Article:

Bonney, M.S., de Angelis, M., Dal Borgo, M. et al. (2023) Contextualisation of information in digital twin processes. Mechanical Systems and Signal Processing, 184. 109657. ISSN: 0888-3270

https://doi.org/10.1016/j.ymssp.2022.109657

Mechanical Systems and Signal Processing 184 (2023) 109657

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Contextualisation of information in digital twin processes

Matthew S. Bonney [a,*], Marco de Angelis [b], Mattia Dal Borgo [c], David J. Wagg [a]

[a] Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, United Kingdom
[b] Institute for Risk and Uncertainty, University of Liverpool, United Kingdom
[c] Siemens Industry Software NV, Interleuvenlaan 68, 3001 Leuven, Belgium

ARTICLE INFO

Communicated by J.E. Mottershead

Keywords:
Digital twin
Trust
Crystal-box
Information management
DTOP-Cristallo
Contextualisation

ABSTRACT

Digital twins are required to process a large amount of data during operation, in order to achieve specific tasks, over the lifetime of the physical twin that they relate to. One important feature of processing data is the identification of trust in both the underlying data and processed information that arises from the data. Trust, as it is defined here, will typically be built from several contributory sources. While there are both quantitative and qualitative sources of trust, this paper focuses on the qualitative aspects of trust via the transparency of the algorithmic process that is available in the crystal-box modelling. The crystal-box idea is also extended to include the concept of a ‘crystal-box workflow’. The key idea is that in order to assist the user of the digital twin to interpret the results they are presented with, via the digital twin interface, the information needs to be contextualised. This work shows an example of how this can be done for a vibration testing (specifically modal testing) example on a scaled three-storey structure. The information is contextualised for the user via ‘profiles’, which collate and augment the processed information together. In particular, synthetic results are generated in order to augment a limited set of physically recorded data, and these synthetic results are then used to assist the user in contextualising the physically recorded data. Implementation results are shown using an open-source digital twin code called DTOP-Cristallo.

1. Introduction

Digital twins have been considered for use in a wide variety of systems and applications, as described in the recent review papers: [1–6]. The review in [1] describes the current challenges of the digital twin field, while [2] gives a detailed literature review of published work on digital twin technology. A review of the general concepts and applications is given in [3], while [4] reviews the internet-of-things aspects of digital twin and [5] describes the current state-of-the-art of digital twins with possible future directions for further development. Most recently [6] discusses issues associated with scaling current small scale digital twins to a larger industrial setting.

In practice, realisations of digital twins are achieved through hardware, software and network connectivity, which essentially requires an internet-of-things (IoT) capability for a typical scenario—see for example the review in [4].

In this paper the open-source system used to demonstrate the concepts is called DTOP-Cristallo, and recent papers describing its development and demonstrative examples can be found in [7–10].

One characteristic of a digital twin that is consistent across any application is the relationship between data, information, and trust. Raw data is gathered from sensors and contains the physical response of the system to a particular set of conditions (or loadings, dynamic excitation, etc.). Information obtained from the measured data, either through data processing or model-based

* Corresponding author.
E-mail address: m.bonney@sheffield.ac.uk (M.S. Bonney).

https://doi.org/10.1016/j.ymssp.2022.109657
Received 1 February 2022; Received in revised form 3 July 2022; Accepted 4 August 2022
Available online 16 August 2022

analysis, is typically used to provide useful insights regarding the key characteristics of the physical twin (which can include non-measurable quantities or predictions). Trust, as it will be defined here, relates to both quantitative and qualitative measures. For example, one typical quantitative measure describes the confidence in the information based on estimations of uncertainty in the data (e.g. [11,12]). Whereas a qualitative measure of trust could be knowledge regarding the reliability of the methods used to process the data over a period of time.

The concept of trust in a digital twin is strongly related to the quantification of uncertainty and the verification/validation of models. There is a wide variety of available open-source software that can address this aspect of trust. This includes OpenCossan [13], OpenTurns [14], and UQpy [15]. While this work does not directly address this quantification aspect of trust, the ideology that this paper uses has the flexibility to address both qualitative and quantitative aspects of trust. It is planned in the future to demonstrate the inclusion of one of these open-source software into the demonstrator digital twin to fully address all aspects of trust in a digital twin.

Digital twins contain multiple sources of information, such as information from processing the measured data and information generated from models contained within the digital twin. Due to this multiplicity of information sources, an important component in the development of a digital twin is a method to assess the levels of trust in the different information available. It is commonly assumed that processed sensor data has a higher degree of trust compared to simulated information. However, the simulations can make predictions for non-testable scenarios or quantities, so it is also important to have a measure of trust in this information as well. One way to help instill trust in these situations, is to use the models to simulate the experimental data (e.g. calibrate the models) within the digital twin. This serves two main functions, first to ensure that the calibration/ optimisation was performed as expected and secondly to reassure the user about the level of trust in the model.

To address the issue of trust in digital twins, this work introduces the concept of crystal-box workflow. This uses the concept of crystal-box modelling that enables users to see all aspects of the processing, (but not typically alter) and apply it to the step-by-step workflow processes within the digital twin. The purpose of the crystal-box workflow is the ability of the user to have full transparency on how the data is acquired, processed, and used. The transparency allows the user to interrogate the workflow in detail, therefore help build a qualitative trust in the data and predictions made. This work is the first work to explicitly implement this type of modelling framework within a digital twin. In this paper, initial work is performed on instilling a qualitative trust into a common structural dynamics workflow and provide the framework for future work to produce a quantitative measure of trust.

The term contextualisation is used in this work for the explanation of the interpretability and understanding of the software associated with the digital twin and the results it creates. In software engineering and computer science, the terms interpretability and explainability closely relate to the concept of trust and have unique definitions, but mainly described by a specific model or software. However, the multi-disciplinary connotations of these words imply different meanings, leading to the use of contextualisation in this work for a more universal understanding.

2. Digital twin platform and required tasks

This paper uses an open-source example, but it is recognised that some digital twin applications may require confidentiality that makes disclosing/sharing the data or derived information problematic. In this case a closed-source proprietary software product that ensures limitations on the information in the digital twin might be most appropriate. A consequence of this approach is that the flow of data and information becomes hidden (except for a limited set of users) [5]. For applications where data and information can be more openly shared, and particularly where digital twin *interoperability* is required, use of open-source digital twin software may be appropriate. Open-source digital twin concepts have been studied by several authors including the Internet of Things (IoT) platform called Eclipse Ditto [16] and the digital twin operational platform (DTOP), DTOP-Cristallo [7–9] that is described in this work.

The digital twin operational platform provides the interface to the user, and enables the user to request specific tasks to be performed. For example, the user may request that the digital twin processes experimental data, stores these processed results, and directly performs numerical simulations based on this data. This type of *workflow* within the digital twin typically processes the data directly after collection and then performs other parts of the workflow—in this case it compares the results with those from numerical simulations. The setting for this type of task could occur in a range of applications, for example those related to asset management-based digital twins. In particular, for the vibration testing task considered in this work, the assumption is made that the relevant workflow would be performed periodically during the life of the system and predictions would be made to help the user determine what kind of maintenance to carry out, and to aid in other types of operational decision making.

2.1. The DTOP framework

The DTOP framework is the underlying hardware and software used for the development and implementation of a digital twin. For this work the DTOP-Cristallo system was used. DTOP-Cristallo is an open-source digital twin platform, developed by the authors and their collaborators [7–10]. This platform has a web-based user interface that provides both accessibility and connectivity. The accessibility component of DTOP-Cristallo enables users to access the digital twin information remotely when it is needed. To allow this requirement, a browser-based interface has been developed using Python Flask [17]. Flask has several options for deployment based on the location of the user in relation to the host. These options can be seen in Fig. 1.

