/IP	TCP/IP		Band	Worldwide unlicensed 2.4 GHz	850/900/1800/ 1900 MHz (2G)
	Methodology	Data-centric	Document-centric		Topology	Line, ring, star, tree, or mesh	Cells
	Message size	Small	Large		Transmission distance	approx. 100 m	approx. 35 km
	Default port	1883 or 8883 (over SSL)	80 and 443 (over SSL)		Max. number of nodes	Unlimited	Unlimited

data from different smart devices to be combined into a single unified format and to be stored into the database. If required, the data processing service can also communicate with the connected smart devices via MQTT to send data requests and provide data configuration. The analytic service provides the structural model simulations of the VE along with the learning through probabilistic inference. The latter implies algorithms for producing the simulated structural response $\widehat{\mathbf{s}} = \mathbf{m}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{e})$, and for the Bayesian inference of model parameters $\boldsymbol{\theta}$, as stated in Sections 3.1 and 3.2, respectively.

The outputs of the analytic service are called upon request by the workflow service acting as CS of the DT. The workflow service provides the autonomous management of the DT through Petri net simulation, as explained in Section 3.3. Through the workflow service, the sequence of DT events, like performing Bayesian updating of the VE with respect to the PE, are envisaged as automated and adaptive actions which depend on the current state of the PE, whose information is requested from the database. In general, this service manages the interrelations between the rest of services and controls the required information or data to be exchanged between them.

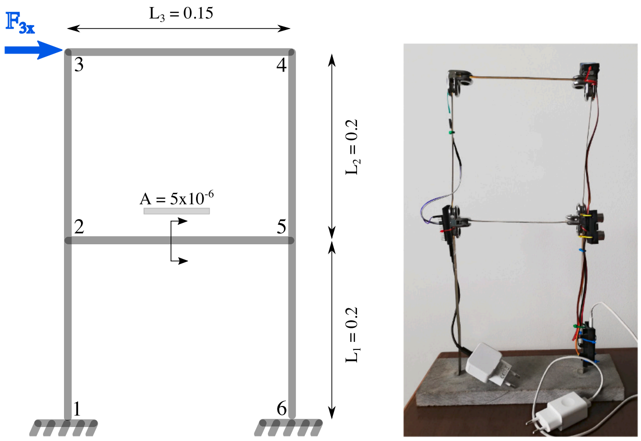
5. Proof of concept

In this section, the proposed DT context and settings are applied to a small scale laboratory test structure to support a sample integrity management operation based on structural elastic deformation.

5.1. Description

As depicted in Fig. 5, the test structure is a two-storey metallic frame made of an aluminium alloy which has been designed with fixed supports (representing the foundations) and rigid joints. The structure is subjected to an unknown horizontal point load (F_{3x}) applied to node 3, as shown in Fig. 5a.

Under these settings, a structural integrity scenario is idealised based on the value of the applied force, such that when F_{3x} is higher than a user-defined threshold force value F_ξ, an alarm is triggered. Besides, the proposed structural DT should be able to autonomously: (a) sense if a force F_{3x} is applied to the structure; (b) determine if the VE needs updating based on a pre-established displacement sensitivity; (c) perform Bayesian model updating of the VE to mach the state of the PE, which includes the inference of the unknown force F_{3x} with quantified uncertainty, among others; (d) decide if the integrity alarm should be triggered based on the inferred values of F_{3x}; (e) activate a pin led and a digital screen connected to a relay should the alarm is triggered and rearm the system after the actuator's notification; (f) keep the virtual and physical entities in continuous communication providing visualisation of the virtual one in compliance to the state of the PE; (g) provide instant information of any of the sensor's measurements.



(a) Frame model (b) Physical frame

Fig. 5. Model and picture of the two-storey test structure used in the proof of concept. Units expressed in SI.

The DT context is determined by both, the measured and modelled, elastic displacements of the structural joints. In mathematical terms $\mathbf{s} = \mathbf{s}(\mathbf{e}, \mathbf{w}) = [s_{ix}, s_{iy}, s_{iz}]$ and $\widehat{\mathbf{s}} = \widehat{\mathbf{s}}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{e}) = \left[\widehat{s}_{ix}, \widehat{s}_{iy}, \widehat{s}_{iz}\right]$, respectively, where $i = 1, \ldots, 6$, and x, y, z refer to horizontal, vertical, and rotation displacement. In this study, the measured displacements are taken for joint 5 using an ultrasound proximity sensor, thus $\mathbf{s}$ is specified here as $\mathbf{s} = [\mathbf{0}, -, -, -, \{s_{5x}, -, -\}, \mathbf{0}]$. The measurement error vector $\mathbf{w}$ is set to $\mathbf{w} = [10^{-4}]$, expressed in meter units, and relates to the sensitivity of the distance sensor. These values are used within the Bayesian inference module as standard deviation parameter for the likelihood function, considered as a Gaussian PDF. In the referred Bayesian inference module, the Metropolis-Hastings algorithm is implemented with a total of $N_s = 5 \cdot 10^4$ simulations, and a Gaussian PDF as proposal distribution with standard deviation given so that the resulting acceptance rate lies between the recommended interval [0.2,0.4] [63].

