Computers and Structures 243 (2021) 106410

Contents lists available at ScienceDirect

Computers and Structures

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

Machine learning based digital twin for dynamical systems with multiple time-scales

S. Chakraborty [a,*], S. Adhikari [b]

[a] Department of Applied Mechanics, Indian Institute of Technology Delhi, Hauz Khas 110016, India
[b] College of Engineering, Swansea University, Swansea, UK

ARTICLE INFO

Article history:
Received 13 May 2020
Accepted 30 September 2020

Keywords:
Digital twin
Multi-timescale dynamics
Mixture of experts
Gaussian process
Frequency

ABSTRACT

Digital twin technology has a huge potential for widespread applications in different industrial sectors such as infrastructure, aerospace, and automotive. However, practical adoptions of this technology have been slower, mainly due to a lack of application-specific details. Here we focus on a digital twin framework for linear single-degree-of-freedom structural dynamic systems evolving in two different operational time scales in addition to its intrinsic dynamic time-scale. Our approach strategically separates into two components – (a) a physics-based nominal model for data processing and response predictions, and (b) a data-driven machine learning model for the time-evolution of the system parameters. The physics-based nominal model is system-specific and selected based on the problem under consideration. On the other hand, the data-driven machine learning model is generic. For tracking the multi-timescale evolution of the system parameters, we propose to exploit a mixture of experts as the data-driven model. Within the mixture of experts model, Gaussian Process (GP) is used as the expert model. The primary idea is to let each expert track the evolution of the system parameters at a single time-scale. For learning the hyperparameters of the 'mixture of experts using GP', an efficient framework that exploits expectation-maximization and sequential Monte Carlo sampler is used. Performance of the digital twin is illustrated on a multi-timescale dynamical system with stiffness and/or mass variations. The digital twin is found to be robust and yields reasonably accurate results. One exciting feature of the proposed digital twin is its capability to provide reasonable predictions at future time-steps. Aspects related to the data quality and data quantity are also investigated.

1. Introduction

Design and analysis of complex engineering systems using high fidelity computational simulations are an integral part of modern engineering practice. In the context of aerospace and mechanical engineering, computational simulations were historically employed to support conceptual design, prototyping, manufacturing, production, test-data correlation and safety assessment. Over the last decade, there has been a shift in taking advantage of computational simulations in providing service throughout the whole product life cycle [1,2], going well beyond the production stage. In the context of civil infrastructure, the idea of fusing digital information with real-life structures are also evolving at a significant pace [3]. The methodologies, algorithms, techniques, software and computer applications which mimic the evolution of a complex real system through computational and digital means are broadly termed as 'digital twins'. The key technologies and solutions that enabled the espousal of the technology include Artificial Intelligence (AI)/ Machine Learning (ML), and Internet-of-Things (IoT), among others. Factors, such as growing usage of connected devices across various organizations, increasing adoption of cloud platforms, and the emergence of high-speed networking technologies will fuel the growth of the digital twin technology.

By its very definition, digital twins are extremely diverse and can mean very different approaches to different applications. In this paper, we are interested in digital twins of structural dynamic systems, as many physical engineering systems can be expressed in this form. A digital twin is a virtualized proxy of a real physical dynamic system. While a numerical model of a physical system attempts to closely match the behaviour of a dynamic system, the digital twin also tracks the temporal evolution of the dynamic system. Once a digital twin has been trained and developed, it can be used to make crucial decisions at a point in time which is significantly far in the future from the time of manufacturing of

* Corresponding author.
E-mail addresses: souvik@am.iitd.ac.in (S. Chakraborty), S.Adhikari@swansea.ac.uk (S. Adhikari).

https://doi.org/10.1016/j.compstruc.2020.106410
0045-7949/© 2020 Elsevier Ltd. All rights reserved.

an engineering dynamic system. A general mathematical framework for digital twin has been suggested in [4]. Application specific development of digital has also been pursued by the researchers. For example, [5–9] investigated digital twin for prognostics and health monitoring, [10–14] developed digital twin for manufacturing and [15–18] developed digital twin for automotive and aerospace engineering. These references give an excellent idea of what can be achieved currently using the digital twin approach.

Philosophically, the development of digital twins can either be pursued from a physics-based modelling point of view or from a data-based modelling point of view. While physics-based approaches are generally more robust, there are often cases where physics of a given system is unknown or only partially known. Under such circumstances, it is not possible to employ a purely physics-based modelling approach. Naturally, data-based modelling approaches are the only viable alternative. The exact application of data-based modelling approaches is highly dependent on the underlying problem at hand. In case, the physics of the problem is completely unknown, the problem can be pursued from a purely data-driven point-of-view. On the other hand, if the physics of the problem is partially known, data-driven modelling techniques can be used to compensate for the missing physics. Historically, physics-based approaches have been mostly employed for the dynamic analysis of complex structures. However, with the development of sensor technology and wireless data transmission, the availability of useful data is set to increase in the future. Keeping this in mind, a hybrid physics-based and data-based approach for digital twin is pursued here.

Dynamic systems differ crucially from other systems due to the fact that their response due to external excitation change with time. The rate of change depends on their characteristics time period. This is a fundamental property of a dynamic system and mathematically it depends on the stiffness to mass ratio of the system. Typically, smaller structures have faster time periods and larger structures, for example, a large wind turbine, have slower time periods. However, irrespective of their characteristics time periods, the time-scale of their operational life is large. An example, for a large wind turbine, the time period is in the order of 10s of seconds [19], while its operation life is in 10s of years. To address this fundamental mismatch in the time-scales, the digital twin proposed in [20] explicitly considered two different time scales. The intrinsic timescale is a fast time scale, while the operational time scale is a slow time scale. Physics-based methods, such as the finite element method, are used for dynamic evolution in the intrinsic timescale, while data-based methods (e.g., surrogate models) are used for dynamic evolution in the operational timescale. The digital twin of a complex dynamic system will arise from the fusion of physics and data-based approaches, also known as the grey box modelling technique. The separation of computational approaches based on the two different time scales was exploited in [21] where Gaussian Process Emulators (GPE) was used in the slow time-scale.

One key aspect of the digital twin technology is to use the sensor data collected from the physical system to update the digital twin and then use the same for predicting the future states. In this regard, the role of machine learning (ML) algorithms become enormous. One of the reasons behind the recent thrust in digital twin technology is the development of advanced ML algorithms (e.g., deep neural network [22–25], Gaussian process [26–28]) that can be readily used to update the model and make future predictions. For example, [5] used two deep learning algorithms within the digital twin framework for prognosis and diagnosis of systems. Similarly, in [21], GPE was used for learning the evolution of the system parameters. Forecasting models such as autoregressive moving average [29] and autoregressive integrated moving average [30] can also be used. A detailed review of the impact of machine learning algorithms on the digital twin technology can be found in [31].

Although the separation of two time-scales provides a logical framework for developing computational and mathematical methods for the construction of digital twins, questions remain on how to define and propose the slow time scale. The fast time-scale is a fundamental property of dynamical systems and therefore is unambiguous for a given system. The same is not true for the slow operational time scale. In [20,21] the idea of a single time-scale for the evolution of the digital twin was used for its entire operational period. However, there is no physical or mathematical reason as to why this must be restricted to only one time-scale. It is perfectly possible that various factors in a complex digital twin evolve at different time scales. For example, the mass of a system can change due to corrosion, while the stiffness of a system can degrade due to fatigue. These two processes will have a very different time scale of evolution. Therefore, in a more general setting, a digital twin can evolve in different time scales in addition to its intrinsic time scale. The solution of such systems is difficult from a computational point-of-view as we need a time-step of the order of the fastest scale. As a result, tasks such as health-monitoring, damage prognosis and remaining useful-life prediction becomes excessively difficult. The difficulty increases exponentially for cases where the underlying physics governing the time-evolution is not known and data available is noisy. The key idea proposed and investigated in this paper is that the digital twin of a dynamic system evolves in two different operational time-scales. In principle, there can be more than two operational time-scales. Approaches proposed in the paper can form the basis of considering such problems.

The rest of the paper is organised as follows. In Section 2, the problem undertaken in this study is discussed. Details about the proposed digital twin framework for the multi-timescale dynamical system is discussed in Section 3. The performance of the proposed digital twin in capturing the (multiscale) temporal evolution of the system parameters is presented in Section 4. Finally, Section 5 presents the concluding remarks.

2. The problem statement

We consider a physical system that can be represented by a single degree of freedom (SDOF) spring mass and damper system.

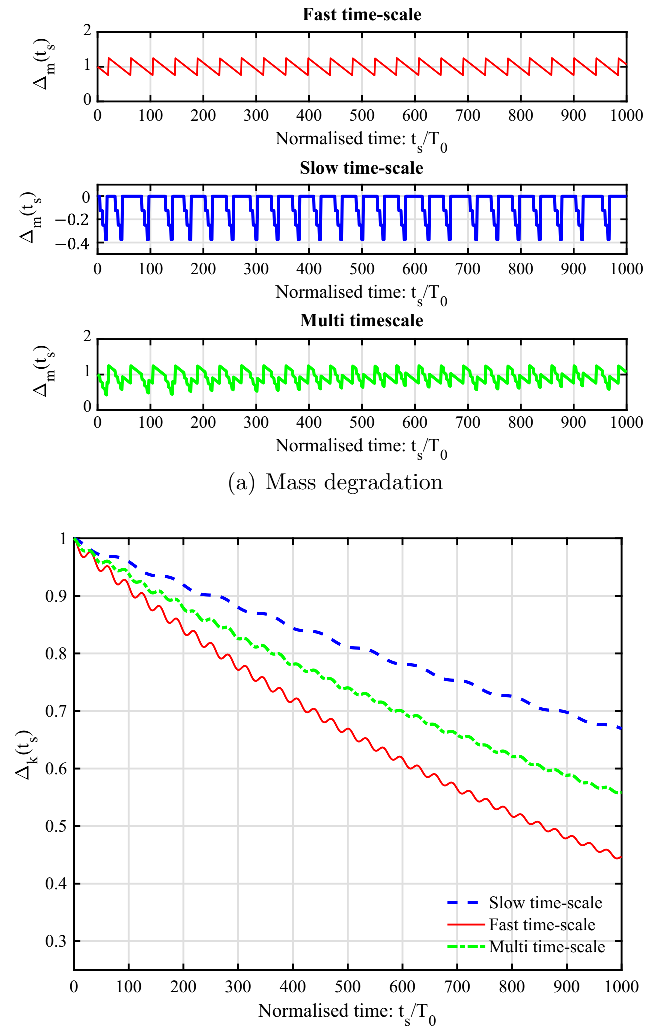
$$m_0 \frac{\mathrm{d}^2 u_0(t)}{\mathrm{d}t^2} + c_0 \frac{\mathrm{d}u_0(t)}{\mathrm{d}t} + k_0 u_0(t) = f_0(t), \tag{1}$$

where m_0, c_0 and k_0 are, respectively the mass, damping and stiffness of the system. Here t is the intrinsic time of the system. Equation (1) is often referred to as the 'nominal system' and m_0, c_0 and k_0 as the nominal mass, nominal damping and nominal stiffness, respectively. $f_0(t)$ and $u_0(t)$ are respectively the forcing function and the dynamic response of the nominal system. At this stage, it is worthwhile to mention that a more realistic infinite-dimensional system expressed by using partial differential equations can be discretized into finite-dimensional systems by using standard numerical techniques such as the Galerkin method. These discretized systems are often represented by SDOF systems (as in Eq. 1) using orthogonal transformations.