There are three main options for accessibility built into DTOP-Cristallo, all of which require no changes in the programming, only in the ‘‘server’’ deployment specifications. The first option is the standalone deployment. This only allows access to the digital

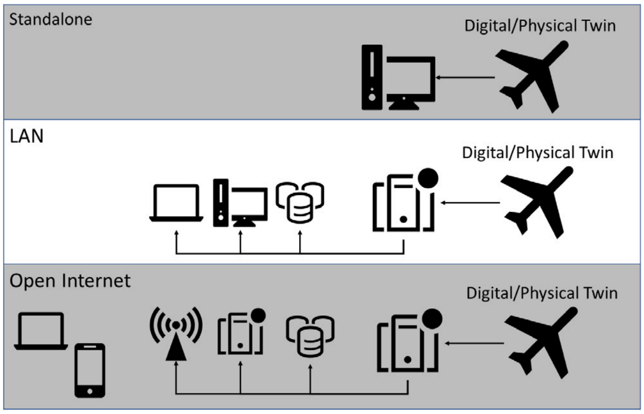

Fig. 1. Accessibility options for DTOP framework.

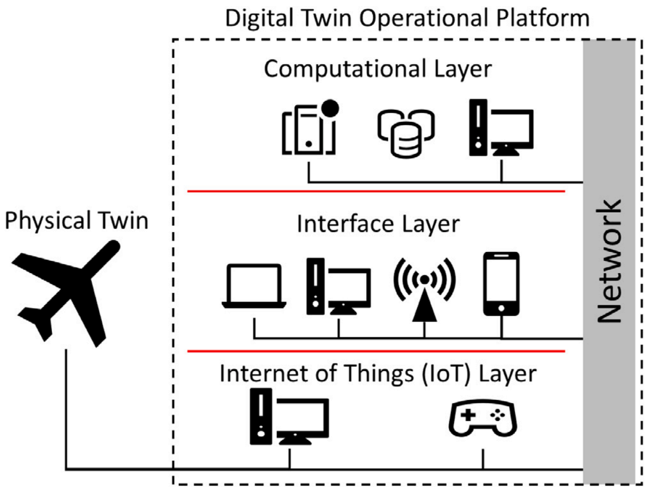

Fig. 2. Connectivity layers for DTOP-Cristallo.

twin on the single machine that the DTOP is deployed to. Using this option allows for very secure data management since only a singular machine/user can access this information, which is an important factor for certain digital twins such as in energy production facilities and government systems. This is thought of as an extension of traditional IoT platforms and inherently has a high level of data security due to the physical requirement of being at the specific machine.

The second accessibility option is the local area network (LAN) deployment. This extends the idea that a specific physical location is required, but generalises it to allow multiple users access at the same time through a LAN, which may (or may not) be separate from the internet depending on the context. For many complex engineering systems, several parts are designed, built, and maintained in parallel. In an aircraft, for example, the structural and electrical responses are analysed in nearly fully parallel via different sensors and users. Utilising a LAN deployment allows for multiple people on-site the ability to view and use the digital twin concurrently.

The final accessibility option removes the physical location requirement for the other options and allows for any approved user to access the digital twin with any internet access. In general, this decreases the data security, but it also greatly increases the usability and possible use cases. One example of a system that can significantly increase usability through the open internet deployment is for civil structures such as bridges and buildings. For a large number of these systems, it is either impractical or impossible (for systems like off-shore wind turbines) to require physical connection to the digital/physical twin.

The other main component of DTOP-Cristallo relates to the connection of the physical twin, digital twin, users, and computational resources. In general, this connection is described in three layers, as seen in Fig. 2. These layers primarily relate to how data, information, and knowledge is transferred (or flows) between the layers. The separation of the DTOP framework into layers is particularly useful in the discussion and research on digital twins. A large part of previous work on the DTOP framework focuses on either the interface layer [8,10] or the computational layer [7,9] independently. This work focuses on the interaction between stored and simulated data from the computational layer and contextualising it within the interface layer to aid in decision making, which is made by a user.

Fig. 3. Main page of DTOP-Cristallo.

In Fig. 2, the upper layer of the digital twin is the computing layer. This consists of the computational hardware that performs computationally intensive operations such as numerical simulations. Additionally, this includes other functions such as data storage and other services. This layer can either be physical hardware, such as a local high-performance computing cluster, or cloud-based depending on the available services already being used and the requirements of the system.

The bottom-layer for digital twin in Fig. 2 is the IoT layer that connects the digital twin with the physical twin. This primarily consist of two main components, the data-acquisition (DAQ) system and the controller. The DAQ system collects the sensor readings that are attached to the physical twin. This data is then transmitted to the network so the other layers can utilise this data for decision making or numerical simulations. The controller communicates with the physical twin in a two-way communication. These controllers typically use physical equipment such as servos/actuators to interact with the physical system and takes in sensor readings to ensure that the response of the system matches the intended operation (seismic absorption, change of angle of attack, or directional heading are all examples).

The final mid-layer provides connectivity between the user(s) and all other parts of the digital and physical twins. This interface layer determines the accessibility and usability for the user. The use of Python Flask is one way to implement this layer through the use of a browser-based interface via template HTML files.

2.1.1. DTOP-Cristallo

In this work, DTOP-Cristallo is used to create a demonstration digital twin for a 1:5 scaled three-storey structure constructed out of aluminium. This has been used in a variety of previous studies including [8], and for a general explanation about the implementation and also use in other engineering research, the interested reader can find details in [7,9]. The main page of DTOP-Cristallo shows both the available tools and a schematic of the system of interest—a screen shot is shown in Fig. 3.

2.2. Parameter estimation

To process the experimental and synthetically generated data, a tool within DTOP-Cristallo has been specifically created. This is the 'profile generator' tool that processes the data and generates a new set of model parameters that best fit the experimental data of the structural response found through the routine (i.e. periodic) testing and the addition of the synthetically generated data, which is based on the experimental data that is used to give context (for example, giving indications of parameter changes for asset management and maintenance scheduling).

The main mathematical process used in the profile generator is called circle fitting, and once the model parameters have been identified using this technique they are then stored in the 'profile hub' and can be used to perform various simulations that contextualise the experimental result. Here it is assumed that these can be used to aid the user in making decisions regarding the physical twin. These simulations can include non-testable scenarios, non-measurable quantities, and regeneration of calibration data to name a few possibilities. We note the following points regarding the implementation of this task:

- The experimental data is provided in a comma-separated value (CSV) format containing the frequency response functions (FRFs) at each sensor location, which can be written as,

$$[\mathbf{f},\ \ddot{\mathbf{X}}_1(j\omega),\ \ddot{\mathbf{X}}_2(j\omega),\ \ddot{\mathbf{X}}_3(j\omega)]\ , \tag{1}$$

Fig. 4. Selection of frequency response data file.

where the first column vector $\mathbf{f} = \{f_1, f_2, \ldots, f_p\}^T$ is the frequency vector in 'Hz', which represents the range and resolution of the spectra. The following column vectors $\ddot{\mathbf{X}}_1(j\omega), \ddot{\mathbf{X}}_2(j\omega), \ddot{\mathbf{X}}_3(j\omega)$ are the accelerances in 'g/N', where $\omega = 2\pi f$ is the circular frequency in 'rad/s' and $j = \sqrt{-1}$ represents the imaginary unit. These FRFs can be generated through various types of tests including impact hammer or shaker excitation.