The modelled joint displacements $\widehat{\mathbf{s}}$ for the VE are obtained using a beam-element frame model as follows:

$$\widehat{\mathbf{s}}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{e}) = [\mathbf{K}]^{-1}[\mathbf{F}(\boldsymbol{\theta})]^T \tag{6}$$

where $\mathbf{F}(\boldsymbol{\theta}) = \left[\mathbf{F_1}, \mathbf{0}, \left\{\underbrace{F_{3x}}_{\boldsymbol{\theta}}, 0, 0\right\}, \mathbf{0}, \mathbf{0}, \mathbf{F_6}\right]$ is the vector of applied forces and moments to the joints of the frame structure. In this study, F_{3x} is the only force applied to the structure, which is a priori unknown, thus it is represented using the model parameter $\boldsymbol{\theta}$. The chosen prior PDF for $\boldsymbol{\theta}$ is given by a uniform PDF between the interval 0.1 and 5 in [N], namely $\mathscr{U}[0.1 \quad 5]$ [N], which represents our initial degree of belief about the possible values of the referred force. Note that $\mathbf{F}_1$ and $\mathbf{F}_6$ are the forces and moments applied to joints 1 and 6, respectively, however their values are not needed since these joints are fixed, hence their displacements and rotations are known and equal 0. In Eq. 6, the stiffness matrix $[\mathbf{K}] = [K_{ij}]$, $i, j = 1, \ldots, 6$, is a 18 × 18 elastic matrix of a plane framed structure with rigid joints. The values of K_{ij} depend on the geometrical and material input parameters $\mathbf{u} = (L_1, L_2, L_3, E, A, I)$, where L_1, L_2, L_3, and A are given in the Fig. 5. The term I is the cross-sectional inertia momentum given by $I = 10^{-9}$ [m^4], whereas $E = 9 \cdot 10^8$ [N/m^2] is the Young's modulus of the material. Finally, in this proof of concept, the influence of the environmental variables (e.g., temperature, humidity, etc.) over the structural response is not considered, thus $\mathbf{e} = \varnothing$.

5.2. Monitoring, communication, and hardware set-up

As presented in the previous section, the test structure is monitored during runtime using two smart devices: one sensor S_1 located at joint 5, and one relay A_1, whose positions are given in Fig. 5. These devices are supplied with electricity by a micro USB of 220 V AC to 5 V DC / 500 mA power. The main properties of the smart devices used in this example are given in Table 3.

Signals from the devices are integrated into the DT environment by

Table 3
Properties of the sensing and actuator devices used in the proof of concept.

ID	Device	Use range	Output	Electrical inputs	Notes
S_1	HC-SR04-P, Ultrasonic sensor	Proximity (<4 [m])	Distance [m]	3.3–5 [V] / 15 [mA]	40 [Hz] pulse-echo ultrasonic signal.
A_1	SRD-05VDC SL-C Relay	Switch (<10 [A])	250 [V] AC/ 30 [V] DC	5 [V] / 20 [mA]	Generates electro-mechanical switch.

connecting them to two different IoT boards: (a) an ESP8266 12-e board, which communicates to the integration platform via Wi-fi; and (b) an ESP32 TTGO T-CALL board providing cellular GSM connection and other alternative ways of cellular communication such as SMS messaging. Fig. 6 provides an overview indicating how the smart devices are connected to the IoT boards and attached to the physical structure. In this proof of concept, the main communication channel is the Wi-fi through the ESP8266 12-e board, which is based on a system on chip (SoC) integrating processor with a 32-bit architecture, working as a host for the API. In case of not availability of Wi-fi connection, then the GSM channel would provide the connection to the API through the ESP32 TTGO T-CALL e-board.

5.3. Workflow model

The workflow model for autonomous structural integrity decision making considered in this proof of concept is depicted in Fig. 7. The referred figure shows a HLPN comprising eight places (p_1 to p_8), seven transitions (t_1 to t_7), and two cold transitions (ϵ) for data arrival and system rearm. The places represent discrete-event states like 'data arrival', 'system updated', 'waiting mismatch evaluation', etc. To help the visual interpretation of the system states within the HLPN graph, coloured text labels are used in Fig. 7. The grey text labels provide explanatory information about place p_6 and one of the cold transitions. Changes in the state of the DT system are the result of a number of automated actions, which are triggered by the firing transitions t_1 to t_7. An overview and description of the actions associated to each transition are provided in Table 4. Note that the transitions t_1, t_5 and t_6 are based on transition conditions $\mathscr{C}_1, \mathscr{C}_5$ and $\mathscr{C}_6$, respectively, whose algebraic predicates are given in the third column of the aforementioned table. The activation of the referred transitions occurs when their variables fulfil the respective conditions $\mathscr{C}_i, i = 1, 5, 6$.

The dynamics of the HLPN can be described as follows. The system is assumed to start at $k = 0$ when new data arrive from the sensor S_1. At this stage, the VE is assumed to be updated with respect to the PE. This is represented by one token in both p_1 and p_4, thus $\mathbf{M}_0 = (1,0,0,1,0,0,0,0)^T$. Subsequently, t_4 is fired producing one token to place p_5 and removing it from p_4, thus the DT identifies that the structure is subjected to a new unknown force and a decision has to be made on whether identifying the force and updating the VE. This is carried out via transitions t_1 and t_5, which are based on transition conditions $\mathscr{C}_1$ and $\mathscr{C}_5$, respectively (recall Table 4). They are enabled depending on a mismatch evaluation given as follows:

$$\jmath = \frac{\| s_{5x} - \widetilde{s}_{5x} \|}{\widetilde{s}_{5x}} \tag{7}$$

where s_{5x} is the measured horizontal displacement in node 5 and $\widetilde{s}_{5x}$ represents its previously recorded value after the Bayesian inference, being $\widetilde{s}_{5x} = 0$ for $k = 0$. Thus, if the latest measured displacement differs by more than a 10% with respect to the previous recorded value (physically located in the database, as stated in Fig. 7), then t_1 is fired

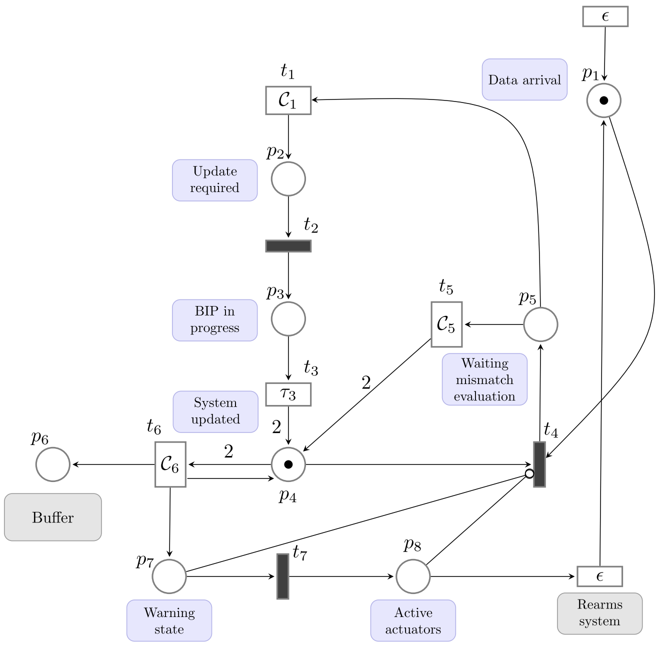


Fig. 7. HLPN used as workflow model of the proof of concept of Section 5. The dark small rectangles indicate symbolic transitions, whereas the blue and grey text labels provide some explanatory information about key places. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Description of the transitions of the HLPN model shown in Fig. 7. In the third row, the symbol τ_3 represents the time in seconds required for the VE to process the Bayesian inference. The term $\jmath$ refers to the mismatch value, and is given in Eq. 7.