The nominal system discussed in Eq. 1 has fixed system paramaters m_0, c_0 and k_0. For a digital twin, however, the system parameters, namely mass, damping and stiffness, and the forcing function changes with the service time t_s. A generalized equation of motion of this system can be represented as

$$m(t_s) \frac{\partial^2 u(t, t_s)}{\partial t^2} + c(t_s) \frac{\partial u(t, t_s)}{\partial t} + k(t_s) u(t, t_s) = f(t, t_s). \tag{2}$$

It is to be noted that the service time t_s is much slower than the intrinsic time t. The nominal system discussed in Eq. 1 can be viewed as the initial model at $t_s = 0$. The service time t_s can repre-



(a) Mass degradation

(b) Stiffness degradation

Fig. 1. Multi-timescale mass and stiffness degradation functions. The multi-timescale degradation functions are obtained by combining the fast and the slow time-scales shown in each figure. T_0 is the natural time-period of the underlying un-damped system.

sent the number of cycles in a aircraft. From Eq. 2, we note that the mass $m(t_s)$, damping $c(t_s)$, stiffness $k(t_s)$ and $f(t,t_s)$ changes with the 'service time' t_s, for instance due to the degradation in the system during its service time. Eq. 2 represents the equation of motion of the digital twin. Note that when $t_s = 0$, Eq. 2 reduces to the nominal system represented in Eq. 1. It is evident that the digital twin is completely described by the functions $m(t_s), c(t_s)$ and $k(t_s)$. Therefore, for using the digital twin in practice, one needs to estimate the functions $m(t_s), c(t_s)$ and $k(t_s)$.

In recent studies, physics-based [20] and data-based approaches [21] for estimating the functions $m(t_s), c(t_s)$ and $k(t_s)$ have been developed. However, these studies have a number of limitations.

- The physics-based digital twin proposed in [20] is not sufficiently accurate when the sensor data is noisy.
- The data-based digital twin proposed in [21] only works for systems having a single operational time-scale. The approach is not applicable for multi-timescale dynamical systems [32,33].
- One of the objectives of digital twin is to predict the future response, so as to understand the behaviour of the physical twin in future. Unfortunately, neither the physics-based [20] nor the data based [21] digital twins previously proposed is capable of predicting the future responses.

The objective of this study is to develop an efficient framework for addressing some of the above-mentioned limitations. More specifically, we are interested in developing digital twins for multi-timescale dynamical systems. Unlike the digital twins developed in [20,21], the digital twin developed in this paper should also, be able to predict future responses.

For developing the digital twin, it is assumed that sensors are deployed on the physical system. Recent developments in the field of Internet of Things (IoT) has provided us with numerous new data collection technologies and this provides the necessary connectivity between the physical and digital twins. Using the sensors, measurements are taken intermittently at t_s. It is assumed that the functions $m(t_s), c(t_s)$ and $k(t_s)$ are so slow that the dynamics of the system in Eq. 2 is decoupled. In other words, m_s, c_s and k_s of the system is constant as far as the instantaneous dynamics of the system is concerned. Without loss of generality, we assume

$$k_s(t_s) = k_s\left(t^{(s)}, t^{(f)}\right) = k_0\left(1 + \Delta_{\tilde{k}}\left(t^{(s)}, t^{(f)}\right)\right) \tag{3}$$

where

$$1 + \Delta_{\tilde{k}}\left(t^{(s)}, t^{(f)}\right) = \Delta_k^{(s)}\left(t^{(s)}\right) + \Delta_k^{(f)}\left(t^{(f)}\right) = \Delta_k\left(t^{(s)}, t^{(f)}\right). \tag{4}$$

Here $t^{(s)}$ and $t^{(f)}$ represent a slower and a faster time scale of evolution of the respective processes. The 'multi-scale' nature of the problem arises from the fact that $t^{(s)}$ and $t^{(f)}$ can be significantly different from each other. Here we implicitly assume that the compound function $\Delta_k\left(t^{(s)}, t^{(f)}\right)$ is effectively expressed as a linear sum of two different functions with two separate time variables. This is done for mathematical simplicity. Although the function $\Delta_k\left(t^{(s)}, t^{(f)}\right)$ is expressed as a linear sum, the two functions themselves are highly nonlinear with respect to their arguments.Without any loss of generality, we express these two different time scales as a function of a single service time-scale t_s with different coefficients. Using this approach we have

$$\begin{aligned}\Delta_k(t_s) &= \Delta_k^{(s)}(t_s) + \Delta_k^{(f)}(t_s)\\ &= \underbrace{0.5e^{-\alpha_k^{(s)}t_s}\frac{\left(1+\epsilon_k^{(s)}\cos\left(\beta_k^{(s)}t_s\right)\right)}{\left(1+\epsilon_k^{(s)}\right)}}_{\Delta_k^{(s)}\left(t^{(s)}\right)} + \underbrace{0.5e^{-\alpha_k^{(f)}t_s}\frac{\left(1+\epsilon_k^{(f)}\cos\left(\beta_k^{(f)}t_s\right)\right)}{\left(1+\epsilon_k^{(f)}\right)}}_{\Delta_k^{(f)}\left(t^{(f)}\right)}.\end{aligned} \tag{5}$$

In Eq. 5, we have assumed that the stiffness degradation results from two different processes - one relatively slow and one relatively fast. In the context of aircraft, relatively fast degradation can be linked to areas which are subjected to repeated loading/unloading such as the landing gears and bulkheads in the fuselage. On the other hand, slow stiffness degradation are often related to environmental effects and/or maintenance issues such as corrosion and surface delamination. Numerical values considered for the stiffness degradation are: $\alpha_k^{(s)} = 0.4 \times 10^{-3}, \epsilon_k^{(s)} = 0.005, \beta_k^{(s)} = 7 \times 10^{-2}$, $\alpha_k^{(f)} = 0.8 \times 10^{-3}, \epsilon_k^{(f)} = 0.01$ and $\beta_k^{(f)} = 2 \times 10^{-1}$. Similarly, we also assume

$$m(t_s) = m_0(1 + \Delta_m(t_s)), \tag{6}$$

where

$$\Delta_m(t_s) = \Delta_m^{(s)}(t_s) + \Delta_m^{(f)}(t_s). \tag{7}$$

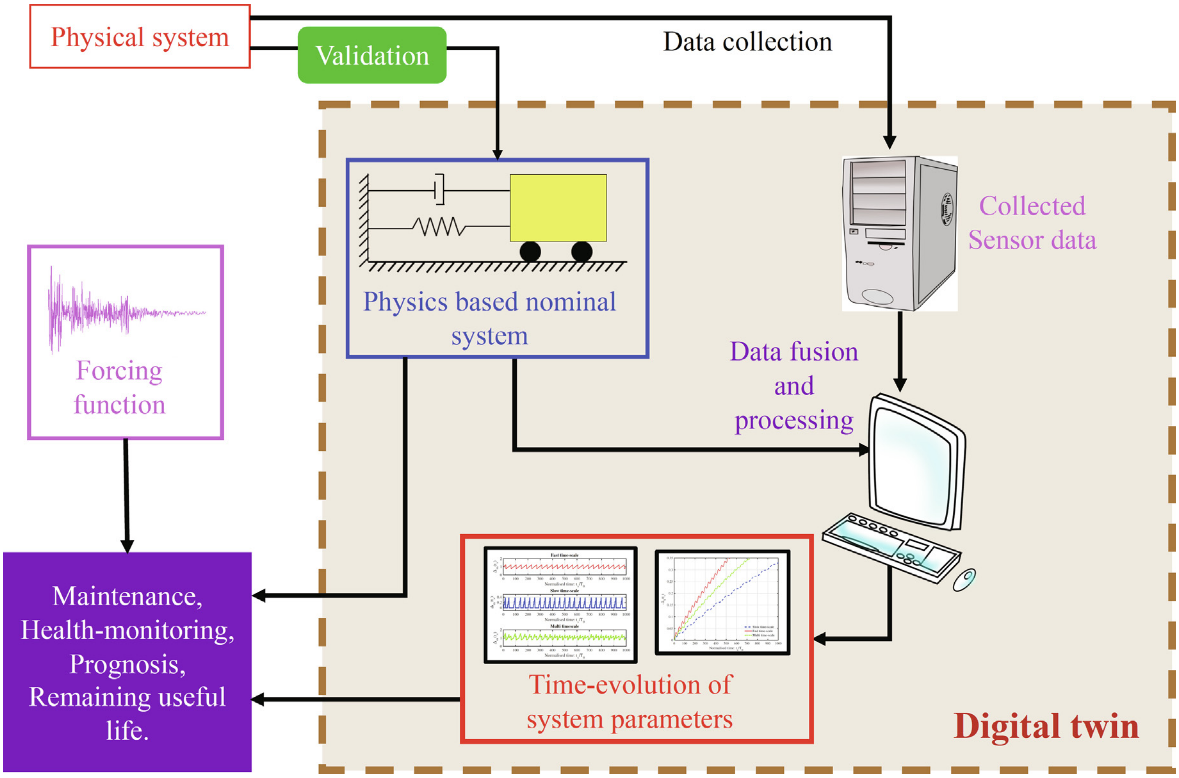

Physical system
Validation
Data collection
Physics based nominal system
Collected Sensor data
Forcing function
Data fusion and processing
Maintenance, Health-monitoring, Prognosis, Remaining useful life.
Time-evolution of system parameters
Digital twin

Fig. 2. Schematic representation of the digital twin. It has three primary building blocks, namely data fusion and processing, determining time evolution of the system parameters and making predictions using the digital twin. This digital twin can be used for several tasks including prognosis, health-monitoring, maintenance and remaining useful life prediction.

Similar to the stiffness degradation case, the mass degradation is also a function of two time-scales - the relatively slower time-scale $\Delta_m^{(s)}(t_s)$ and the relatively faster time-scale $\Delta_m^{(f)}(t_s)$. We have assumed,

$$\Delta_m^{(f)}(t_s) = \epsilon_m \,\text{SawTooth}(\beta_m(t_s - \pi/\beta_m)), \tag{8}$$

where $\beta_m = 0.15$ and $\epsilon_m = 0.25$. The slower time-scale is represented as

$$\Delta_m^{(s)}(t_s) = \begin{cases} m_1 & \text{if } t_1 \leqslant t_s < t_2 \\ m_2 & \text{if } t_2 \leqslant t_s < t_3 \\ m_3 & \text{if } t_3 \leqslant t_s < t_4 \\ 0 & \text{elsewhere} \end{cases}. \tag{9}$$

From a physical point-of-view, Eq. 8 can be associated with fuel loading and unloading of an aircraft. On the other hand, Eq. 9 can be associated with the case where the aircraft drops a bomb during its flight. Schematically, the mass and stiffness degradation are shown in Fig. 1. For the mass degradation function, the difference in the scale arises from the functional form and the time-discretization necessary for discretizing the process. For the stiffness degradation function, the difference in scale is due to periodicity of the functions. The damping is considered to be constant. The key consideration is that a digital twin of the dynamical system should be able to track these kinds of changes occurring at multiple-scales by exploiting sensor data measured on the system. Moreover, a digital twin should also be able to predict future degradation.