- The application reads the data from a user selected CSV file containing the FRFs, as shown in Fig. 4. Then it segments the FRFs around each resonance (currently hard-coded but in the future will be user-selected) for each sensor, which gives,

$$\begin{bmatrix} \ddot{\mathbf{X}}_1^1(j\omega) & \ddot{\mathbf{X}}_1^2(j\omega) & \ddot{\mathbf{X}}_1^3(j\omega) \\ \ddot{\mathbf{X}}_2^1(j\omega) & \ddot{\mathbf{X}}_2^2(j\omega) & \ddot{\mathbf{X}}_2^3(j\omega) \\ \ddot{\mathbf{X}}_3^1(j\omega) & \ddot{\mathbf{X}}_3^2(j\omega) & \ddot{\mathbf{X}}_3^3(j\omega) \end{bmatrix}, \tag{2}$$

where each of the shortened FRFs $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$ represents a simplified single degree-of-freedom (DOF) system that matches traditional structural dynamic theory, where $\alpha = 1, 2, 3$ refers to the DOF number and $\beta = 1, 2, 3$ refers to the mode number, where this system only contains 3 modes of interest. In the complex plane, each $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$ resembles a partial circle if the monitored structure is lightly damped and the resonances are well separated [18,19]. There is a vast amount of literature [18–21], which demonstrates how fitting a circle to this data leads to identifying the modal parameters and mode shapes of the structure. A brief overview on how this method has been implemented in DTOP-Cristallo is given in Appendix A.

- After circle fitting each mode at each sensor location, the system matrices of the lumped parameter model can be calculated, as displayed in Fig. 5. This process is used to estimate the mass, stiffness and damping values of the lumped parameter model that are then stored in the 'profile hub'. The structure is modelled as three mass–spring–damper systems connected in series to each other and to the ground at one end. Appendix B contains the mathematical formulation of the lumped parameter model. The mass matrix is calculated as,

$$\mathbf{M} = m_m \boldsymbol{\Psi} \boldsymbol{\Psi}^T, \tag{3}$$

where m_m is the modal mass, which is given by $m_m = m_{tot}/3$, with m_{tot} the total mass of the structure, and $\boldsymbol{\Psi}$ is the matrix of experimentally determined mode shapes. The stiffness matrix is given by,

$$\mathbf{K} = m_m \boldsymbol{\Psi} \boldsymbol{\Lambda}^{-1} \boldsymbol{\Psi}^T, \tag{4}$$

where $\boldsymbol{\Lambda}^{-1}$ is the diagonal matrix of squared natural frequencies, and the damping matrix is taken to be,

$$\mathbf{C} = m_m (\boldsymbol{\Psi}^T)^{-1} \boldsymbol{\Xi} \boldsymbol{\Psi}^{-1}, \tag{5}$$

where $\boldsymbol{\Xi}$ is defined as,

$$\boldsymbol{\Xi} = \begin{bmatrix} 2\eta_{n,1}\omega_{n,1} & 0 & 0 \\ 0 & 2\eta_{n,2}\omega_{n,2} & 0 \\ 0 & 0 & 2\eta_{n,3}\omega_{n,3} \end{bmatrix}, \tag{6}$$

where the damping ratios $\eta_{n,\beta}$ and natural frequencies $\omega_{n,\beta}$ are estimated using the derivation presented in Appendix A. From the matrices given by Eqs. (3), (4) and (5), the model parameters (m_1, m_2, m_3), (k_1, k_2, k_3), and (c_1, c_2, c_3) can be approximated using the notions of a three DOF system fixed at one side and free at the other end, as presented in Appendix B. Due to the orthogonality of the mode shapes, the mass matrix $\mathbf{M}$ given by Eq. (3) is diagonal and the mass of each individual floor can be identified as an element of the diagonal. The stiffness matrix $\mathbf{K}$ given by Eq. (4) is fully populated with elements $k_{ij} \quad \forall\, i = 1, 2, 3 \quad \& \quad \forall\, j = 1, 2, 3$, thus the individual stiffness elements can be calculated as,

$$\begin{aligned} k_3 &= (k_{33} - k_{32} - k_{23})/3 \\ k_2 &= (k_{22} - k_3 - k_{21} - k_{12})/3 \;. \\ k_1 &= k_{11} - k_2 \end{aligned} \tag{7}$$

Similarly, the damping matrix $\mathbf{C}$ given by Eq. (5) is also fully populated with elements $c_{ij} \quad \forall\, i = 1, 2, 3 \quad \& \quad \forall\, j = 1, 2, 3$, thus the individual damping elements can be calculated as,

$$\begin{aligned} c_3 &= (c_{33} - c_{32} - c_{23})/3 \\ c_2 &= (c_{22} - c_3 - c_{21} - c_{12})/3 \\ c_1 &= c_{11} - c_2 \end{aligned} \tag{8}$$

Note that, in the absence of a more physically meaningful model, the damping coefficients are assumed to be directly proportional to the stiffness coefficients—see Appendix B. This is a widely accepted practice in the modelling of lumped mass systems, but as will be seen later in Section 4, leads to limitations in the matching between experimental and numerical results. This type of limitation in the physical-based model is called *model-form error* and is a form of epistemic uncertainty that is typically present in all digital twins of complex engineering systems. Although there are more sophisticated damping models that could be used, in this paper we accept this level of model-form error in the damping model, and comment on it's consequences in Section 4.

- The final step is to store the model parameters as a new profile in CSV format, which can then be used by the other tools of DTOP-Cristallo. Currently, this uses the CSV format, but future implementations can easily be stored in a database structure such as SQL.

2.3. Simulated data generator

The methodology described in Section 2.2 provides an efficient means to determine the parameters of the three-storey structure, under the assumption of linear dynamics. Given a single time-history converted to a FRF, the technique can be deployed to determine the stiffness vector $k = [k_1, k_2, k_3]$ and the mass vector $m = [m_1, m_2, m_3]$, as well as the damping vector $c = [c_1, c_2, c_3]$ via least-squares regression.

While it is not part of a predictive workflow, it is important to verify that the process for identification is suitably accurate and gives additional context to the user for instilling trust in any future predictions made using these parameter values. To aid in this verification, a synthetic data generator tool was designed to generate time-histories using the identified parameters within an ordinary-differential-equation (ODE) model of the three-storey structure (i.e. a 3 DOF lumped-mass system is assumed for the ODE model). The resulting synthetic displacement time-histories can then be transformed into the frequency domain to display the associated frequency characteristics, such as FRF or power spectral density (PSD), that can be compared to the experimental data.

Depending on the exact requirements of the application, other synthetic features can also be added at this stage. For example, nonlinear terms can be added to the ODE equations-of-motion and generate synthetic data with nonlinear dynamical characteristics (such as large amplitude deflections) if the user desires this type of data. In this way, experimental data produced by a particular means of exciting the structure (in this case a hammer excitation was used) can be augmented with a range of synthetically generated data simulated by the digital twin to enable the user to have a better contextual understanding of the data, and the behaviour in parameter regions close to the values where that data was collected. This has the potential to be particularly useful in an asset management context, where slow, but significant parameter changes over time are often expected, induced by wear, ageing and material degradation.

The use of the data simulator is therefore twofold: (i) it can be used to verify the profile generation procedure; (ii) it can be used by the DTOP to produce additional synthetic data of the physical twin to contextualise the physically recorded data. An example of the results from the simulator is shown in Fig. 6. This simulator is programmed with the capability to mimic a variety of scenarios. For the results shown in Fig. 6, a monitoring scenario is used with a sudden impact to mimic a collision/ hammer impact. This is interpreted as corresponding to non-zero initial conditions of the system before the hammer excitation (modelled as an exponential of negative time squared) is applied to the system after half a second of recording data.