Transition	Type	Rule	Description
t_1	Conditional	$\mathscr{C}_1 : \{\jmath \geqslant 0.1\}$	Evaluates mismatch
t_2	Symbolic	–	Call the BIP module
t_3	Timed	Enabled after τ_3	Execute the BIP
t_4	Symbolic	–	Initiates VE–PE mismatch
t_5	Conditional	$\mathscr{C}_5 : \{\jmath < 0.1\}$	Evaluates mismatch
t_6	Conditional	$\mathscr{C}_6 : \{\text{mean}(F_{3x}) \geqslant F_\xi\}$	Checks value of inferred force
t_7	Symbolic	–	Activates actuator

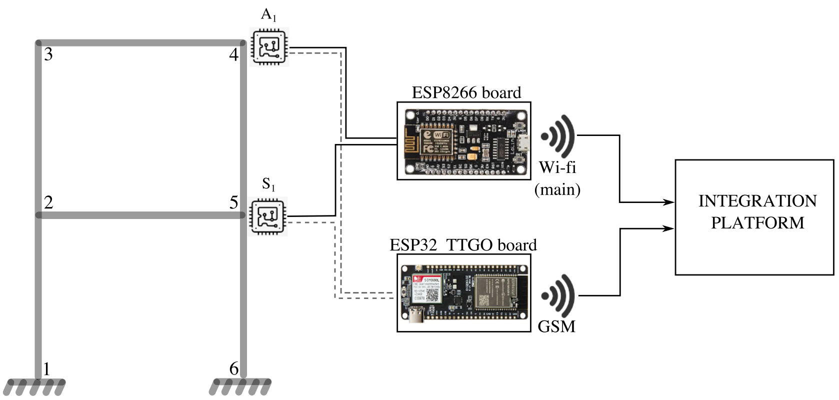


Fig. 6. Connection scheme of the smart devices used in the proof of concept.

and the DT performs a structural Bayesian updating. This is represented through the workflow sequence $\{p_2, t_2, p_3, t_3\}$, which will finally produce one token to p_4 and the system turns back to 'updated' state. Otherwise, t_5 will directly produce one token to p_4, meaning that the DT does not require a Bayesian updating of the VE with respect to the PE, and thus the DT keeps its previous 'updated' state.

Note from the HLPN graph depicted in Fig. 7 that while the DT remains in the 'updated state', which implies that place p_4 is marked, an interrogation is performed about the values of the inferred force to check whether its mean value exceeds the threshold F_ξ, whereupon transition t_6 is fired. This transition activates a sequence of warning states and actions given by nodes $\{p_7, t_7, p_8\}$, which autonomously reveal when the structure is subjected to a force that can compromise its integrity, then a visual alarm is triggered by the LED and screen actuators (through firing t_7), and the system turns to 'warning state'. At such standpoint, the system is rearmed waiting for new data arrival, represented through the cold transition (ε), whereby the warning state is dismissed until new evaluation. Finally note also that when the warning sequence is activated, a token is collected in p_6 acting as a buffer of information, which can be used for diagnosis purposes as it provides us with the amount of times that the structure has been subjected to force values above the integrity threshold F_ξ.

5.4. Results and discussion

Once the smart devices are active and connected to their corresponding e-boards, a number of structural integrity scenarios are explored by consecutively applying a series of 15 load cases to node 3 of the test structure. In every case, the overall DT behaviour is commanded through the HLPN shown in Fig. 7, whose dynamics is evaluated through the state evolution equation (recall Eq. 4) in confluence with the execution rules summarised in Section 3.3 to obtain the sequence of system states described by the marking $\mathbf{M}_k$, $k > 0$. The results for the DT behaviour in terms of the applied and inferred loads, are depicted in Fig. 8. In this figure, the actual applied forces to the test structure (drawn using grey box-dotted line) have been measured through a thin-film force sensor external to the DT, i.e., its data are not processed within the DT environment, thus its measured values are only revealed visually to validate the inferred values by the DT. Moreover, the referred figure provides indication of the cases when the main changes of HLPN states happen.

Note that in every load case requiring updating of the DT, the Bayesian module is called to produce a posterior PDF of the inferred values of the unknown force $\boldsymbol{\theta}$, whereby the rest of the model response (displacements and rotations of nodes) can be reproduced. To serve as example of the DT front-end visualisation capabilities, Fig. 9 provides a plot of the posterior versus prior PDF of model parameter $\boldsymbol{\theta}$ for load case # 13, along with the corresponding graphical visualisation of the VE. Note that the front-end also shows indication of some key state variables from the PE under the referred load case.

Extended information about the response of the DT to the test loads is provided in Table 5. In addition, a summary of the results for the behaviour of the HLPN model is provided for test cases 1, 2, and 10 in Tables 6, 7 and 8, respectively. In these tables, the fourth column indicates the sequence of main events, like activation of the warning state and/or firing of conditional transitions. The right-most column provides description of the overall DT behaviour in liaison to the the HLPN states.