3. Digital twin for multi-timescale dynamical systems

In this section, we discuss the proposed digital twin framework for multi-timescale dynamical systems. A schematic representation of the framework is shown in Fig. 2. The framework proposed has two primary components - (a) data processing by using the physics of the problem (physics-based nominal model) and (b) Learning the time-evolution of system parameters by using machine learning (ML). Once the material degradation is known, the future responses can be predicted by combining the ML predicted material properties with the physics of the problem defined by the governing differential equation. To track the multi-timescale nature of the degradation functions, we propose to use the concept of mixture of experts (MOE) where each expert is employed to track a single time-scale. Based on the success of the Gaussian process (GP) in solving problems having single time-scale [21], we propose to use GP as the experts within the MOE framework. The overall framework is referred to as the mixture of experts using Gaussian process (ME-GP). We first present the details on data processing and then proceed to discuss the concept of the proposed ME-GP.

3.1. Data collection and processing

One major player in the development of the digital twin technology is the IoT. Advances in IoT have provided us with several new data collection technologies that, in turn, drives the development of the digital twin technology and enables connectivity between and physical and the digital twins. The overall idea of the digital twin technology is based on the idea of this connectivity. This connectivity is established by placing sensors on the physical twin to collect data and communicate it to the digital counterpart by using cloud technology. With advances in the sensor technologies, we now have different sensors for collecting different type of responses. In this work, we work with the natural frequency of the system. The advantage of using natural frequency resides in the fact that it is a scalar quantity and hence, we can avoid working with a big data-set. We assume that the frequency of the system can be measured in an online fashion. Available literature illustrates that this is feasible. In [34], a vision-based sensor capable of remotely measuring the structural response was proposed. The effectiveness of the proposed sensor was illustrated by conducting field tests on railway bridges in both time and frequency domains. A sensor that infers the natural frequency of a system from vibration induced strain was proposed in [35]. Applicability of this sensor was illustrated by conducting experiments on metal pipe under vibration and impact load. Electrical strain gauge, piezoelectric accelerometer and fibre Bragg gratting sensors were used for obtaining the natural frequency of the system. Both these studies show that measuring the natural frequency of a system is feasible, a fact that we use in our study.

In this paper, three different cases have been considered. In the first case, it is assumed that only the stiffness degrades. In the second case, we assume the stiffness to be constant; the variability in the observations is due to the variation in the mass. Lastly, in the third case, we assume that both mass and stiffness vary. The collected data needs to be processed differently for each of these three cases. Details on how the data is processed for each of these three cases are furnished below.

3.1.1. Stiffness degradation

We assume that the mass and damping of the nominal model in Eq. 2 are unchanged and only the stiffness degrades. Accordingly, the equation of motion for this case is written as

$$m_0\frac{\mathrm{d}^2u(t)}{\mathrm{d}t^2}+c_0\frac{\mathrm{d}u(t)}{\mathrm{d}t}+k_0(1+\Delta_k(t_s))u(t)=f(t). \tag{10}$$

where all the terms have similar notations as defined before. Note that Eq. 10 is a special case of Eq. 2 where t_s is fixed. Solving the characteristic equation, the damped natural frequency of the system can be represented as

$$\lambda_{s_{1,2}}(t_s)=-\zeta_0\omega_0\pm\mathrm{i}\omega_0\sqrt{1+\Delta_k(t_s)-\zeta_0^2}, \tag{11}$$

where ω_0 and ζ_0 are respectively the natural frequency and damping ratio of the system at $t_s=0$. Eq. 11 can be rearranged as

$$\lambda_{s_{1,2}}(t_s)=-\underbrace{\frac{\zeta_0}{\sqrt{1+\Delta_{\hat{k}}(t_s)}}}_{\zeta_s(t_s)}\times\underbrace{\omega_0\sqrt{1+\Delta_{\hat{k}}(t_s)}}_{\omega_s(t_s)}\pm\mathrm{i}\underbrace{\omega_0\sqrt{1+\Delta_{\hat{k}}(t_s)}\sqrt{1-\left(\frac{\zeta_0}{\sqrt{1+\Delta_{\hat{k}}(t_s)}}\right)^2}}_{\omega_{d_s}(t_s)}, \tag{12}$$

where $\omega_s(t_s)=\omega_0\sqrt{1+\Delta_{\hat{k}}(t_s)}$ i, $\zeta(t_s)=\zeta_0/\sqrt{1+\Delta_{\hat{k}}(t_s)}$ and $\omega_{d_s}(t_s)=\omega_s(t_s)\sqrt{1-\zeta_s^2(t_s)}$ represent the evolution of the natural frequency, damping ratio and damped natural frequency with t_s. As the natural frequency extraction techniques in literature generally extract the damped natural frequency, we have considered it to be the data available from the physical twin. It can be shown [20]

$$\Delta_{\hat{k}}(t_s)=-\tilde{d}_1(t_s)\left(2\sqrt{1-\zeta_0^2}-\tilde{d}_1(t_s)\right), \tag{13}$$

where

$$\tilde{d}_1(t_s)=\frac{d_1(\omega_{d_0},\omega_{d_s}(t_s))}{\omega_0}. \tag{14}$$

The function $d_1(\omega_{d_0},\omega_{d_s}(t_s))$ in Eq. 14 is the distance between ω_{d_0} and $\omega_{d_s}(t_s)$

$$d_1(\omega_{d_0},\omega_{d_s}(t_s))=\|\omega_{d_0}-\omega_{d_s}(t_s)\|_2. \tag{15}$$

Now given the fact that the initial damped frequency of the system, ω_{d_0} is known and we have sensor measurements for $\omega_{d_s}(t_s)$, one can easily compute $\tilde{d}_1(t_s)$ and $\Delta_{\hat{k}}(t_s)$ at t_s by using Eq. 14 and substituting it into Eq. 13. Note that the sensor measurements $\tilde{d}_1(t_s)$ are likely to be corrupted by noise and hence, the estimates for $\Delta_{\hat{k}}$ are also noisy. In this study, this noisy estimates, $\Delta_{\hat{k}}$ at discrete time t_s are used for developing the digital twin for the multi-timescale dynamical system.

3.1.2. Mass evolution

In this case, we consider that the stiffness and damping of the nominal model in Eq. 2 are constant, and the variation in the observed natural frequency is due to variation in the mass during the service life. Accordingly, the equation of motion of the physical system reduces to

$$m_0(1+\Delta_m(t_s))\frac{\mathrm{d}^2u(t)}{\mathrm{d}t^2}+c_0\frac{\mathrm{d}u(t)}{\mathrm{d}t}+k_0u(t)=f(t). \tag{16}$$

Again, Eq. 16 is a special case Eq. 2 where only m varies and the stiffness is constant. Solving for the damped natural eigenfrequencies as before

$$\lambda_{s_{1,2}}(t_s)=-\omega_s(t_s)\zeta_s(t_s)\pm\mathrm{i}\omega_{d_s}(t_s), \tag{17}$$

where

$$\omega_s(t_s)=\frac{\omega_0}{\sqrt{1+\Delta_{\hat{m}}(t_s)}}, \tag{18a}$$

$$\zeta_s(t_s)=\frac{\zeta_0}{\sqrt{1+\Delta_{\hat{m}}(t_s)}}\quad\text{and} \tag{18b}$$

$$\omega_{d_s}(t_s)=\omega_s(t_s)\sqrt{1-\zeta_s^2(t_s)}. \tag{18c}$$

are the evolution of natural frequency, damping ratio and damped natural frequency of the digital twin. Similar to the stiffness degradation case, we have

$$\Delta_{\hat{m}}(t_s)=\frac{-2\tilde{d}_2(t_s)^2+4\tilde{d}_2(t_s)\sqrt{1-\zeta_0^2}-1+2\zeta_0^2}{2\left(-\tilde{d}_2(t_s)+\sqrt{1-\zeta_0^2}\right)^2}+\frac{\sqrt{1-4\tilde{d}_2(t_s)^2\zeta_0^2+8\tilde{d}_2(t_s)\sqrt{1-\zeta_0^2}\zeta_0^2-4\zeta_0^2+4\zeta_0^4}}{2\left(-\tilde{d}_2(t_s)+\sqrt{1-\zeta_0^2}\right)^2} \tag{19}$$

where $\tilde{d}_2(t_s)$ is the equivalent of $\tilde{d}_1$ for the stiffness evolution case. Again we emphasize that the sensor based estimates of the damped natural frequencies are noisy and hence, the estimated $\Delta_{\hat{m}}(t_s)$ are also noisy. In this case, we utilize the noisy $\Delta_{\hat{m}}(t_s)$ at discrete time t_s for developing the digital twin for multi-timescale systems.

3.1.3. Mass and stiffness evolution

In this case, we consider the evolution of mass and degradation of stiffness, simultaneously. The equation of motion in this case is represented as

$$m_0(1+\Delta_m(t_s))\frac{\mathrm{d}^2u(t)}{\mathrm{d}t^2}+c_0\frac{\mathrm{d}u(t)}{\mathrm{d}t}+k_0(1+\Delta_k(t_s))u(t)=f(t). \tag{20}$$

All the notations in Eq. 20 have same meaning as before. The damped natural eigenfrequencies of this system can are represented as

$$\lambda_{s_{1,2}}=-\omega_s(t_s)\zeta_s(t_s)\pm\mathrm{i}\omega_{d_s}(t_s), \tag{21}$$

where

$$\omega_s(t_s)=\omega_0\frac{\sqrt{1+\Delta_{\hat{k}}(t_s)}}{\sqrt{1+\Delta_{\hat{m}}(t_s)}} \tag{22a}$$

$$\zeta_s(t_s) = \frac{\zeta_0}{\sqrt{1+\Delta_{\hat{m}}(t_s)}\sqrt{1+\Delta_{\hat{k}}(t_s)}} \quad \text{and} \tag{22b}$$

$$\omega_{d_s}(t_s) = \omega_s(t_s)\sqrt{1-\zeta_s^2}. \tag{22c}$$

ω_s, ζ_s and ω_{d_s}, respectively represent the evolution of natural frequency, damping ratio and damped natural frequency. Unlike the previous two cases, both $\Delta_m(t_s)$ and $\Delta_k(t_s)$ are unknowns in this case and hence, we need two equations to solve these unknowns. To that end, we consider the real and imaginary parts of Eq. 21 separately to derive the two equations necessary for estimating $\Delta_m(t_s)$ and $\Delta_k(t_s)$. With this setup, we arrive at the following expression [20]

$$\Delta_{\hat{m}}(t_s) = -\frac{\tilde{d}_{\mathcal{R}}(t_s)}{\zeta_0 + \tilde{d}_{\mathcal{R}}(t_s)}, \tag{23a}$$

$$\Delta_{\hat{k}}(t_s) = \frac{\zeta_o \tilde{d}_{\mathcal{R}}^2(t_s) - (1-2\zeta_0^2)\tilde{d}_{\mathcal{I}}(t_s) + \zeta_0^2 \tilde{d}_{\mathcal{I}}^2(t_s)}{\zeta_0 + \tilde{d}_{\mathcal{R}}(t_s)}, \tag{23b}$$

where $\tilde{d}_{\mathcal{R}}(t_s)$ and $\tilde{d}_{\mathcal{I}}(t_s)$, as before, are distance measures

$$\begin{aligned} \tilde{d}_{\mathcal{R}}(t_s) &= \frac{d_{\mathcal{R}}(t_s)}{1+\Delta_{\hat{m}}(t_s)}, \quad \tilde{d}_{\mathcal{I}}(t_s) \\ &= \sqrt{1-\zeta_0^2} - \frac{\sqrt{(1+\Delta_{\hat{k}}(t_s))(1+\Delta_{\hat{m}}(t_s)) - \zeta_0^2}}{1+\Delta_{\hat{m}}(t_s)}. \end{aligned} \tag{24}$$

Note that $\Delta_{\hat{m}}(t_s)$ and $\Delta_{\hat{k}}(t_s)$ are estimated from noisy observations of λ and hence, are noisy. In this case also, we will utilize $\Delta_{\hat{m}}(t_s)$and $\Delta_{\hat{k}}(t_s)$obtained at discrete time t_s within the digital twin framework for multi-timescale dynamical system.