Fig. 5. Model parameter estimation and profile generation.

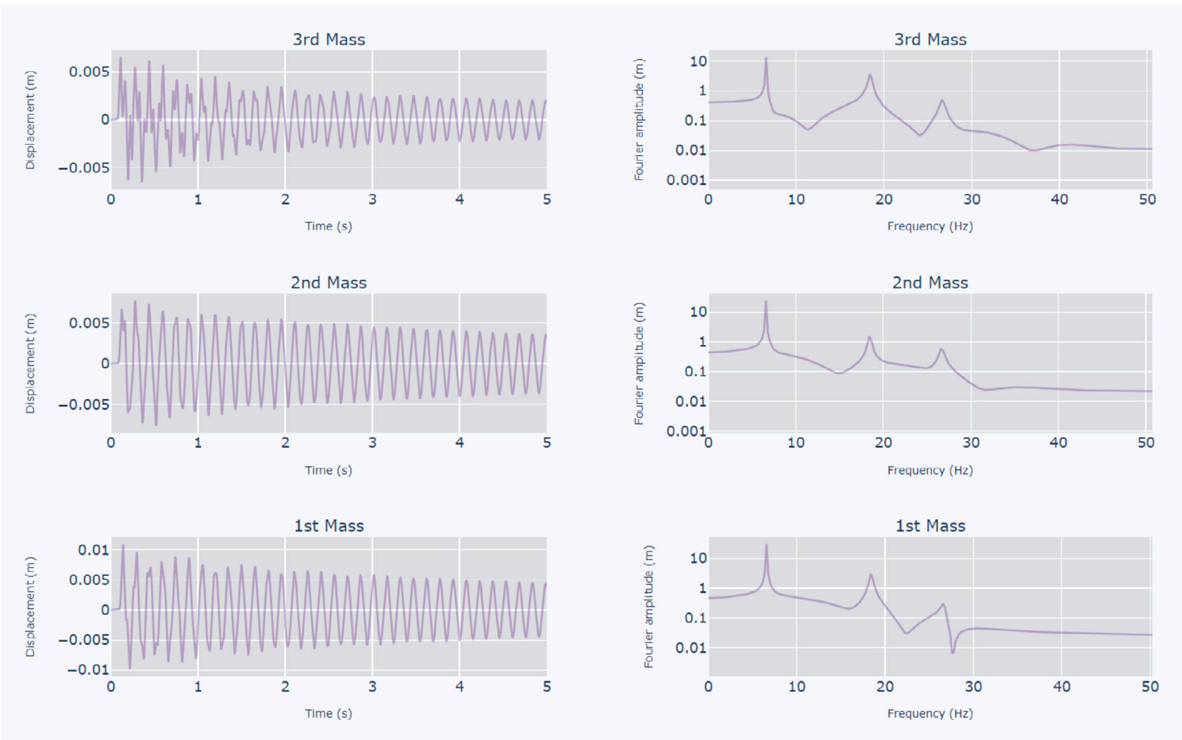

Fig. 6. Outputs from the hammer testing data simulator tool. On the left an example of the displacement time-histories of each of the three masses is shown, and on the right the corresponding Fourier amplitude for each mass. The hammer tool is simulated with zero initial conditions and with the hammer hitting first mass at time $t = 0.1$ s with a force intensity of $F = 300$ N.

3. Trust in digital twins

One of the most potentially useful characteristics of digital twins is the association of digital twins with decision making. The digital twin is commonly used as a collection of data from sensors, simulations, and expert knowledge with future intentions to introduce automation for low-level decision making. Despite this planned future aspiration, a large number of digital twins are used to supply information to the user to aid in the decision making process. These decisions range from normal operations to repair scheduling to accident avoidance. Because of this utilisation of information in decision making, the concept of "trust" is important when discussing the transparency and traceability of information used in decision making.

The concept of trust in regards to information relates heavily to the uncertainty analysis of the simulation information that is informed by experimental data. This requires a holistic approach to uncertainty quantification and understanding of the simulations. The uncertainty quantification aspect has an intrinsic relationship to trust. When a measurement has very low levels of uncertainty, the predictions made from this data is (normally) considered to be well trusted since the variability is quite small. This is also known as quantifiable trust.

In addition to having a robust uncertainty quantification performed as part of the simulation, it is also important for trust if the user understands the assumptions and techniques that the simulations are performing, also called qualitative trust. Some of the best methods to increase this trust is for the predictions to be based on a fundamentally known theory that has been accepted and validated multiple times in the research and practitioner communities over a number of years. For example, finite element analysis or computational fluid dynamics would fall into this category. An alternative is, especially for less well established methods, is to provide robust documentation that explains both the derivation and implementation for the simulation.

Another method that could be used to increase the trust is through the ability to examine the implementation code directly. This can be done via one of two methods with the first being a pure open-source simulation code. The use of open-source code can greatly increase the trust, since the user is able to see the exact implementation of the algorithm. However, the aspect of open-source simulation code where anyone can modify it needs to be applied with caution, since the modifications might introduce errors or invalidate the derivation. To circumvent these issues, the use of crystal-box modelling (as opposed to black, white, or grey-box modelling) allows the user to fully understand the implementation while any modifications can only be carried out by approved and verified users.

3.1. Crystal-box modelling

In software engineering and computer science, the literature on verification and validation is vast—see for example [22–24] and references therein. In this context, verification and validation typically takes place in the form of tests. The concept of *white-* or *glass-* or *clear-box testing* is well established. A white-box *test* refers to the testing of computer programs at the level of the source code, that is aimed at creating an error-free environment by examining all code. However, in engineering, verification and validation relate to the various models used and less on the software used to evaluate these models. In engineering, a white-box model is a physics-based model whose assumptions and boundary conditions are clearly stated; whereas a black-box model is based on an input–output relationship (e.g such as that obtained using regression or a deep-learning models) whose internal relationships are determined by providing the best-fit to any given data. A grey-box model is the combination of white- and black-box components for enhanced modelling. In this paper, the concept of grey-box model is taken to the digital twin level via an augmented transparency dimension which is not limited to the programming logic, but extends to data collection and data sharing processes, as well as to the machine-learning code used for model validation. In this context, a crystal box can be considered to be a see-through grey box model with an enhanced transparency dimension.

A crystal-box model as an abstraction of a grey-box model (combination of physics- and data-based models), is an emerging concept [25,26]. A crystal-box model can be seen as a grey-box model whose internals are accessible, human-readable and inspectable. The definition of such a model is not limited to open-source code, but extends to code that is protected (as opposed to open) and is provided alongside its grammar and compiler, see Fig. 7 for a illustrative summary. The code present in a crystal-box includes the simulation model code and the code used to fetch the experimental data. So, a crystal-box is a grey box with an augmented transparency dimension. With this framework, the user can inspect the code and understand its limitations, as well as understanding how the empirical data propagates from source to model.

When a user is presented with a crystal-box, an informed decision can be made because, even though all the modelling assumptions are not clearly stated, it is possible to pinpoint them by inspecting the code. No model is perfect, so the crystal-box framework guides the users (who have been given the model) to make their own informed decisions and establish if the model is good enough. The peculiar component of transparency unlocks the potential and adds an attractive verification feature to the framework, which can now be inspected for modelling errors. It can be argued that this kind of framework provides the natural space for model-form uncertainty and uncertainty propagation to be developed further.