Observe for load case 1 (refer to Table 6) that the transition t_1 is fired at $k = 1$ since it is enabled with a token in its preset place p_5 and the condition $\mathscr{C}_1$ is true. This produces one token to p_2 whereupon the Bayesian updating of the DT states is carried out, after which the system recovers back to the updated state with two tokens in p_4. When updated, the resulting inferred force results higher than the threshold force $F_\xi = 1.5$ [N], then transition t_6 is fired and the warning sequence $\{p_7, t_7, p_8\}$, along with their corresponding actuators, are activated, as can be observed in Table 6 after time step $k = 4$.

On the other hand, note that in load case 2, the measured displacement by the ultrasound sensor gives a value of 36 [mm], which differs by less than a 10% with respect to the previous one (i.e, 39 [mm]). Thus, the HLPN identifies that an updating of the VE with respect to the PE is not required, then the system is kept under the system updated state by activating t_5 to recover two tokens back to p_4. Notwithstanding, the HLPN detects that the force applied to the structure is still higher than the threshold force $F_\xi = 1.5$ [N], then transition t_6 is fired again whereupon the warning events are activated and the system switches

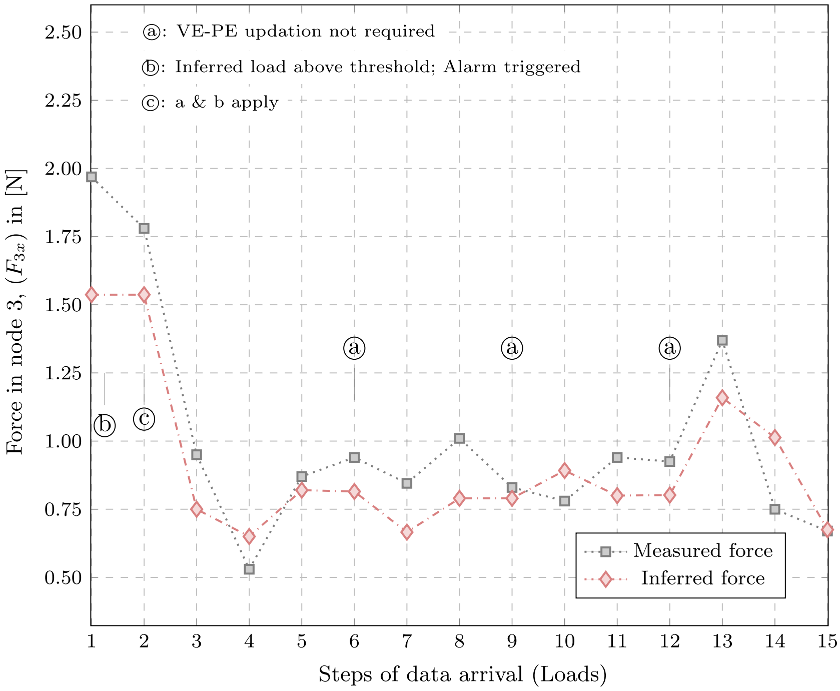
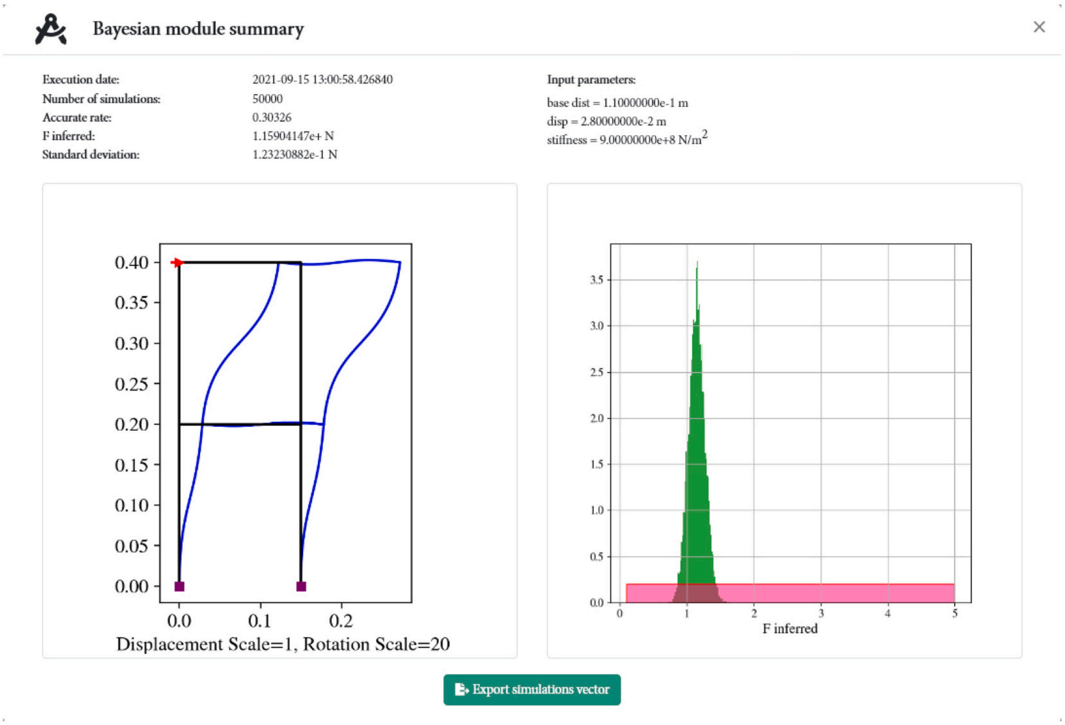


Fig. 8. Sequence of applied (in grey) and inferred (in red) loads, along with indication of main DT actions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Bayesian module summary

Execution date:	2021-09-15 13:00:58.426840
Number of simulations:	50000
Accurate rate:	0.30326
F inferred:	1.15904147e+ N
Standard deviation:	1.23230882e-1 N

Input parameters:

base dist = 1.10000000e-1 m
disp = 2.80000000e-2 m
stiffness = 9.00000000e+8 N/m^2

Fig. 9. Plot of the front-end Bayesian module for load case 13 (refer to Table 5), with indication of the posterior PDF of the inferred force (right side) along with a graphical representation of the updated VE (left side). In the right panel, the pink rectangle represents the uniform prior PDF of the applied force, whereas the posterior PDF is given in green colour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Summary of measured and inferred DT variables for the sequence of 15 load cases applied to the test structure. In the third and forth rows, the symbol θ represents the unknown parameter, which coincides with the force F_{3x}.