3.2. Mixture of experts with Gaussian process

In the next part of the digital twin framework, we use the processed data to learn the evolution of the system parameters. Note that the evolution of the system parameters are of multi-timescale nature and hence, is difficult to learn. In this paper, we propose the use of mixture of experts (MoE) within the digital twin framework. MoE is used to learn the evolution of the system parameters. We argue that each of the experts within MoE learns the evolution at a single scale and hence, MoE can predict the evolution of the system parameters. As experts within the MoE framework, we propose the use of GP. The effectiveness of GP in predicting parameter evolution at a single scale has already been established in a previous study [21].

Suppose, we have a sequence of observations $\boldsymbol{y}_{t_s} \in \mathbb{R}^d$ at discrete time $t_s, s = 1, 2, \ldots, \tau$. For the digital twin problem in this study, $\boldsymbol{y}_{t_s}$ can either be $\Delta_k(t_s)$ and/or $\Delta m(t_s)$ at discrete time t_s. This observations are generated from some unknown process having multiple time-scales. We assume that the observations are generated from M hidden states $x_t^{(m)}, m = 1, 2, \ldots, M$, also referred to as experts. This hidden states are generally assumed to be independent and can evolve independent of each other. We here assume that the independent hidden states evolves according to a GP

$$x_m | t \sim \mathcal{GP}(\mu_m(t; \boldsymbol{h}), \kappa_m(t_1, t_2; \boldsymbol{l}), \quad m = 1, 2, \ldots, M, \tag{25}$$

where

$$\mu_m(t) = \boldsymbol{h}^T \boldsymbol{\phi}(t), \tag{26}$$

represents the mean of the GP. $\boldsymbol{h}$ in Eq. 25 represents the unknown coefficients and $\boldsymbol{\phi}(t)$ represents the basis function vector. κ_m in Eq. 25 represents the correlation function with length-scale parameter $\boldsymbol{l}$. $\boldsymbol{h}$ and $\boldsymbol{l}$ are referred to as the hyperparameters of the GP. Eq. 25 can be viewed as the prior, parameterized by the hyperparameters $\boldsymbol{h}$ and $\boldsymbol{l}$ in the space of the hidden state. These hidden states can be coupled in a generative manner to obtain the underlying data

$$y_{t_s} = y(t = t_s) = \sum_{m=1}^{M} z_m(t; \boldsymbol{\theta}^g) x_m(t; \boldsymbol{\theta}^e). \tag{27}$$

$z_m(t)$ is the m–th gating function and defined as

$$z_i(t) = \frac{\pi_i \mathcal{N}(t|\mu_i, \lambda_i^{-1})}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(t|\mu_j, \lambda_j^{-1}\right)}, \quad \sum_{j=1}^{M} \pi_j = 1. \tag{28}$$

$\boldsymbol{\theta}^g = \left[\mu_j, \lambda_j\right]_{j=1}^{M}$ are the hyperparameters of the gating function. $\pi_j, j = 1, 2, \ldots, M$ in Eq. 28 represents the mixing coefficient. x_m in Eq. 27 is the m-th GP expert. $\boldsymbol{\theta}^e = [\boldsymbol{h}_m, \boldsymbol{l}_m]_{m=1}^{M}$ is the hyperparameter associated with the expert function.

For using the model defined in Eqs. (25)–(28), all the hyperparameters need to be estimated based on the training data $\boldsymbol{\mathcal{D}} = [\boldsymbol{y}_{t_s}, t_s]$. One way to achieve this is by maximizing the data likelihood of the model.

$$p(y_{t_s}|t_s, \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{M} p(i|t_s, \boldsymbol{\theta}^g, \boldsymbol{\pi}) p(y_{t_s}|t_s, \boldsymbol{\theta}^e,) \tag{29}$$

$p(i|t_s, \boldsymbol{\theta}^g, \boldsymbol{\pi})$ in Eq. 29 is the posterior conditional probability, where $t_s$is assigned to the partition corresponding to the i–th expert, i.e.,

$$p(i|t_s, \boldsymbol{\theta}^g, \boldsymbol{\pi}) = z_i(t). \tag{30}$$

$p(\boldsymbol{y}_{t_s}|t_s, \boldsymbol{\theta}^e)$, on the other hand, is the probability distribution of the i–th expert and hence is a GP

$$p(y_{t_s}|t_s, \boldsymbol{\theta}^e,) = \mathcal{N}(\mu_i(t_s; \boldsymbol{h}), \kappa_i(t_{s,1}, t_{s,2}; \boldsymbol{l})). \tag{31}$$

Substituting Eqs. (28) and (31) into Eq. 29, we obtain

$$p(y_{t_s}|t_s, \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{M} \frac{\pi_i \mathcal{N}(t|\mu_i, \lambda_i^{-1})}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(t|\mu_j, \lambda_j^{-1}\right)} \mathcal{N}(y_{t_s}|\mu_i(t_s; \boldsymbol{h}), \kappa_i(t_{s,1}, t_{s,2}; \boldsymbol{l})). \tag{32}$$

Note that Eq. 32 is analytically intractable. Using the training samples $\boldsymbol{y}_{t_s}$ and $\boldsymbol{t}_s, s = 1, 2, \ldots, \tau$, the likelihood can be represented as

$$p(\boldsymbol{y}_{t_s}|\boldsymbol{t}_s, \boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{s=1}^{\tau} \sum_{i=1}^{M} \frac{\pi_i \mathcal{N}(t_s|\mu_i, \lambda_i^{-1})}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(t_s|\mu_j, \lambda_j^{-1}\right)} \mathcal{N}(y_{t,s}|\mu_i(t_s; \boldsymbol{h}), \kappa_i(t_{s,1}, t_{s,2}; \boldsymbol{l})). \tag{33}$$

One way to estimate the parameters in Eq. 33 is by using maximum likelihood estimator where we maximize the likelihood in Eq. 33. However, such an approach often leads to over-fitting. An alternative to the maximum likelihood estimator is to using an Bayesian approach and compute the posterior distribution of the hyperparameters. However, as the likelihood is intractable for the problem in hand, such an approach is computationally expensive. In this paper, with adopt a hybrid approach where some of the parameters are treated in a Bayesian way while for the other parameters, point estimates are computed. More specifically, within the proposed framework, we compute point-estimates for the mixing coefficients $\boldsymbol{\pi}$. The hyperparameters corresponding to the gating distribution and the experts are treated in a Bayesian way.

To estimate the hyperparameters using the proposed hybrid approach, we first use Bayes rule to compute the posterior distribution of the hyperparameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$

$$p(\boldsymbol{\theta}, \boldsymbol{\pi}|\boldsymbol{y}_s, \boldsymbol{t}_s) = \frac{p(\boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{y}_s|\boldsymbol{t}_s, \boldsymbol{\pi}, \boldsymbol{\pi})}{p(\boldsymbol{y}_s|\boldsymbol{t}_s)}, \tag{34}$$

where $p(\boldsymbol{\pi}, \boldsymbol{\theta})$represents prior distribution of the hyperparameters and $p(\boldsymbol{y}_s | \boldsymbol{t}_s, \boldsymbol{\pi}, \boldsymbol{\pi})$ is obtained from Eq. 33. Recall that the goal is to compute point-estimates for the mixing parameters $\boldsymbol{\pi}$. This can be achieved by maximizing the log-posterior for the mixing coefficients

$$\mathcal{L}(\boldsymbol{\pi}) = \log p(\boldsymbol{\pi} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) = \log \int p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) \mathrm{d}\boldsymbol{\theta}. \tag{35}$$

Unfortunately, this is not straightforward as Eq. 35 involves integration over the unknown $\boldsymbol{\theta}$. In this work, we propose to use expectation maximization for computing the mixing coefficients by maximizing the log-posterior in Eq. 35. In expectation maximization, we iterate over a series of increasing lower-bound of $\mathcal{L}(\boldsymbol{\pi})$ by using the Jensen's inequality

$$\begin{aligned}\mathcal{L}(\boldsymbol{\pi}) =) = \log p(\boldsymbol{\pi} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) \quad &= \log \int p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) \mathrm{d}\boldsymbol{\theta} \\ &= \log \int q(\boldsymbol{\theta}) \frac{p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} \\ &\geqslant \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} \\ &= F(q, \boldsymbol{\pi}),\end{aligned} \tag{36}$$

where $q(\boldsymbol{\theta})$ is an auxiliary distribution. It is obvious that the equality in Eq. 36 holds when $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta} | \boldsymbol{\pi}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)$. Using expectation maximization, $\boldsymbol{\pi}$ is estimated by iterating over the E-step (expectation step) and the M-step (maximization step).

- **E-step**: Given an estimate of $\boldsymbol{\pi} = \boldsymbol{\pi}^{(s)}$ in step s, we compute the lower-bound

$$\begin{aligned}F(q^{(s)}, \boldsymbol{\pi}) = \quad & \int q^{(s)}(\boldsymbol{\theta}) \log p(\boldsymbol{\pi}.\boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) \mathrm{d}\boldsymbol{\theta} \\ & - \int q^{(s)}(\boldsymbol{\theta}) \log \int q^{(s)}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}.\end{aligned} \tag{37}$$

- **M-step**: Maximize $F(q^{(s)}, \boldsymbol{\pi})$ to update $\boldsymbol{\pi}$.

$$\begin{aligned}\boldsymbol{\theta}^{(s+1)} \quad &= \arg\max_{\boldsymbol{\theta}} F(q^{(s)}, \boldsymbol{\pi}) \\ &= \arg\max_{\boldsymbol{\theta}} \left[\mathbb{E}_{q^{(s)}(\boldsymbol{\theta})} (\log p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)) \right].\end{aligned} \tag{38}$$

The second equality in Eq. 38 holds because the second term of $F(q^{(s)}, \boldsymbol{\pi})$ is independent of $\boldsymbol{\pi}$. It is important to note that the optimal distribution $q^{(s)}(\boldsymbol{\theta}) = p(\boldsymbol{\theta} | \pi^{(s)}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)$ is intractable. We propose to use sequential Monte Carlo (SMC) sampler [36] to generate samples from $p(\boldsymbol{\theta} | \pi^{(s)}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)$ so that the expectation in the E-step can be represented as

$$\mathbb{E}_{q^{(s)}(\boldsymbol{\theta})} (\log p(\boldsymbol{\pi}, \boldsymbol{\theta} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)) \approx \sum_{i=1}^{N_s} W^{(s,i)} \log p\left(\boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}^{(s,i)} | \boldsymbol{y}_{t_s}, \boldsymbol{t}_s\right), \tag{39}$$

where $\boldsymbol{\theta}^{(s,i)}$ is the i–th sample generated from $p(\boldsymbol{\theta} | \pi^{(s)}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)$, and $W^{(s,i)}$ is the corresponding weight.