Two main approaches can be envisaged to implement the crystal-box framework to increase trust in a digital twin: a *quantitative approach* whereby verification takes place automatically by means of uncertainty propagation; a *qualitative approach* which consists in building the infrastructure to enable user inspection and implement transparency. This general concept can be visualised in Fig. 7. Quantitative approaches to crystal-boxes are described in [25,26], where the idea of a crystal-box modelling was introduced as a stepping stone to rigorous (intrusive) automatic uncertainty propagation. With automatic uncertainty propagation, trust in scientific calculations can be engineered and checked systematically. This process is also known as automatic verification, which lies at the very core of trust. In this paper however, the focus is placed on the qualitative aspect.

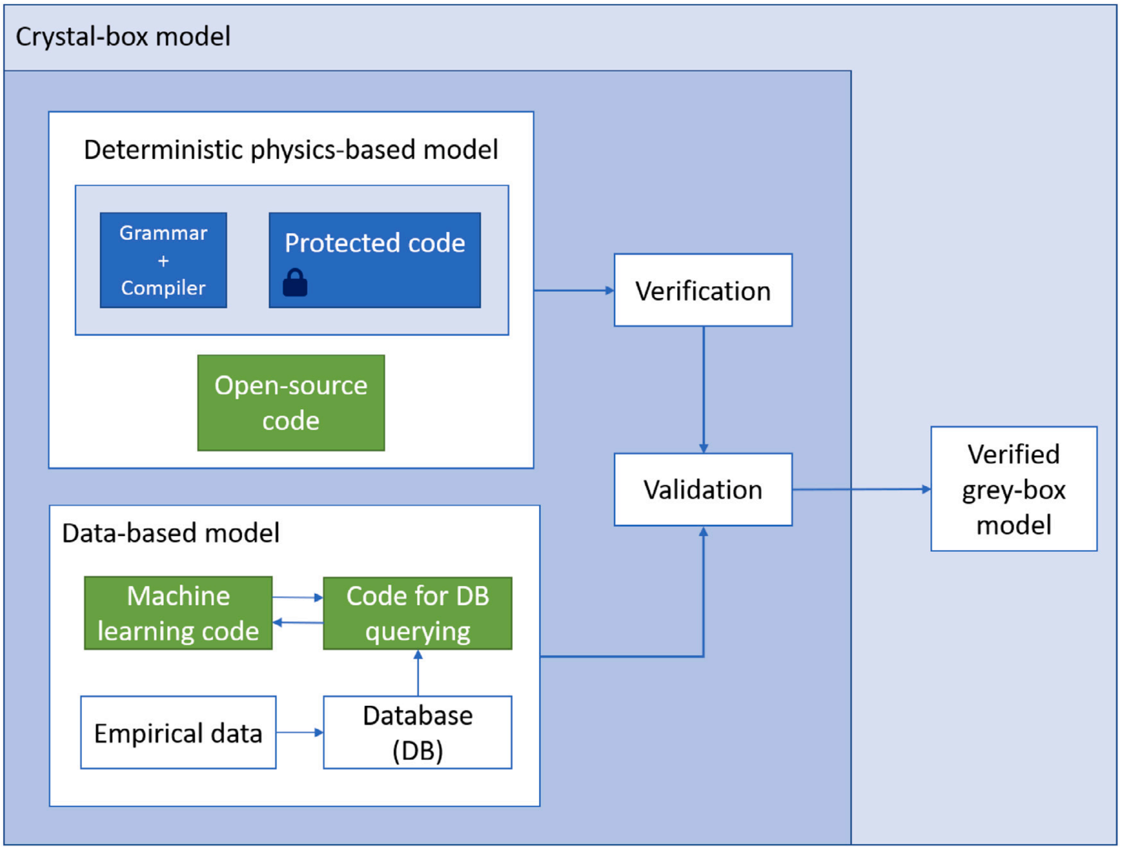

Fig. 7. Crystal-box vs. grey-box model diagram. In a traditional grey-box model, deterministic physics-based and data models are combined via validation. In a crystal-box framework, the code that is behind the models can be inspected by the user. The green boxes denote open source, while the dark blue boxes denote that the code exists but is protected for proprietary reasons. The set of protected code, grammar and compiler is enough for a designated agent or bot to perform verification and inspection tasks without revealing the "secrets" of the protected code.

The qualitative approach builds upon rigorous information management, where data can flow from source to end while retaining all the vital semantics that makes that particular instance of data valuable and unique. The information management process greatly benefits from transparency of simulations and experimentation. Digital twins can act as the lenses through which decision makers can inspect the physics of a particular asset. The idea of crystal-box modelling can also be extended to describe the specific workflow in addition to individual models/simulations. This idea relates to the traceability and transparency of the processing of data into information. While the data is held within the metaphorical crystal, all processes and conditioning of the data into information can be seen and replicated. This transparency and repeatability helps build the trust in the processes used within the digital twin to take the raw data into usable information for decision making.

3.2. *Contextualisation and workflow*

Contextualisation is achieved by implementing the crystal-box model into a workflow within the digital twin. Contextualisation can be seen as the continuous comparison of validated profiles. A profile is the collection of *flat* data files (e.g. JSON-like) that fully characterise the state of the digital twin at any given update. A workflow is the high-level process that leads to a profile, and consists in (1) collecting the empirical data, (2) producing the simulated data, and (3) updating the model. The *profile hub* tool is then able to take a profile and populate the various simulation tools with the newly updated values. While there are many options for simulation tools built into a digital twin, this work demonstrates the contextualisation tool that is currently in development. The workflow implements the idea of transparent operations, whereby the user can browse the history of available profiles, for example by setting the clock back to a past update or see the different updates in one place. This workflow is shown in Fig. 8, which shows the processes that occur to generate a calibrated model from the experimental data via generation of synthetic data. While this particular workflow does not make statistical predictions, this will nonetheless enhance user confidence and instil trust in the digital twin model.

Fig. 8 can be seen as the contextualisation of the crystal-box model depicted in Fig. 7 to a particular digital twin application. For example, the circle fit component in Fig. 8 is a standard robust regression strategy for estimating the natural frequencies and modal

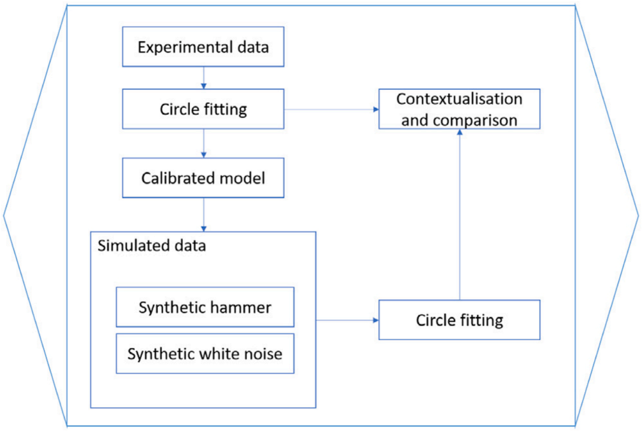

Fig. 8. Contextualisation of modal information for calibrated models.

information of a three-degree-of-freedom linear structure, which makes no distributional assumptions, and can be used to compute confidence intervals (although not demonstrated in this work). The *simulated data* component that leads to synthetic hammer or white-noise data are an implementation of the physical model depicted in Fig. 7. This component can be verified with additional code for automatic uncertainty propagation. The *calibrated model* is an instance of validation, while the contextualisation is an instance of a robust grey-box model.

4. Demonstrative example

As previously mentioned, an important role of the digital twin is to provide information to a user to aid in decision making. This information contains aspects such as uncertainty quantification and contextualisation. Uncertainty quantification is a widely studied area of research, both for data analysis [12,25] and digital twins [2,5]. However, the properties of contextualisation and the qualitative trust are not commonly studied. This demonstration will focus solely on the qualitative aspects of trust through contextualisation via the crystal-box workflow.