Load case	Measured force [N]	θ_{mean} [N]	θ_{std} [N]	s_{5x} [mm]	$J \geqslant$ 10%	Activates warning
1	1.92	1.60	0.13	39	Yes	Yes
2	1.77	1.60	0.13	36	No	Yes
3	0.95	0.75	0.12	20	Yes	No
4	0.53	0.65	0.11	15	Yes	No
5	0.87	0.82	0.12	21	Yes	No
6	0.94	0.82	0.12	19	No	No
7	0.85	0.67	0.12	16	Yes	No
8	1.01	0.77	0.12	18	Yes	No
9	0.83	0.77	0.12	19	No	No
10	0.78	0.88	0.12	23	Yes	No
11	0.94	0.80	0.12	20	Yes	No
12	0.92	0.80	0.12	22	No	No
13	1.37	1.16	0.12	28	Yes	No
14	0.75	1.01	0.12	24	Yes	No
15	0.67	0.66	0.13	16	Yes	No

from normal operation to warning state mode where the actuator A_1 triggers visual alarms, as explained above. This is reflected in Table 7 after time step $k = 2$.

Finally, the load case 10 represents the situation in which the DT detects that an updating of the VE with respect to the PE is required. Once updated, the resulting inferred force does not reach the threshold value, thus the workflow model stops and awaits for new data.

In general, these results corroborate that the proposed DT framework is able to autonomously respond to the arrival of data through adaptation whereby the VE can be updated based on the performance of the PE measured from the sensors. Correspondingly, the results also show that PE can be influenced by the information from the VE thorough actuators. Moreover, the proposed proof of concept has also shown that these interactions can be effectively executed through a HLPN acting as workflow

Table 6
Summary of events and actions carried out by the PN workflow model under load case 1.

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main events	Description
$k = 0$	$(1\,0\,0\,1\,0\,0\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	New data arrived	PN starts
$k = 1$	$(0\,0\,0\,0\,1\,0\,0\,0)^T$	$(1\,0\,0\,0\,0\,0\,0)$	$\mathscr{C}_1$ {→True}	Checks to update VE & PE {→True}
$k = 2$	$(0\,1\,0\,0\,0\,0\,0\,0)^T$	$(0\,1\,0\,0\,0\,0\,0)$	Update required	Call the BIP module
$k = 3$	$(0\,0\,1\,0\,0\,0\,0\,0)^T$	$(0\,0\,1\,0\,0\,0\,0)$	BIP under execution	VE updating according to PE
$k = 4$	$(0\,0\,0\,2\,0\,0\,0\,0)^T$	$(0\,0\,0\,0\,0\,1\,0)$	$\mathscr{C}_6$ {→True}	System updated; $F_{3x} \geqslant F_{\xi}$→ {True}
$k = 5$	$(0\,0\,0\,1\,0\,1\,1\,0)^T$	$(0\,0\,0\,0\,0\,0\,1)$	Action required	Activates actuator A_1
$k = 6$	$(0\,0\,0\,1\,0\,1\,0\,1)^T$	$(0\,0\,0\,0\,0\,0\,0)$	Warning state	System rearm; awaiting new data

Table 7
Summary of events and actions carried out by the PN workflow model under load case 2.

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main events	Description
$k = 0$	$(1\,0\,0\,1\,0\,1\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	New data arrived	PN starts
$k = 1$	$(0\,0\,0\,0\,1\,1\,0\,0)^T$	$(0\,0\,0\,0\,1\,0\,0)$	$\mathscr{C}_5$ {→True}	Checks to update VE & PE {→False}
$k = 2$	$(0\,0\,0\,2\,0\,1\,0\,0)^T$	$(0\,0\,0\,0\,0\,1\,0)$	$\mathscr{C}_6$ {→True}	System previously updated; $F_{3x} \geqslant F_{\xi}$→ {True}
$k = 3$	$(0\,0\,0\,1\,0\,2\,1\,0)^T$	$(0\,0\,0\,0\,0\,0\,1)$	Action required	Activates actuator A_1
$k = 4$	$(0\,0\,0\,1\,0\,2\,0\,1)^T$	$(0\,0\,0\,0\,0\,0\,0)$	Warning state	System rearm; awaiting new data

Table 8
Summary of events and actions carried out by the PN workflow model under load case 10.

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main PN events	Description
$k = 0$	$(1\,0\,0\,1\,0\,2\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	Data arrival	PN starts
$k = 1$	$(0\,0\,0\,0\,1\,2\,0\,0)^T$	$(1\,0\,0\,0\,0\,0\,0)$	$\mathscr{C}_1$ {→True}	Checks to update VE & PE {→True}
$k = 2$	$(0\,1\,0\,0\,0\,2\,0\,0)^T$	$(0\,1\,0\,0\,0\,0\,0)$	Update required	Call the BIP module
$k = 3$	$(0\,0\,1\,0\,0\,2\,0\,0)^T$	$(0\,0\,1\,0\,0\,0\,0)$	BIP under execution	VE updating according to PE
$k = 4$	$(0\,0\,0\,2\,0\,2\,0\,0)^T$	$(0\,0\,0\,0\,0\,0\,0)$	–	System updated; waiting new data

model of the DT by providing the event-based basis for system-level synchronisation between the VE and PE.