Often posterior distributions are multi-modal and conventional Markov Chain Monte Carlo (MCMC) [37] may get trapped in a local mode. This results in long mixing time making the process inefficient. One algorithm that addresses this issue is the SMC sampler [36,38]. SMC provides a parallelizable framework for efficiently drawing samples from multi-modal posterior distributions. The idea of annealing is introduced to construct auxiliary distributions. We traverse from the prior to the posterior through this auxiliary distributions; this ensures a smooth transition from the tractable prior to the intractable posterior. It can be shown that samples drawn using SMC converges asymptotically to the target distribution [36].

For using SMC to approximate the E-step of the expectation maximization algorithm, we first express $p(\boldsymbol{\theta} | \pi^{(s)}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s)$ as

$$p(\boldsymbol{\theta} | \pi^{(s)}, \boldsymbol{y}_{t_s}, \boldsymbol{t}_s) \propto p(\boldsymbol{\theta}) p(\boldsymbol{y}_{t_s} | \boldsymbol{t}_s.\boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}). \tag{40}$$

where $p(\boldsymbol{\theta})$ is the prior and $p(\boldsymbol{y}_{t_s} | \boldsymbol{t}_s.\boldsymbol{\pi}^{(s)}, \boldsymbol{\theta})$ is the likelihood of the model defined in Eq. 33. In this work, we set the prior as a multivariate Gaussian distribution with zero mean and identity covariance matrix. Therefore, the parameters $\boldsymbol{\theta}$are independent in the prior. For ease of representation, we write the likelihood in a compress form as $p(\boldsymbol{\mathcal{D}} | \boldsymbol{\theta})$ and the posterior of $\boldsymbol{\theta}$ as $p_n(\boldsymbol{\theta})$. With these notations, Eq. 40 is represented as

$$p_n(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) p(\boldsymbol{\mathcal{D}} | \boldsymbol{\theta}) \tag{41}$$

Based on Eq. 41, we formulate the following auxiliary distribution in SMC

$$p_t(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) p^{\gamma_t}(\boldsymbol{\mathcal{D}} | \boldsymbol{\theta}), \tag{42}$$

where $t = 0, 1, \ldots, n$ and $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_n = 1$are the annealing parameters. Using the SMC sampler, samples are drawn from such a sequence of probability distribution by utilizing importance sampling and re-sampling. At step t, the idea is to generate a sufficient collection of $\left\{\boldsymbol{\theta}_r^{(i)}, \boldsymbol{w}_r^{(i)}\right\}, \ \ i = 1, \ldots, N_s$ such that the empirical distribution converges asymptotically to the target distribution $p_r(\boldsymbol{\theta})$. Sampling at $t = 0$ is trivial (as we sample the prior). From $t = 1$ onward, we employ importance sampling sequentially to the auxiliary distributions. A predefined Markov transition kernel is used to that end. Assuming, at step $t - 1, N_s$ samples $\left\{\theta_{t-1}^{(i)}\right\}, i = 1, \ldots, N_s$ are generated according to the proposal distribution φ_{t-1}, a kernel K_t with invariant distribution p_t is proposed such that the new samples are marginally distributed as [39]

$$\varphi_t = \int \varphi_{t-1} K_t(\boldsymbol{\theta}, \boldsymbol{\theta}\prime) \mathrm{d}\boldsymbol{\theta}. \tag{43}$$

Following [36], we have utilized the Metropolis-Hasting kernel with invariant distribution p_t to move the samples based on a random walk proposal

$$\varphi_t = \boldsymbol{\mathcal{N}}\left(\boldsymbol{\theta}_{r-1}^{(i)}, \mathbf{v}^{(i)}\right), \tag{44}$$

where $\mathbf{v}^{(i)}$ is the covariance matrix. To represent the discrepancy between the proposal distribution φ_t and the target distribution p_t at step $t, 0 < t \leqslant n$, unnormalized importance weights $w_t^{(i)}$ are generated.

$$w_t^{(i)} = w_t^{(i-1)} \frac{p_t\left(\boldsymbol{\theta}_{t-1}^i\right)}{p_{t-1}\left(\boldsymbol{\theta}_{t-1}^i\right)} \tag{45}$$

The computed weights are normalized as

$$W_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N_s} w_t^{(j)}} \tag{46}$$

As pointed out in [36,40], the SMC sampler degenerates and the variance of the importance weight increases. In this work, we measure the degeneracy based on the effective sample size (ESS) [39]

$$\mathrm{ESS}_t = \left(\sum_{t=1}^{N_s} \left(W_t^{(i)}\right)^2\right)^{-1}. \tag{47}$$

We consider degeneracy to have occurred if

$$\mathrm{ESS}_t < \mathrm{ESS}_{\min}, \tag{48}$$

where $\mathrm{ESS}_{\min}$ represents the threshold. In this work, we have defined $\mathrm{ESS}_{\min} = c \times N_s (c < 1)$. In case, $\mathrm{ESS}_t < \mathrm{ESS}_{\min}$, resampling is carried out to relieve the degeneracy of the sampler. Once samples corresponding to the target distribution are obtained, we utilize them to compute the expectation in the E-step of the expectation maximization algorithm. The steps involved in the SMC sampler are shown in Algorithm 1. The steps involved in training the proposed mixture of experts using GP algorithm are shown in Algorithm 2

Algorithm 1. Sequential Monte Carlo sampler

```
Input: Number of samples to generate $N_s$, the prior distribution $p(\boldsymbol{\theta})$, the
  number of steps $n$ and the threshold parameters $c$
Initialize $N_s$ particles $\boldsymbol{\theta}_0^{(i)}$, $i = 1, \ldots, N_s$ by directly sampling the prior
  distribution $p(\boldsymbol{\theta})$ and set the corresponds weights to be one, $w_0^{(i)} = 1$.
for $t = 1, \ldots, n$ do
    for $i = 1, \ldots, N_s$ do
        Sample $\boldsymbol{u}_i$ from uniform distribtuion $\mathcal{U}(\mathbf{0}, \mathbf{I})$.
        Sample $\tilde{\boldsymbol{\theta}}$ from the proposal distribution $\mathcal{N}\left(\boldsymbol{\theta}_{t-1}^{(i)}, \mathbf{v}_i\right)$.
        if $\boldsymbol{u}_i < \min\left\{\frac{p_t(\tilde{\boldsymbol{\theta}})}{p_t\left(\boldsymbol{\theta}_{t-1}^{(i)}\right)}\right\}$ then
            $\boldsymbol{\theta}_t^{(i)} \leftarrow \tilde{\boldsymbol{\theta}}$
        else
            $\boldsymbol{\theta}_t^{(i)} \leftarrow \boldsymbol{\theta}_{t-1}^{(i)}$
        end
    end
    Set weights of each particle according to Eqs. (45) and (46).
    Compute $\text{ESS}_t$ using Eq. (47).
    Resample if $\text{ESS}_t < \text{ESS}_{\text{min}}$.
end
Use $\boldsymbol{\theta}_n^{(i)}$ and $W_n^{(i)}$, $i = 1, \ldots, N_s$ to compute the expectation in the E-step of the
  expectation maximization algorithm.
```

Algorithm 2. Mixture of experts using Gaussian process

```
Input: Number of experts $M$, the training data $\mathcal{D} = [\boldsymbol{y}_{t_s}, \boldsymbol{t}_s]$, $s = 1, \ldots, \tau$, initial
  values of mixing coefficients $\boldsymbol{\pi}^{(i)}$ and threshold $\epsilon$.
$\boldsymbol{\pi} \leftarrow \boldsymbol{\pi}^{(i)}$.
$\lambda = 10\epsilon$.
repeat
    $\boldsymbol{\pi}_s \leftarrow \boldsymbol{\pi}$.
    Compute $F\left(q^{(s)}, \boldsymbol{\pi}\right)$ using SMC sampler (Algorithm 1).
    Update $\boldsymbol{\pi}$ by solving the optimization problem in Eq. (38).
    Compute error threshold
        $$\lambda = ||\boldsymbol{\pi} - \boldsymbol{\pi}_s||_2$$
until $\lambda \leq \epsilon$;
Outcome: Optimized $\boldsymbol{\pi}$, $N_s$ samples and corresponding weights from the
  posterior of $\boldsymbol{\theta}$ $\boldsymbol{\theta}_n^{(i)}$, $W_n^{(i)}$, $i = 1, \ldots, N_s$.
```

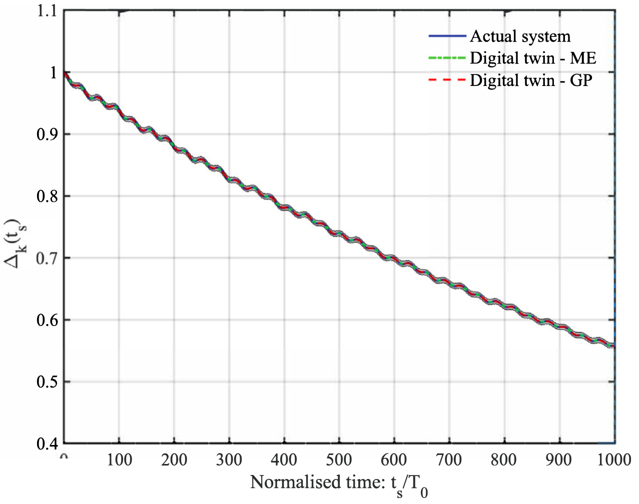


Fig. 3. Results obtained using GP based and ME-GP based digital twin for the case where only stiffness changes. Ideal scenario is considered where clean data throughout the service life is available.