One important aspect to note about this demonstrative example is that this work is not focused on the development of accurate digital twin models. The focus is to provide the relevant information to the user about the chosen model. There are a myriad of models that can be used to represent the system and each has its own associated modelling assumptions that lead to model-form errors. The purpose of this contextualisation is to provide information related to the model to the user to make informed decisions. One aspect of this contextualisation is the ability to provide abstracted information about the models and data. The user is able to see the data, how that data was generated, and how an associated model was created if there is one.

To demonstrate this contextualisation, this section goes through a typical process from taking experimental data, producing a calibrated model through the circle fitting procedure, creating synthetic data, and comparing the synthetic data to the experimental data (i.e. verification and contextualising). This comparison is the main focus of the contextualisation, which firstly demonstrates the accuracy of the model calibration and instills trust for the user in the simulations that will be made from the calibrated model.

4.1. Model calibration and synthetic data generation

The crystal-box workflow demonstrated in this paper is shown in Fig. 8. This shows all the processing that occurs to the data before it is displayed to the user for context. Using this crystal-box formulation gives several advantages to traditional verification workflows. Firstly, the use of a crystal-box formulation allows for a transparent and traceable flow of data. This provides additional information such as any filtering used, how the model was generated, and any extra parameters that might not be present in the data (such as excitation type to generate the FRF). In addition to this transparency of information, providing access to how the data is generated allows for interoperability and the potential of transfer learning. The first step in using this workflow is processing the experimental data and generating the calibrated model. This is done through the circle fitting routine described in Section 2.2. After the circle fitting procedure, this information is stored into a CSV file that serves as the archived profile. This profile contains the fitted model parameters (mass, stiffness, and damping) and the modal characteristics of the data (natural frequencies, damping ratios, and mode shapes). The current implementation generates this profile locally, but planned work is to incorporate these profiles into a database along with the data itself to reduce local requirements for using the operational platform. As shown in Fig. 1, there are multiple version of the DTOP available, and by using a centralised database, the use of the LAN and open-internet becomes simpler due to the use of network information compared to local file IO.

After the calibrated model is stored as a profile, the model simulates two types of excitation to generate a time history. These types of excitation include a hammer impact excitation to mimic the calibration data and a burst random (also known as white noise) to get a better understanding of a wider frequency bandwidth range. These types of excitation are common to better understand the general dynamics of the system.

Table 1
Tabular comparison of multiple profiles.

Profile name	ω_1 [Hz]	ω_2 [Hz]	ω_3 [Hz]	η_1 [%]	η_2 [%]	η_3 [%]
Profile (1) Experimental	6.500	17.469	26.500	1.07	1.04	0.93
Profile (2) Hammer	6.200	17.800	26.411	1.96	3.45	3.90
Profile (3) White noise	6.200	17.767	26.400	3.13	2.54	4.71

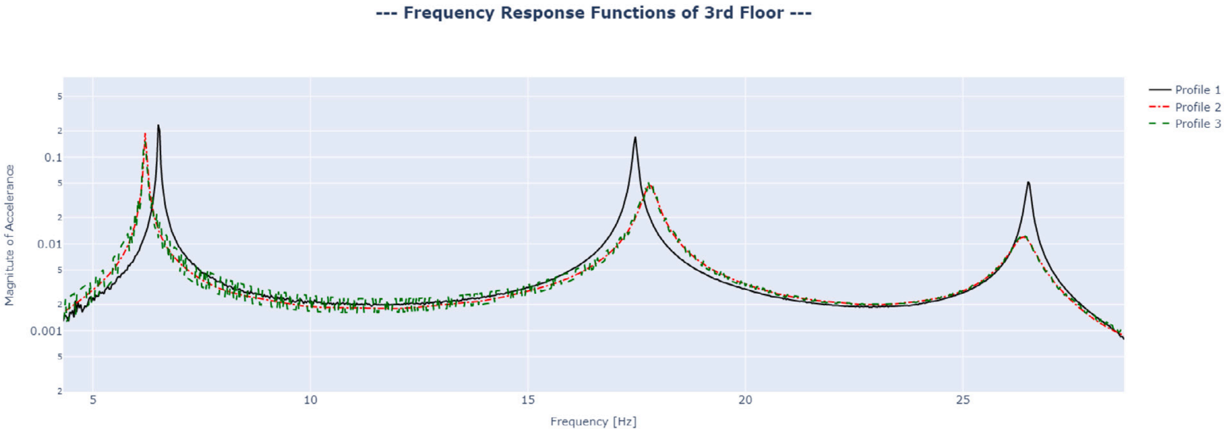

Fig. 9. Comparison of raw FRF data.

4.2. Contextualisation of information

After these time histories are generated for the synthetic data, the equivalent FRF is generated then processed through the same circle fit routine. This stores the results into separate profiles that can be compared. These three are then compared in three main ways: tabular results, FRF graphical comparison, and a comparison of the mode shapes. The first comparison, the tabular results, are shown in Table 1.

This table contains two main pieces of information for each of the three modes of interest, the natural frequency and damping ratio. There are a few things to note about these results. The first main point to note is that the synthetic data produced higher damping ratios compared to the experimental data. There are two main contributions to this. The first is the damping model selected; In the profiling routine, it is assumed that there is viscous damping. While this selection is common in linear vibrations, there is still an epistemic uncertainty associated with this selection. While this is expected to be the main source of the discrepancy, other reasons were also investigated. For the FRFs, there is a frequency resolution of 0.030 Hz for both the experimental and numerical results. However, since this system is lightly damped, the damping ratio calculation is highly sensitive to the resolution of the resonance. To test this, a simple interval analysis was performed based on half of the resolution. Performing this analysis showed that the experimentally determined damping ratio had a range of 0.005 centered at the values found in Table 1. While this does not account for the full 0.03 difference found between the experimental and numerical damping ratios, it is a sizable factor. The remaining difference is assumed to be a mixture of numerical damping introduced in the ODE solver and the model-form error for the selected damping model, described in Section 2.2. The ability to investigate this sensitivity and aid in identifying the sources of this error is another great benefit of the crystal-box workflow. Since all the processes can be investigated, the user can identify that the frequency resolution is a sizable source of error between the model and experimental data.

In addition to the increase in damping, there is also a change in the natural frequency. The largest change was seen in the fundamental frequency where there was a 4.6% difference. Some of this difference is due to the increase in damping, however the primary factor for this discrepancy is model-form error. Specifically because our implementation assumed a lumped mass model for each storey of the structure. This ignores factors such as the rotational inertia of the system, and the mass distribution in the columns. Despite this model-form error (except for the damping) the model is still 95% accurate. It is important to note, that most digital twins will contain models with significant model-form errors. It is generally not possible to completely eliminate the error, so it should be quantified and then propagated into the predictive simulations to provide a quantitative measure of trust for the simulation predictions.

A comparison used to give context to the model is the comparison of the raw FRF spectra, which can be seen in Fig. 9. These FRFs shown in Fig. 9 demonstrate the same differences that are presented in Table 1 but in a graphical format. The first noticeable aspect is the decrease in the amplitude of the response particularly for the second and third mode. Primarily, this is due to the increase in damping that is associated with the synthetic data. It is well understood that the amplitude of the resonance peak is inversely proportional to the damping present. Since the numerical simulations produced larger damping, they also produce lower amplitude than the experimental data. As mentioned above, this is due to the limitation of the model (i.e. the model-form error).