6. Concluding remarks

This article presented a DT framework for structures which was conceived from mathematical idealization and applied to the level of hardware/software technology integration. The framework was defined under a structural integrity context with main objective being the monitoring and control of physical and environmental variables affecting the structural performance. Within this context, the virtual and physical counterparts interacted within a real-time IoT-based SHM environment. PNs and more specifically, HLPNs have been devised as effective tools as workflow models to ensure the autonomous management of the DT, and specially, the interaction between the physical and virtual counterparts. A proof of concept has been provided over a laboratory test structure to conceptualise the framework as well as to show some of the challenges faced in a real-world application. The following are some concluding remarks:

(i) The proposed DT framework has demonstrated that the physical-to-virtual, yet virtual-to-physical interactions are possible within a context of structural integrity;

(ii) A probabilistic Bayesian approach has been proposed for model updating for its maturity within the context of SHM, structural control, and structural integrity [38,39]. However, other available learning methods (e.g., Machine Learning methods) can be integrated as complement or as replacer to the Bayesian one, with no loss of generality;

(iii) The framework has been demonstrated in a proof of concept using a laboratory test structure to avoid very high forces to produce the integrity scenarios (i.e., node displacements). The latter might have required unnecessarily heavy experimental setup. However, it is important to remark that the use of a small-scale test implies unobserved structural aspects that can only be revealed in a large-scale model, thus being one of the limitations of this work. Notwithstanding, the proof of concept has enabled the conceptualisation, formulation, and technology integration of the proposed DT, as a previous step to be applied to larger (even real-world) structures;

(iv) The results have demonstrated that the proposed DT can assist the decision making for failure prevention since the VE can be used to perform reliability and risk assessment under damage, and also because alarms can be automatically triggered under failure scenarios;

(v) More research is needed to explore methods for optimal computational allocation (like edge or fog computing [64]) within the software/hardware integration of the DT to allow an efficient application to cases requiring complex structural models, pervasive monitoring, and citizen-centered sensors [55]. Also, desirable work would also imply demonstration of the proposed DT within a full-scale structural application.

To conclude, the authors remark that another limitation of the proposed work is the need of adaptation of the analytic service to the case under study, or in other words, its lack of easy extrapolation to different case studies, which is just a limitation of the DT technology itself. In addition to the latter, we can add that the IoT technology is under continuous development, hence the definition of a long-term monitoring scheme of a DT for structural integrity, should be periodically updated and supervised to accommodate the changes in the technology, like communication and security protocols, to cite but any.

Notwithstanding, it is important to note that the IoT-based SHM is only one, but a major, element within a bigger technology integration of the DT, as depicted in Fig. 4. Thus, a framework for formulation and integration like the one proposed here contributes to support the required future DT development even in the scenario of technology change.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This paper is part of the ENHAnCE ITN project (https://www.h2020-enhanceitn.eu/) funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957; and also of the HYPERION project (https://www.hyperion-project.eu/) funded from European Union's Horizon 2020 research and innovation programme under grant agreement No. 821054. Funding for open access charge: Universidad de Granada / CBUA. The authors would also like to acknowledge Mr. Ramón Jesús Suárez Pérez from University of Granada for his support on the IoT experimental work. Funding for open access charge: Universidad de Granada / CBUA.

References

[1] E. Zio, Reliability engineering: old problems and new challenges, Reliab. Eng. Syst. Saf. 94 (2) (2021) 125–141, https://doi.org/10.1016/j.ress.2008.06.002.

[2] C.R. Farrar, N.A. Lieven, Damage prognosis: the future of structural health monitoring, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 365 (1851) 623–632, https://doi.org/10.1098/rsta.2006.1927.

[3] M. Flah, I. Nunez, W.B. Chaabene, M.L. Nehdi, Machine learning algorithms in civil structural health monitoring: A systematic review, Arch. Comput. Methods Eng. 28 (4) (2021) 2621–2643, https://doi.org/10.1007/s11831-020-09471-9.

[4] U.T. Tygesen, M.S. Jepsen, J. Vestermark, N. Dollerup, A. Pedersen, The true digital twin concept for fatigue re-assessment of marine structures, in: ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers Digital Collection, 2018, https://doi.org/10.1115/OMAE2018-77915.

[5] P.E. Love, J. Matthews, The 'how' of benefits management for digital technology: from engineering to asset management, Autom. Constr. 107 (2019), 102930, https://doi.org/10.1016/j.autcon.2019.102930.

[6] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, in: Transdisciplinary Perspectives on Complex Systems, Springer, 2017, pp. 85–113, https://doi.org/10.1007/978-3-319-38756-7_4.

[7] E.J. Tuegel, A.R. Ingraffea, T.G. Eason, S.M. Spottswood, Reengineering aircraft structural life prediction using a digital twin, Int. J. Aerospace Eng. (2011) 11–14, https://doi.org/10.1155/2011/154798.

[8] E. Glaessgen, D. Stargel, The digital twin paradigm for future NASA and U.S. air force vehicles, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA, 2012. https://arc.aiaa.org/doi/abs/10.2514/6.2012-1818.

[9] E. Negri, L. Fumagalli, M. Macchi, A review of the roles of digital twin in cps-based production systems, Procedia Manufact. 11 (2021) 939–948, https://doi.org/10.1016/j.promfg.2017.07.198.

[10] S. Haag, R. Anderl, Digital twin–proof of concept, Manufact. Lett. 15 (2021) 64–66, https://doi.org/10.1016/j.mfglet.2018.02.006.

[11] Q. Liu, J. Leng, D. Yan, D. Zhang, L. Wei, A. Yu, R. Zhao, H. Zhang, X. Chen, Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system, J. Manuf. Syst. 58 (2021) 52–64, https://doi.org/10.1016/j.jmsy.2020.04.012.

[12] D. Iglesias, P. Bunting, S. Esquembri, J. Hollocombe, S. Silburn, L. Vitton-Mea, I. Balboa, A. Huber, G. Matthews, V. Riccardo, et al., Digital twin applications for

the jet divertor, Fusion Eng. Design 125 (2021) 71–76, https://doi.org/10.1016/j.fusengdes.2017.10.012.

[13] Í.A. Fonseca, H.M. Gaspar, Challenges when creating a cohesive digital twin ship: a data modelling perspective, Ship Technol. Res. 68 (2) (2021) 70–83, https://doi.org/10.1080/09377255.2020.1815140.

[14] C.A. Tokognon, B. Gao, G.Y. Tian, Y. Yan, Structural health monitoring framework based on internet of things: A survey, IEEE Internet Things J. 4 (3) (2021) 619–635, https://doi.org/10.1109/JIOT.2017.2664072.