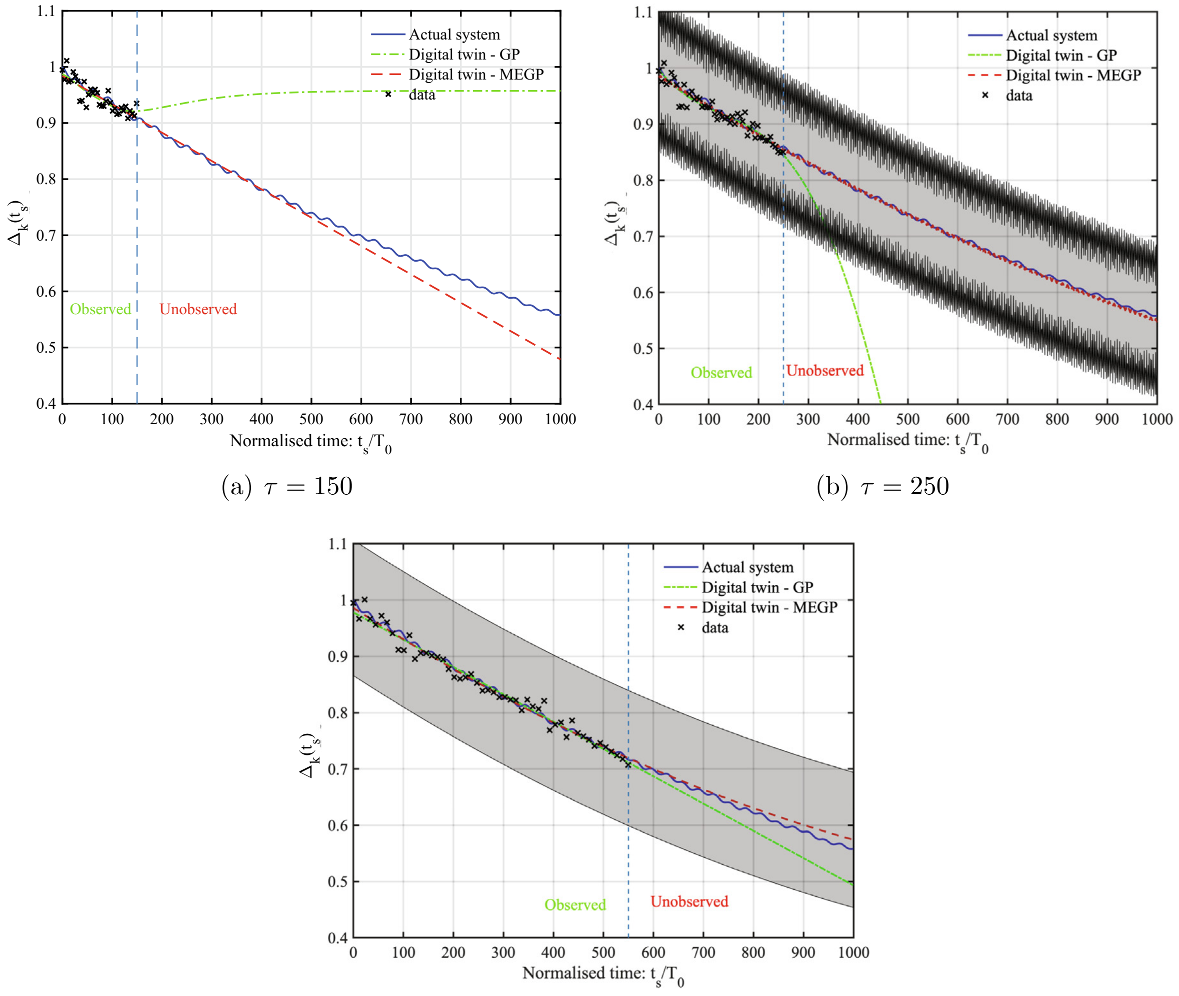
Once the hyperparameters of the proposed model are predicted by using the proposed approach, we proceed to make predictions using the proposed approach. Since we use a partially Bayesian approach to obtain the hyperparameters $\boldsymbol{\theta}$, it is possible to utilize the same to make *probabilistic* predictions. Suppose, we are interested in obtaining y_{t^*} at time-step t^*. This can be obtained by computing the posterior predictive distribution.

$$p(y_{t^*}|t^*,\boldsymbol{\mathcal{D}},\pi^*) = \int_{\theta} \sum_{i=1}^{M} \underbrace{p(i|t^*,\pi_i^*,\theta_i^g)}_{\text{gating}} \underbrace{p(y^*|t^*,\theta_i^e)}_{\text{expert}} \underbrace{p(\theta_i^e,\theta_i^g|\boldsymbol{\mathcal{D}})}_{\text{posterior}} \mathrm{d}\boldsymbol{\theta}. \qquad (49)$$

The integral above can be approximated by using Monte Carlo integration. In particular, we use the samples drawn from the posterior along with the correspond weights and the EM estimate of $\boldsymbol{\pi}^*$ to draw samples from the posterior predictive distribution in Eq. 49.

3.3. Algorithm

We now proceed to discuss how the components discussed in Sections 3.1 and 3.2 interacts with each other within the digital twin framework shown in Fig. 2, and how the digital twin enhanced with ME-GP can be used for multi-timescale dynamical systems. Given a physical system, the first step towards developing a digital twin is to develop a physics-driven nominal model for the system. For the current work, the nominal model is represented by Eq. 2. Next, the collected responses (damped natural frequencies of the system) are processed by using the procedure discussed in Section 3.1. To be more specific, we process the collected damped natural frequencies to obtain change in the mass, ($\Delta_m(t_s)$) and stiffness

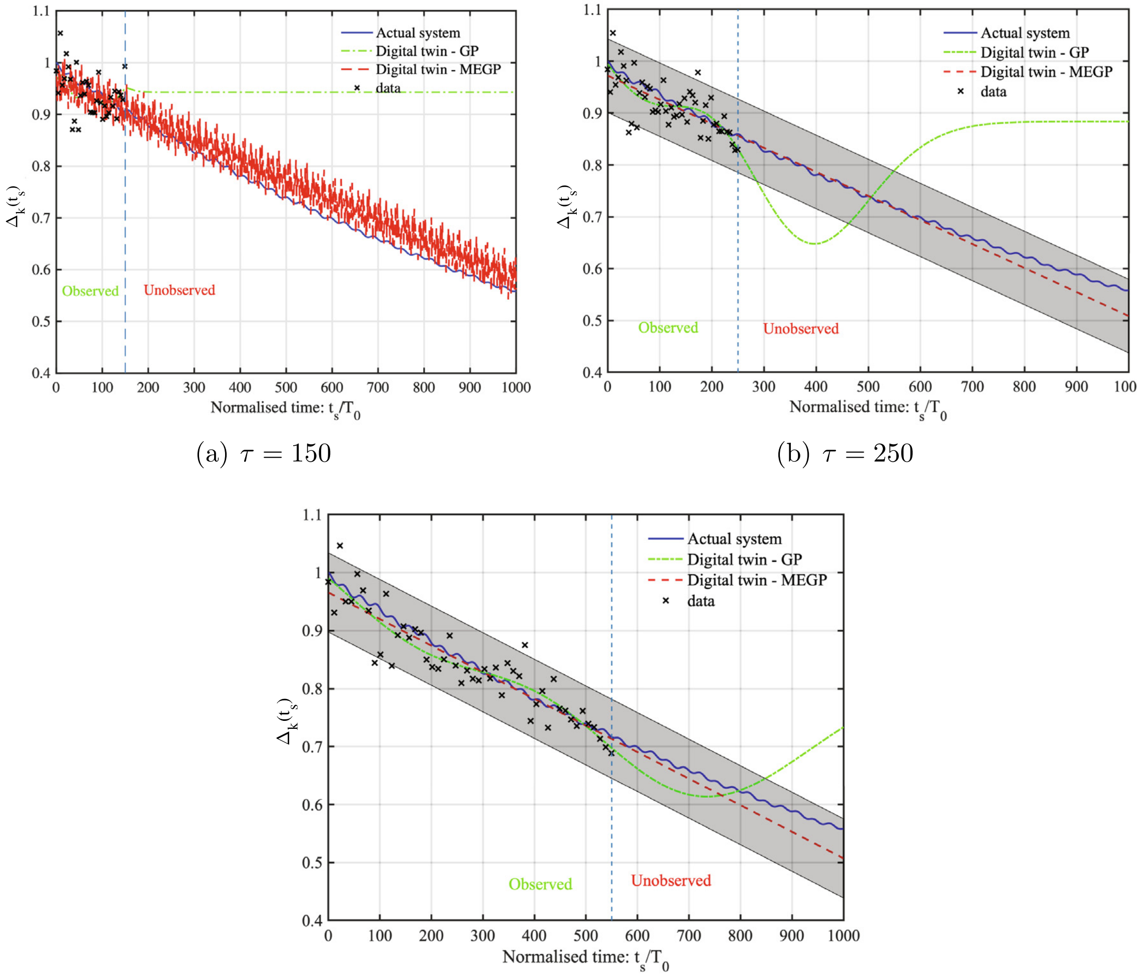


(a) $\tau = 150$

(b) $\tau = 250$

(c) $\tau = 550$

Fig. 4. Results obtained using GP and ME-GP based digital twin with observations up to $\tau = [150, 250, 550]$ and noise variance $\sigma_0 = 0.005$. For $\tau = 150$, we have 35 sensor data whereas for $\tau = [250, 550]$, we have 50 sensor data (shown using cross). The observed and unobserved regime are differentiated by using a vertical line.



(a) $\tau = 150$

(b) $\tau = 250$

(c) $\tau = 550$

Fig. 5. Results obtained using GP and ME-GP based digital twin with observations up to $\tau = [150, 250, 550]$ and noise variance $\sigma_0 = 0.015$. For observation up to 150, we have 35 sensor data whereas for observation up to 250 and 550, we have 50 sensor data (shown using cross). The observed and unobserved regime are differentiated by using a vertical line.

($\Delta_k(t_s)$) of the system. In the third step, the time-evolution of mass and stiffness, $\eta : t \rightarrow \Delta_k, \Delta_m$ is learned by using ME-GP. Finally, Using the trained ME-GP, we compute the future mass and stiffness, substitute them into the nominal model and solve it to obtain the future responses of interest. These future responses of interest can be used for health-monitoring, computing remaining useful life, devising a maintenance strategy and identifying defects and/or cracks in the system. How the algorithm within digital twin works is shown in Algorithm 3.

Algorithm 3. Proposed digital twin

```
1 Input: Nominal model and damped natural frequency of the
    physical system at different time-instants,
    $\mathcal{D} = [\lambda_s, t_s]$, $s = 1, \ldots, \tau$.
2 Process the collected data to obtain $\Delta_k(t_s)$ and/or $\Delta_m(t_s)$ at $t_s$
    (See Section 3.1).
3 Use ME-GP to learn the time-evolution of $\Delta_m$ and/or $\Delta_k$ (See
    Section 3.2).
4 Obtain $\Delta_k(t^*)$ and/or $\Delta_m(t^*)$ at $t^*, t^* > \tau$ (See Section 3.2).
5 Substitute $k^* = (1 + \Delta_k(t^*))$ and/or $m^* = (1 + \Delta_m(t^*))$ into
    the nominal model and solve it to obtain responses
    expected in the future.
6 Take engineering decision.
7 Repeat steps (2) – (6) as more data becomes available
```

The proposed digital twin framework has multiple advantages.

- The framework proposed utilizes both physics-driven model (ordinary and partial differential equations) and data-driven models (ME-GP). The physics-driven model ensures extrapolatibility of the proposed digital twin. On the other hand, the data-driven model ensures that the proposed digital twin is not limited by the facts that there may be missing physics.
- Including physics-based model also enables us to predict other responses of interest. For example, although we only have sensor information about the damped natural frequency of the system, the proposed digital twin can easily predict other responses such as strains, displacements and velocity.
- The fact that we utilize ME-GP enables the digital twin to track even multi-timescale dynamical systems such as the one considered in this paper.