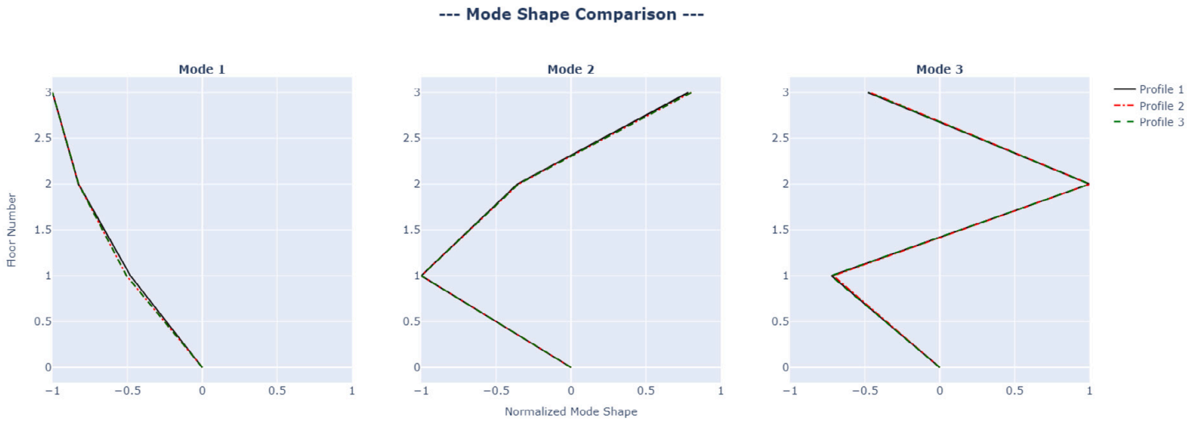

Fig. 10. Comparison of the first 3 mode shapes of the data.

The other main difference between the profiles in Fig. 9 is the difference in the peak frequency. These differences can be used to perform a sanity check on the profile generator tool by checking the peak using the cursor. DTOP-Cristallo presents all figures as a javascript object, so the user can perform operations such as zoom, pan, and save the figure to provide better contextualisation. While this comparison might be difficult when looking at all three natural frequencies, the ability to zoom allows the user to better show the differences between the profiles. The final contextualisation is a comparison of the determined mode shapes shown in Fig. 10.

The mode shapes shown in Fig. 10 show that the calculations represent the same modal characteristics. While there are slight differences between the profiles, the general shapes are the same and the difference are mainly caused by the slight differences in the calculated mass and stiffness values. If the differences were uniform (the same difference for each floor), the shapes would be identical. However if there are small differences between each floor's error, slightly different mode shapes will be calculated.

5. Conclusions

Trust in a digital twin is comprised of several components including the trust in the data, understanding of the processing, uncertainty quantification, information management, and transparency in predictions. These components consist of both quantifiable and qualitative confidence. Uncertainty quantification can lead to numerical indicators of confidence while the understanding of the data processing is qualitative but just as important when making decisions. To identify the qualitative trust in a digital twin, this work focuses on the transparency and repeatability of data through the use of crystal-box workflows. The crystal-box modelling methodology introduces a transparent and repeatable processing of data and models. While the initial introduction of crystal-box modelling focuses on the intrusive uncertainty propagation (also important to quantifying digital twin trust), this work takes the methodology and applies it to the data workflow. This shows every manipulation and prediction on the data and information to instill repeatability of predictions and qualitative trust to the user of the digital twin.

The example shown in Section 4 highlighted the well known limitation of using a lumped mass physics-based model with a proportional damping model. As discussed in Section 4, the resulting model-form error led to limitations in the matching of experimental and simulated data (e.g. a reduction in quantitative trust). However, using a crystal-box workflow is a way of enabling the user to understand (at least in large part) why this is occurring and therefore increasing qualitative trust.

This work demonstrates the capabilities that are inherent in digital twins through the use of model calibration, data processing, and storage to establish trust in decision making. One of the most important aspects of qualitative trust is providing contextualisation of novel data to the user. The contextualisation on which this work focuses is the ability to reproduce the data used to generate the model. This reproducible nature of the models establishes the baseline level of trust for these models to be used in practice. Commonly, these models are used to make predictions such as untestable scenarios or immeasurable quantities. To demonstrate the instillation of trust to the user, this work compares model properties and modal characteristics of the data for both experimental and synthetic data. This contextualisation gives the user an understanding of the model that is designed to provide information to aid in decision making.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: David Wagg reports financial support was provided by Engineering and Physical Sciences Research Council.

Data availability

available on GitHub https://github.com/Digital-Twin-Operational-Platform/Cristallo.

Acknowledgements

The authors would like to acknowledge the support of EPSRC, United Kingdom via the digital twins for improved dynamic design project (grant number EP/R006768/1). For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) license.

Appendix A. Circle fitting theory

This appendix presents an overview of the circle fitting method discussed in Section 2.2 with the intention to show how it was implemented in DTOP-Cristallo. Solving the least squares problem of circle fitting corresponds to finding the centre of the circle $(x_c,\ y_c)$ and its radius r_c, which minimise the residuals function. This application solves the nonlinear problem of least squares circle fitting with an optimisation tool that makes use of the Jacobian of the residuals function for computational speed using a modified Levenberg–Marquardt algorithm. A brief summary of the process is given here. The residual function is defined as,

$$\phi = \sum_{i=1}^{m}(r_i - r_c)^2, \tag{A.1}$$

where m is the number of frequency points in each $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$, and r_i is the distance of each data point in the complex plane from the centre, which can be written as,

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}, \tag{A.2}$$

where $x_i = \Re\mathfrak{e}\{\ddot{\mathbf{X}}_\alpha^\beta(j\omega_i)\}$, and $y_i = \Im\mathfrak{m}\{\ddot{\mathbf{X}}_\alpha^\beta(j\omega_i)\}$. The minimisation of Eq. (A.1) is then carried out using the *Scipy optimise least-squares* function within Python, where the Jacobian of Eq. (A.1), given by

$$\begin{aligned} J_{i,x} &= \frac{\partial\phi_i}{\partial x_c} = \frac{x_c - x_i}{r_i} \\ J_{i,y} &= \frac{\partial\phi_i}{\partial y_c} = \frac{y_c - y_i}{r_i}, \end{aligned} \tag{A.3}$$

has been pre-calculated for computational speed. The quantity r_c is measured as the mean distance of the entries to the centre via

$$r_c = \frac{1}{m}\sum_{i=1}^{m} r_i. \tag{A.4}$$

When the algorithm converges, the centre $(x_c,\ y_c)$ and radius r_c of the circle that best fit each $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$ are saved in the application and used for the parameter estimation.