[15] C. Ye, L. Butler, C. Bartek, M. Iangurazov, Q. Lu, A. Gregory, M. Girolami, C. Middleton, A digital twin of bridges for structural health monitoring, in: 12th International Workshop on Structural Health Monitoring 2019, Stanford University, 2019, https://doi.org/10.12783/shm2019/32287.

[16] F. Jiang, L. Ma, T. Broyd, K. Chen, Digital twin and its implementations in the civil engineering sector, Autom. Constr. 130 (2021), 103838, https://doi.org/10.1016/j.autcon.2021.103838.

[17] M.F. Bado, D. Tonelli, F. Poli, D. Zonta, J.R. Casas, Digital twin for civil engineering systems: an exploratory review for distributed sensing updating, Sensors 22 (9) (2022) 3168, https://doi.org/10.3390/s22093168.

[18] N. Dang, H. Kang, S. Lon, C. Shim, 3d digital twin models for bridge maintenance, in: Proceedings of 10th International Conference on Short and Medium Span Bridges, Quebec city, Quebec, Canada, 2018.

[19] C.U. Grosse, Monitoring and inspection techniques supporting a digital twin concept in civil engineering, in: Proc 5th International Conference on Sustainable Construction Materials and Technologies (SCMT5): In honour of Professor Christian Grosse, 2019.

[20] R. Lu, I. Brilakis, Digital twinning of existing reinforced concrete bridges from labelled point clusters, Autom. Constr. 105 (2019), 102837, https://doi.org/10.1016/j.autcon.2019.102837.

[21] C.-S. Shim, N.-S. Dang, S. Lon, C.-H. Jeon, Development of a bridge maintenance system for prestressed concrete bridges using 3d digital twin model, Struct. Infrastruct. Eng. 15 (10) (2021) 1319–1332, https://doi.org/10.1080/15732479.2019.1620789.

[22] F. Tahmasebinia, D. Fogerty, L.O. Wu, Z. Li, S. Sepasgozar, K. Zhang, S. Sepasgozar, F. Alonso Marroquin, Numerical analysis of the creep and shrinkage experienced in the Sydney opera house and the rise of digital twin as future monitoring technology, Buildings 9 (6) (2019) 1–18, https://doi.org/10.3390/buildings9060137.

[23] C. Boddupalli, A. Sadhu, E. Rezazadeh Azar, S. Pattyson, Improved visualization of infrastructure monitoring data using building information modeling, Struct. Infrastruct. Eng. 15 (9) (2021) 1247–1263, https://doi.org/10.1080/15732479.2019.1602150.

[24] Z. Liu, W. Bai, X. Du, A. Zhang, Z. Xing, A. Jiang, Digital twin-based safety evaluation of prestressed steel structure, Adv. Civil Eng. (2020), https://doi.org/10.1155/2020/8888876.

[25] D. Wagg, K. Worden, R. Barthorpe, P. Gardner, Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications, ASCE-ASME J. Risk Uncertainty Eng. Syst. Part B: Mech. Eng. 6 (3) (2021), https://doi.org/10.1115/1.4046739.

[26] P. Gardner, M. Dal Borgo, V. Ruffini, A.J. Hughes, Y. Zhu, D.J. Wagg, Towards the development of an operational digital twin, Vibration 3 (3) (2021) 235–265, https://doi.org/10.3390/vibration3030018.

[27] D. D'Amico, J. Ekoyuncu, S. Addepalli, C. Smith, E. Keedwell, J. Sibson, S. Penver, Conceptual framework of a digital twin to evaluate the degradation status of complex engineering systems, in: Procedia CIRP 86, 2021, pp. 61–67, https://doi.org/10.1016/j.procir.2020.01.043.

[28] G. Angjeliu, D. Coronelli, G. Cardani, Development of the simulation model for digital twin applications in historical masonry buildings: the integration between numerical and experimental reality, Comp. Struct. 238 (2020) 106282, https://doi.org/10.1016/j.compstruc.2020.106282.

[29] S. Ye, X. Lai, I. Bartoli, A.E. Aktan, Technology for condition and performance evaluation of highway bridges, J. Civ. Struct. Heal. Monit. 10 (4) (2021) 573–594, https://doi.org/10.1007/s13349-020-00403-6.

[30] Q. Lu, X. Xie, A.K. Parlikad, J.M. Schooling, Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance, Automat. Construct. 118 (2020) 103277, https://doi.org/10.1016/j.autcon.2020.103277.

[31] T. Ritto, F. Rochinha, Digital twin, physics-based model, and machine learning applied to damage detection in structures, Mech. Syst. Signal Process. 155 (2021), 107614, https://doi.org/10.1016/j.ymssp.2021.107614.

[32] K. Lin, Y.-L. Xu, X. Lu, Z. Guan, J. Li, Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes, Automat. Construct. 123 (2021) 103547, https://doi.org/10.1016/j.autcon.2020.103547.

[33] S. Kaewunruen, J. Sresakoolchai, W. Ma, O. Phil-Ebosie, Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions, Sustainability 13 (4) (2021), 2051, https://doi.org/10.3390/su13042051.

[34] M. Wang, C. Wang, A. Hnydiuk-Stefan, S. Feng, I. Atilla, Z. Li, Recent progress on reliability analysis of offshore wind turbine support structures considering digital twin solutions, Ocean Eng. 232 (2021) 109168, https://doi.org/10.1016/j.oceaneng.2021.109168.

[35] D. Dan, Y. Ying, L. Ge, Digital twin system of bridges group based on machine vision fusion monitoring of bridge traffic load, IEEE Trans. Intell. Transp. Syst. (2021), https://doi.org/10.1109/TITS.2021.3130025.

[36] Z. Liu, A. Jiang, A. Zhang, Z. Xing, X. Du, Intelligent prediction method for operation and maintenance safety of prestressed steel structure based on digital twin technology, Adv. Civil Eng. (2021), https://doi.org/10.1155/2021/6640198.