4. Illustration of the proposed framework

In this section, we illustrate the performance, utility and applicability of the proposed digital twin framework for multi-timescale dynamical systems. More specifically, we present results for the problem defined in Section 2. Three cases as defined in Section 3 are considered. As already stated, it is assumed that we have access to the damped natural frequency of the system at different (slow) time-steps and the objective is to learn the time-evolution of the mass and/or the stiffness. Once the time-evolution of the mass and/or stiffness is known, the same can be used to determine the response of interest at a given time-instant by solving the physics-driven nominal model. We illustrate how the proposed digital twin can be used to learn the parameters (mass and stiffness) in the past (interpolation) as well as in the future (extrapolation). Lastly, to illustrate the difficulty posed by a multi-timescale dynamical system and to showcase the necessity of the ME-GP, we compare the results obtained using the proposed digital twin with those obtained using a GP based digital twin [21].

4.1. Digital twin via stiffness evolution

First, we consider the case where the variation in the collected natural frequency is due to the degradation in the stiffness of the system. The evolution of stiffness is assumed to be of multi-timescale nature as shown in Fig. 1 (b) (the multi-timescale one). However, neither the pattern of the time-evolution nor the number of scales present in the data is known to us a priori. Without loss of generality, the sensor data is assumed to be transmitted intermittently at a certain regular time-interval. To simulate a realistic scenario, the damped natural frequency data is contaminated with white Gaussian noise having a standard deviation σ_0. It is to be noted that the frequency of data availability depends on a number of factors including the bandwidth of the transmission system and cost of data collection. Therefore, we as engineers are not only interested in the behavior of the system at a future time, but also in the behaviour at an intermediate time.

Consider that we have N_s observations of the damped natural frequency $\lambda_s(t_s)$ equally spaced in time $t_s \in [0, \tau]$. Within the proposed digital twin framework, $\lambda_s(t_s)$ is processed in accordance to the procedure described in Section 3.1.1 to obtain the change in the stiffness $\Delta_{\tilde{k}}(t_s)$. After that, using t_s as the input data and $\Delta_{\tilde{k}}(t_s)$ as the output data, the ME-GP model is trained by using SMC and expectation maximization by using Algorithms 1 and 2. Since the number of scales present in the data is supposed to be unknown for a real-life problem, we have used four experts (different from the two scales that are actually present) within the ME-GP framework. Automatic relevance determination based Matern covariance function and quadratic mean function is considered for all the experts. The threshold parameter C is considered to be 0.85 [33,32] and we generate 1000 samples by using the SMC sampler. The trained ME-GP model is used as a surrogate to the unknown degradation process. The stiffness at a given time t^* can be calculated by using Eq. 3 where $\Delta_k(t_s) \approx \Delta_{\tilde{k}}(t^*)$ is obtained by using the trained ME-GP model. By substituting the estimated stiffness into the nominal model in Eq. 1, it is possible to predict any response of interest at t^*. Also as more data becomes available, the ME-GP model gets updated.

Fig. 3 shows the variation of Δ_k with normalized time. we have 200 measurements equally spaced throughout the service life of the system. The measured data is clean (i.e., no noise). We see that for this case, the proposed digital twin and the GP based digital twin yields identical results. In other words, in the presence of enough (clean) data throughout the service life of the system, the proposed ME-GP based digital twin reduces to a simple GP based digital twin. However, in a real-life setting, seldom do we have access to life-time data. Also, the data obtained is almost always corrupted by some form of noise.

Next, more realistic cases are considered. More specifically, we consider cases where we only have measurements in a certain observation time-window, $[0, \tau]$ and we are interested in predicting the evolution of stiffness at a time t^* where $t^* > \tau$. Moreover the data collected from the physical system is considered to be noisy. Fig. 4 shows the performance of digital twin for cases where $\tau = [150, 250, 550]$. For $\tau = 150$, we have 35 measurement data whereas for the other two cases, we have 50 sensor measurements. For all the three cases, the sensor measurements are contaminated by white Gaussian noise with standard deviation $\sigma_0 = 0.005$. The observed and unobserved regime are also marked in the figure by using a vertical line. Similar to Fig. 3, results have been generated by using the proposed digital twin and the GP based digital twin. For $\tau = 150$ (Fig. 4(a)), the proposed ME-GP based digital twin is found to yield excellent result up to $t_s/T_0 \approx 600$, which is almost four times the observation time window. Even beyond $t_s/T_0 = 600$, the results obtained from the proposed digital twin is found to be satisfactory. For clarity, the predictive uncertainty for this case is not reported. The GP based digital twin, on the other hand, is found to yield erroneous result almost immediately after the observation time-window.

One of the necessary characteristic of digital twin is its capability to update itself as more data becomes available. Figs. 4(b) and (c) show the results when more data is available and the digital twin has been updated accordingly. For Fig. 4(b), we have considered that we have access to 50 data points equally spaced in the time-window $[0, 250]$. Similarly, in Fig. 4(c), we have access to 50 data equally spaced in the time-window $[0, 550]$. We see that with longer observation window, the digital twin is able to capture the evolution of stiffness up to $t_s/T_0 = 1000$ (which is assumed to be the service life of the system). We also see that the predictive uncertainty for both these cases envelopes all the observed data points. This indicates that the uncertainty in the system due to limited and noisy measurements have been adequately captured. The GP-based digital twin has also been updated for the two cases. Although the GP based digital twin yields erroneous result for $\tau = 250$, it is found to yield satisfactory results for $\tau = 550$. However, the ME-GP based digital twin predicted results are still superior.

Lastly, we investigate the case when the noise variance is more. Fig. 5 shows the results corresponding to $\sigma_0 = 0.015$. Same three cases as Fig. 4 are considered. We observe that for $\tau = 150$, the prediction obtained using ME-GP based digital twin oscillates around the actual solution (Fig. 5 (a)). This oscillatory behaviour is probably because ME-GP overfits to the noise in the observations. The GP based digital twin is found to yield erroneous results. The digital twin models are then updated by collecting data up to $\tau = 250$ and $\tau = 550$. For both these cases, ME-GP based digital twin is found to yield reasonably good predictions. However, because of the increase in the noise variance, the predicted results are inferior to those shown in Fig. 4. This indicates the importance of collecting clean data from the physical system. The predictive uncertainty for $\tau = 250$ and $\tau = 550$ are also shown. We note that with increase in the observation time-window, the predictive uncertainty at $t/T_0 = 1000$ is almost similar. This is because the noise in the observations for both the cases are same and the last observation in Fig. 5 (c) is still significantly away from $t/T_0 = 1000$. Also, note that the predictive uncertainty for Fig. 5 is lesser as compared to Fig. 4. Such a counter intuitive observation can be attributed to the fact that with more noise, it becomes more difficult for all ML models including ME-GP to correctly capture the trend. A few observation points in Fig. 5 are outside the predictive uncertainty; this indicates that the ME-GP model is over-confident for this case. The GP based digital twin for all the three cases in Fig. 5 are found to yield erroneous results beyond the observation window. This indicates the superiority of the proposed ME-GP over GP, particularly when predicting the future.

4.2. Digital twin via mass evolution

Next, we consider the case corresponding to Section 3.1.2 where the variation in the damped natural frequency of the system is due to the variation in its mass. The evolution of mass is considered to be of multi-timescale nature as shown in Fig. 1(a). However, similar to the stiffness degradation case, neither the pattern of time-evolution nor the number of scales present in the data is known to us a priori. We again assume the sensor data to be transmitted intermittently at a fixed time interval. To emulate a realistic scenario, the damped natural frequency data is contaminated with white Gaussian noise having a standard deviation σ_0. The objective is to use the proposed digital twin to predict the multi-timescale evolution of mass.

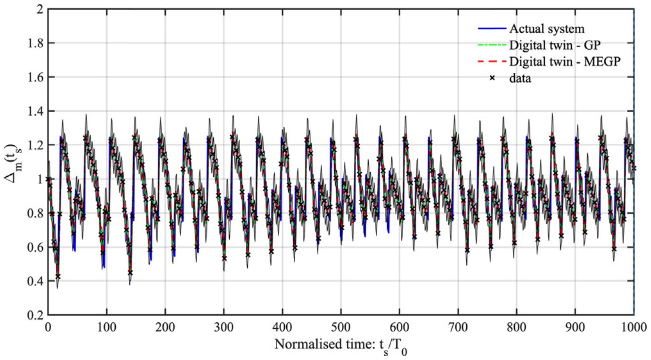
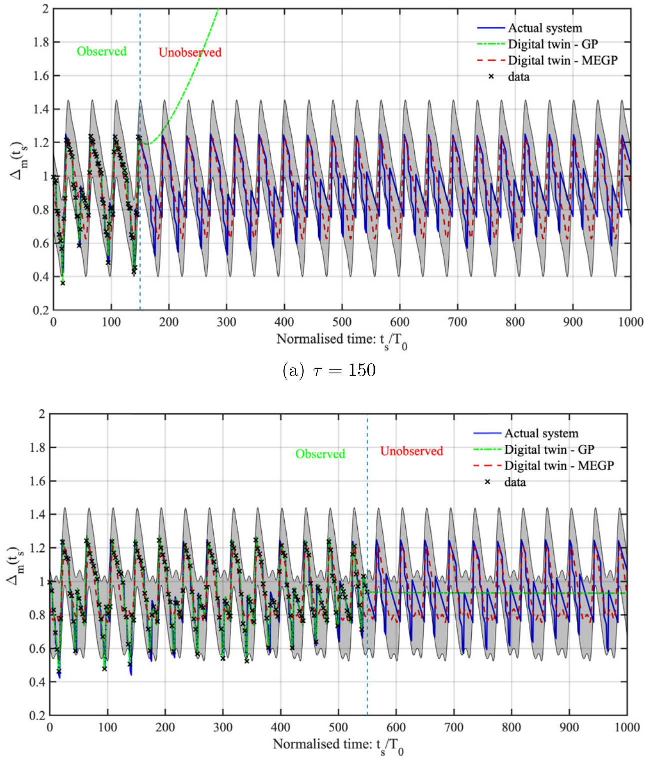


Fig. 6. Results obtained using GP based and ME-GP based digital twin for the case where only mass changes. Ideal scenario is considered where equally spaced clean data throughout the service life is available.

Consider that we have N_s observations of the damped natural frequency $\lambda_s(t_s)$ equally spaced in time $t_s \in [0, \tau]$. Using the procedure described in Section 3.1.2, we first process $\lambda_s(t_s)$ to obtain the change in the mass $\Delta_{\tilde{m}}(t_s)$. Thereafter, using t_s as the input and $\Delta_{\tilde{m}}(t_s)$ as the output, we train an ME-GP model by using Algorithms 1 and 2. The setup for the algorithms are kept similar to that described in Section 4.1 and the trained ME-GP model is treated as a surrogate of the unknown Δ_m. For obtaining the response of the system at t^*, one can utilize the trained ME-GP model to obtain the updated m at t^* and then substitute it into the nominal model and solve it to obtain the responses of interest. As more data becomes available, the digital twin is updated.

Fig. 6 shows the variation of Δ_m with normalized time t_s/T_0. These results correspond to the perfect case where we have 300 measurements equally spaced throughout the service life of the system. The data available are also free of any noise. We observe that the proposed digital twin and the GP based digital twin yields identical results. In other words, for this case, the proposed ME-GP based digital twin reduces to the GP based digital twin proposed in [21]. However, one must note that this is an ideal scenario. In a more realistic setting, the data collected will be noisy and, almost always, data over the complete service-life will not be available.