The application then processes each one of these circles and calculates various structural characteristics such as natural frequency $f_{n,\beta}$, which is taken as the frequency with the largest angular displacement between two adjacent points of $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$ in the complex plane; and the damping ratio, which is calculated as,

$$\eta_{n,\beta} = \frac{|f_{n-1,\beta}^2 - f_{n+1,\beta}^2|}{f_{n,\beta}^2[\tan(\frac{\theta_{n-1,\beta}}{2}) + \tan(\frac{\theta_{n+1,\beta}}{2})]}, \tag{A.5}$$

where $f_{n-1,\beta}$ refers to the frequency point before resonance, $f_{n+1,\beta}$ is the frequency point after resonance and $\theta_{n-1,\beta},\ \theta_{n+1,\beta}$ are the phase angles of the corresponding $\ddot{\mathbf{X}}_\alpha^\beta(j\omega)$ points. The application then calculates the modal amplitudes from the relative diameters and locations on the complex plane of the circles that fitted the experimental data, determining the matrix of mode shapes, which can be written as,

$$\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{X}_1^1 & \mathbf{X}_1^2 & \mathbf{X}_1^3 \\ \mathbf{X}_2^1 & \mathbf{X}_2^2 & \mathbf{X}_2^3 \\ \mathbf{X}_3^1 & \mathbf{X}_3^2 & \mathbf{X}_3^3 \end{bmatrix}, \tag{A.6}$$

where $\mathbf{X}_\alpha^\beta$ represents the modal amplitude of mode β at location α, and is calculated as follows,

$$\mathbf{X}_\alpha^\beta = 2r_{c,\alpha}^\beta\ \eta_{n,\beta}\ \omega_{n,\beta}^2\ \mathrm{sign}[x_{c,\alpha}^\beta], \tag{A.7}$$

where $2r_{c,\alpha}^\beta$ is the diameter of the circle, and $\mathrm{sign}[x_{c,\alpha}^\beta]$ specifies if the location of the centre of the circle is on the positive or negative side of the real axis of the complex plane. Each modal vector is then normalised with the Euclidean norm.

Appendix B. Lumped parameter model

A three DOF model of the structure shown in Fig. 3 can be derived by assuming that each floor can be represented by a lumped mass and the beams connecting the storeys as springs with negligible mass.

The equation of motion of the lumped parameter model shown in Fig. B.11 can then be written as,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}_{ext}, \tag{B.1}$$

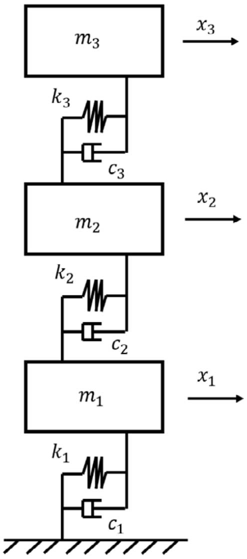

Fig. B.11. Lumped parameter model of the structure.

where $\mathbf{x} = \{x_1\ x_2\ x_3\}^T$ is the displacement vector, $\mathbf{f}_{ext} = \{f_1\ f_2\ f_3\}^T$ is the vector of external forces acting on each storey, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the classical mass, damping and stiffness matrices of the three DOF model fixed at one end and free at the other, which turn out to be,

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}. \tag{B.2}$$

References

[1] Aidan Fuller, Zhong Fan, Charles Day, Chris Barlow, Digital twin: Enabling technologies, challenges and open research, IEEE Access (8) (2020) 108952–108971.

[2] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, Ben Hicks, Characterising the Digital Twin: A systematic literature review, CIRP J. Manuf. Sci. Technol. (2020).

[3] Mengnan Liu, Shuiliang Fang, Huiyue Dong, Cunzhi Xu, Review of digital twin about concepts, technologies, and industrial applications, J. Manuf. Syst. (2020).

[4] Roberto Minerva, Gyu Myoung Lee, Noel Crespi, Digital twin in the IoT context: a survey on technical features, scenarios, and architectural models, Proc. IEEE 108 (10) (2020) 1785–1824.

[5] D.J. Wagg, K. Worden, R.J. Barthorpe, P. Gardner, Digital Twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications, ASCE-ASME J. Risk Uncertain. Eng. Syst. B: Mech. Eng. 6 (3) (2020) 030901, 5.

[6] Steven A Niederer, Michael S Sacks, Mark Girolami, Karen Willcox, Scaling digital twins from the artisanal to the industrial, Nat. Comput. Sci. 1 (5) (2021) 313–320.

[7] Matthew S. Bonney, Paul Gardner, David Wagg, Robin Mills, Case Study of Connectivity of Digital Twins and Experimental Systems, in: Proceedings of the 8th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, 2021, pp. 1416–1425.

[8] Matthew S. Bonney, Marco de Angelis, Mattia Dal Borgo, David Wagg, Digital Twin Operational Platform for Connectivity and Accessibility using Flask Python, in: Proceedings of the 24th International Conference on Model-Driven Engineering Languages and Systems, 2021.

[9] Matthew S. Bonney, David J. Wagg, Robin Mills, Seismic Response of a 3-Storey Structure Under Variable Environment Conditions, in: Proceedings of the 17th World Conference on Earthquake Engineering, 2021.

[10] M.S. Bonney, M. de Angelis, M. Dal Borgo, L. Andrade, S. Beregi, N. Jamia, D.J. Wagg, Development of a digital twin operational platform using Python Flask, Data-Centric Engineering 3 (2022) e13, http://dx.doi.org/10.1017/dce.2022.13.

[11] Paul Gardner, Mattia Dal Borgo, Valentina Ruffini, Aidan J Hughes, Yichen Zhu, David J Wagg, Towards the development of an operational digital twin, Vibration 3 (3) (2020) 235–265.

[12] A. Gray, A. Wimbush, M. de Angelis, P.O. Hristov, D. Calleja, E. Miralles-Dolz, R. Rocchetta, From inference to design: A comprehensive framework for uncertainty quantification in engineering with limited information, Mech. Syst. Signal Process. 165 (2022) 108210.

[13] Edoardo Patelli, Silvia Tolo, Hindolo George-Williams, Jonathan Sadeghi, Roberto Rocchetta, Marco de Angelis, Matteo Broggi, Opencossan 2.0: an efficient computational toolbox for risk, reliability and resilience analysis, 2018.

[14] Michaël Baudin, Anne Dutfoy, Bertrand Iooss, Anne-Laure Popelin, Openturns: An industrial software for uncertainty quantification in simulation, in: Handbook of Uncertainty Quantification, Springer, 2017, pp. 2001–2038.

[15] Audrey Olivier, Dimitris G Giovanis, BS Aakash, Mohit Chauhan, Lohit Vandanapu, Michael D Shields, UQpy: A general purpose Python package and development environment for uncertainty quantification, J. Comput. Sci. 47 (2020) 101204.

[16] Eclipse Foundation, Eclipse ditto, 2021, https://Github.Com/Eclipse/Ditto.

[17] Miguel Grinberg, Flask Web Development: Developing Web Applications with Python, " O'Reilly Media, Inc.", 2018.

[18] I. Kåsa, A circle fitting procedure and its error analysis, IEEE Trans. Instrum. Meas. (1) (1976) 8–14.

[19] Ian D. Coope, Circle fitting by linear and nonlinear least squares, J. Optim. Theory Appl. 76 (2) (1993) 381–388.

[20] Ali Al-Sharadqah, Nikolai Chernov, Error analysis for circle fitting algorithms, Electron. J. Stat. 3 (2009) 886–911.
[21] Veikko Karimäki, Effective circle fitting for particle trajectories, Nucl. Instrum. Methods Phys. Res. A 305 (1) (1991) 187–191.
[22] Vijay D'silva, Daniel Kroening, Georg Weissenbacher, A survey of automated techniques for formal software verification, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 27 (7) (2008) 1165–1178.
[23] Dirk Beyer, M. Erkan Keremoglu, CPAchecker: A tool for configurable software verification, in: International Conference on Computer Aided Verification, Springer, 2011, pp. 184–190.
[24] Shiva Nejati, Next-generation software verification: an AI perspective, IEEE Softw. 38 (3) (2021) 126–130.
[25] Nick Gray, Marco de Angelis, Scott Ferson, Computing with uncertainty: introducing Puffin the automatic uncertainty compiler, in: Proceedings of the 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019), 2019.
[26] Nick Gray, Marco de Angelis, Scott Ferson, The creation of puffin, the automatic uncertainty compiler, 2021.