[37] M. Wang, S. Feng, A. Incecik, G. Królczyk, Z. Li, Structural fatigue life prediction considering model uncertainties through a novel digital twin-driven approach, Comput. Methods Appl. Mech. Eng. 391 (2022), 114512, https://doi.org/10.1016/j.cma.2021.114512.

[38] J.L. Beck, K.-V. Yuen, Model selection using response measurements: Bayesian probabilistic approach, J. Eng. Mech. 130 (2) (2021) 192–203, https://doi.org/10.1061/(ASCE)0733-9399(2004)130:2(192.

[39] J.L. Beck, Bayesian system identification based on probability logic, Struct. Control. Health Monit. 17 (7) (2021) 825–847, https://doi.org/10.1002/stc.424.

[40] C.A. Petri, Kommunikation mit automaten, Ph.D. thesis, Institut für Instrumentelle Mathematik an der Universität Bonn, 1962.

[41] T. Murata, Petri nets: Properties, analysis and applications, Proc. IEEE 77 (4) (2021) 541–580, https://doi.org/10.1109/5.24143.

[42] R. Zurawski, M. Zhou, Petri nets and industrial applications: A tutorial, IEEE Trans. Ind. Electron. 41 (6) (2021) 567–583, https://doi.org/10.1109/41.334574.

[43] K. Jensen, G. Rozenberg, High-Level Petri Nets: Theory and Application, Springer Science & Business Media, 2012. ISBN: 978-3-642-84524-6.

[44] E. VanDerHorn, S. Mahadevan, Digital twin: generalization, characterization and implementation, Decis. Support. Syst. 145 (2021), 113524, https://doi.org/10.1016/j.dss.2021.113524.

[45] C. Kan, C. Anumba, Digital Twins as the Next Phase of Cyber-Physical Systems in Construction, in: Computing in Civil Engineering 2019: Data, Sensing, and Analytics, American Society of Civil Engineers Reston, VA, 2019, pp. 256–264, https://doi.org/10.1061/9780784482438.033.

[46] K. Worden, E. Cross, P. Gardner, R. Barthorpe, D. Wagg, On digital twins, mirrors and virtualisations, in: Model Validation and Uncertainty Quantification Volume 3, Springer, 2020, pp. 285–295, https://doi.org/10.1115/1.4046740.

[47] A. Tarantola, Inverse problem theory and methods for model parameter estimation. SIAM, 2005. ISBN: 978-0-898-71572-9.

[48] J. Chiachio-Ruano, M. Chiachio-Ruano, S. Sankararaman, Bayesian Inverse Problems: Fundamentals and Engineering Applications, CRC Press, 2021. ISBN: 978–1–138-03585-0.

[49] J. Andrews, D. Prescott, F. De Rozières, A stochastic model for railway track asset management, Reliab. Eng. Syst. Saf. 130 (2021) 76–84, https://doi.org/10.1016/j.ress.2014.04.021.

[50] R. David, H. Alla, Continuous petri nets, in: Procceedings of the 8th European Workshop on Application and Theory of Petri Nets, Zaragoza, Spain, 1987, pp. 275–294, https://doi.org/10.1007/978-1-4471-4276-8-20.

[51] M.K. Molloy, Performance analysis using stochastic Petri nets, IEEE Trans. Comput. 31 (9) (2021) 913–917, https://doi.org/10.1109/TC.1982.1676110.

[52] S. Cantero-Chinchilla, J.L. Beck, M. Chiacho, J. Chiacho, D. Chronopoulos, A. Jones, Optimal sensor and actuator placement for structural health monitoring via an efficient convex cost-benefit optimization, Mech. Syst. Signal Process. 144 (2020), 106901, https://doi.org/10.1016/j.ymssp.2020.106901.

[53] C. Papadimitriou, Optimal sensor placement methodology for parametric identification of structural systems, J. Sound Vib. 278 (4–5) (2021) 923–947, https://doi.org/10.1016/j.jsv.2003.10.063.

[54] M. Chae, H. Yoo, J. Kim, M. Cho, Development of a wireless sensor network system for suspension bridge health monitoring, Autom. Constr. 21 (2021) 237–252, https://doi.org/10.1016/j.autcon.2011.06.008.

[55] A.H. Alavi, W.G. Buttlar, An overview of smartphone technology for citizen-centered, real-time and scalable civil infrastructure monitoring, Futur. Gener. Comput. Syst. 93 (2019) 651–672, https://doi.org/10.1016/j.future.2018.10.059.

[56] E. Ozer, M.Q. Feng, D. Feng, Citizen sensors for shm: towards a crowdsourcing platform, Sensors 15 (6) (2015) 14591–14614, https://doi.org/10.3390/s150614591.

[57] E. Ozer, M.Q. Feng, Structural reliability estimation with participatory sensing and mobile cyber-physical structural health monitoring systems, Appl. Sci. 9 (14) (2019) 2480, https://doi.org/10.3390/app9142840.

[58] J. Böke, U. Knaack, M. Hemmerling, Prototype of a cyber-physical façade system, J. Build. Eng. 31 (2020), 101397, https://doi.org/10.1016/j.jobe.2020.101397.

[59] T. Berners-Lee, R. Fielding, H. Frystyk, J. Gettys, J. Mogul, Hypertext transfer protocol-http, Available online, https://www.w3.org/Protocols/Specs.html, 1999. last access: April, 2022.

[60] Open group iot standard, Available online, https://www.opengroup.org/open-group-internet-things-work-group. last access: Abril, 2022.

[61] R.T. Fielding, Architectural Styles and the Design of Network-Based Software Architectures, Ph.D. thesis, University of California Irvine, 2000.

[62] J. Hietala, Real-Time Two-Way Data Transfer with a Digital Twin Via Web Interface, Master's thesis, Aalto University, 2020.

[63] G.O. Roberts, J.S. Rosenthal, Optimal scaling for various metropolis-hastings algorithms, Stat. Sci. 16 (4) (2021) 351–367, https://doi.org/10.1214/ss/1015346320.

[64] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: a survey, Futur. Gener. Comput. Syst. 97 (2021) 219–235, https://doi.org/10.1016/j.future.2019.02.050.