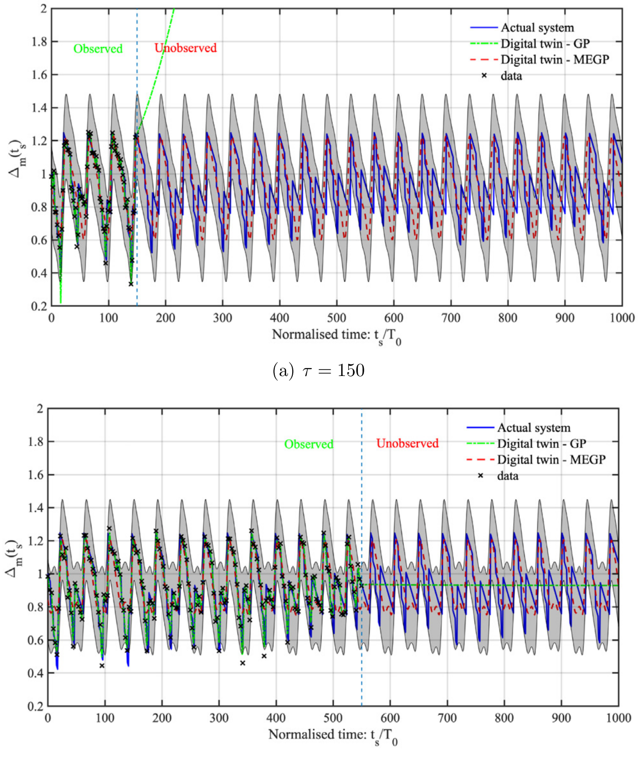
Next we consider a more realistic setup where we have data over a certain time-window $[0, \tau]$ and the goal is to predict Δ_m at a time t^* where $t^* > \tau$. The data collected is contaminated by white Gaussian noise. Fig. 7(a) shows the digital twin predicted results corresponding to $\tau = 150$. The data collected is contaminated by



(a) $\tau = 150$

(b) $\tau = 550$

Fig. 7. Results obtained using GP and ME-GP based digital twin with observation window of [0, 150] and [0, 550], and $\sigma_0 = 0.005$. For observation window of [0, 150], we have 75 sensor data whereas for observation window of [0, 550], we have 175 sensor data (shown using cross). The observed and unobserved regime are differentiated by using a vertical line.



(a) $\tau = 150$

(b) $\tau = 550$

Fig. 8. Results obtained using GP and ME-GP based digital twin with observation window of [0, 150] and [0, 550], and $\sigma_0 = 0.015$. For observation window of [0, 150], we have 75 sensor data whereas for observation window of [0, 550], we have 175 sensor data (shown using cross). The observed and unobserved regime are differentiated by using a vertical line.

white noise with $\sigma_0 = 0.005$. The observed and unobserved regimes are marked by a vertical line. Results using GP based digital twin have also been generated for the sake of comparison. For $\tau = 150$, the ME-GP model trained from only 75 data reasonable predicts the evolution of mass over the service life of the system. More specifically, the primary crest of the response has been accurately captured throughout the service life. However, as time increases, the digital twin over-predicts the trough. The proposed approach being Bayesian in nature also provides the predictive uncertainty as indicated by the shaded plot. Throughout the service life, the true solution is within the shaded plot, indicating that the uncertainty due to limited and noisy data is properly captured. The GP based digital twin, although provides excellent prediction in $t_s/T_0 = [0, 150]$, yields erroneous prediction almost immediately beyond the observation time-window.

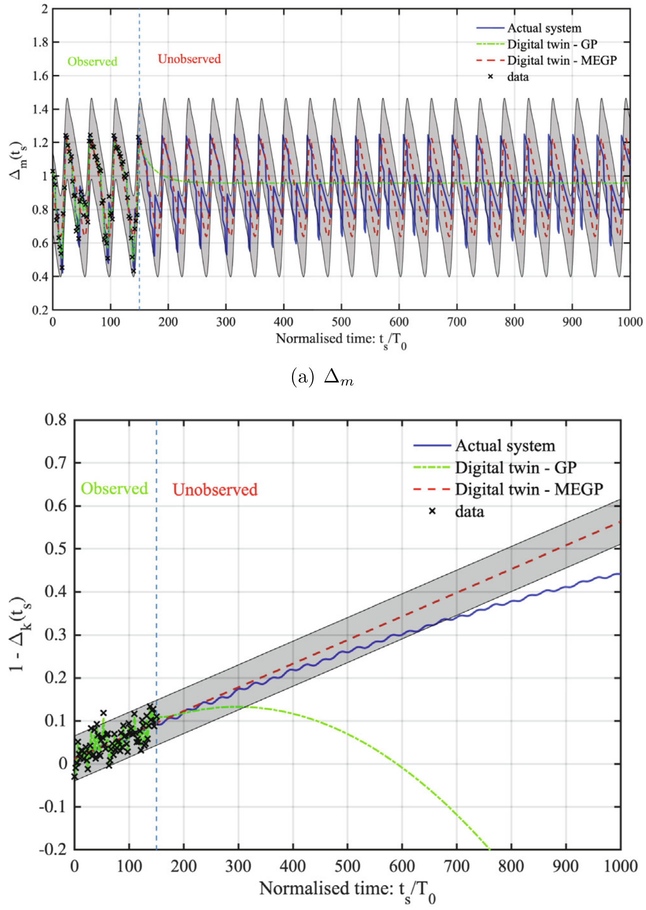
Fig. 7(b) shows the results obtained using the digital twins for $\tau = 550$. 175 equally spaced observations over the time-window are collected. The data collected is contaminated with white Gaussian noise having $\sigma_0 = 0.005$. Within the observation time-window, GP is found to yield better results as compared to proposed ME-GP. However, beyond the observation window, GP fails to provide a reasonable prediction. ME-GP, on the other hand, provides reasonable prediction even beyond the observation window.

Lastly, we consider the case when the noise variance is more. Fig. 8 shows the results corresponding to $\sigma_0 = 0.015$. The predicted time-evolution of mass is found to be similar to that predicted in Fig. 7. The predictive uncertainty is found to increase for this case. This can be attributed to the increase in the noise in the data. GP based digital twin was unable to properly predict the time-evolution beyond the observation window.

4.3. Digital twin via mass and stiffness evolution

Lastly, we illustrate the performance of the proposed digital twin when both mass and stiffness evolves with time. Evolution of both mass and stiffness are of multi-timescale nature as shown in Fig. 1. Similar to the previous cases, the sensor data is assumed to be transmitted intermittently at a fixed time interval. Considering we have N_s observations of the damped natural frequency, the digital twin first process this data to obtain Δ_m and Δ_k. Details on the data processing step are furnished in Section 3.1.3. Thereafter, ME-GP is used to learn the time evolution of mass and stiffness. The parameters for the ME-GP algorithm are kept identical as that discussed in Section 4.3. For obtaining response at a given time-instant t^*, one first obtains the mass and stiffness by using the trained ME-GP model as a surrogate. Thereafter, the responses of interest are obtained by substituting the ME-GP predicted mass and stiffness into the nominal model and solving it. Similar to the previous two cases, we only present the performance of the digital in predicting the time evolution of mass and stiffness; the argument being, if the time evolution of mass and stiffness are accurately captured, the responses predicted will also be accurate.

Given the fact that the proposed digital twin was found to yield almost exact result for the ideal scenario for the previous two cases, we directly proceed to the realistic case-studies. First, we consider the case where the observation window is $[0, 150]$. Within this time-window, it is assumed that we have access to 75 equally spaced sensor measurements. The data collected is contaminated by a white Gaussian noise with $\sigma_0 = 0.025$. Fig. 9 shows the evolution of Δ_m and Δ_k predicted using the digital twin. For Δ_m, it is observed that the digital twin yields reasonable prediction throughout the service life of the system. However, for Δ_k, the results are found to deviate beyond $t_s/T_0 = 350$. Since the proposed approach is Bayesian in nature, the predictive uncertainty has also been computed. For Δ_m, the predictive uncertainty envelopes the true behaviour throughout the service life; this indicates that the uncertainty due to noise and limited data has been appropriately captured. However, for Δ_k, the true solution is outside the envelope beyond $t_s/T_0 = 600$. This indicates that the proposed digital twin is over-confident when $t_s/T_0 > 600$. The GP based digital twin for both Δ_m and Δ_k yields erroneous results beyond $t_s/T_0 = 200$.



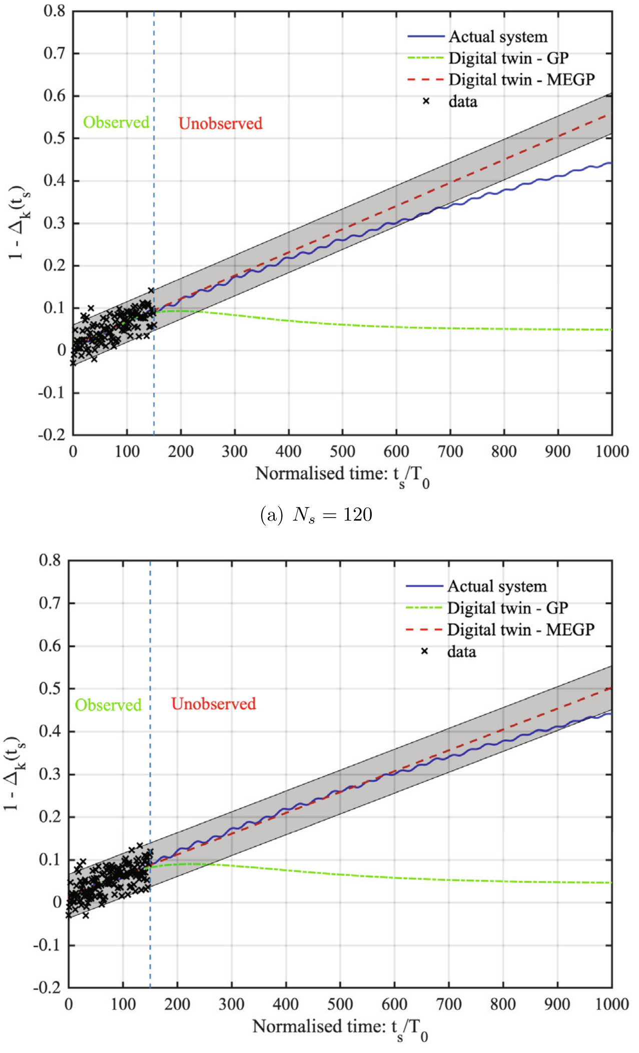
(a) Δ_m

(b) Δ_k

Fig. 9. Digital twin predicted responses, Δ_m and Δ_k. The digital twin is trained using 75 equally spaced sensor measurements in observation window $[0, 150]$. The sensor data are contaminated by white Gaussian noise with $\sigma_0 = 0.025$. The observed and unobserved regime are shown by a vertical line.

For improving the performance of the digital twin, we carried out investigation by providing additional sensor data to the model. More specifically, we provided 120 and 150 equally spaced observations within the same observation window $[0, 150]$. The results are shown in Fig. 10. As the predictions for Δ_m was already reasonable in Fig. 11, only the results corresponding to Δ_k are presented. We observe that with an increase in the number of observations, the digital twin predictions become more and more closer to the actual solution. The predictive uncertainty is also found to improve as it envelopes the true solution.

As the last case study, we carry out an investigation by increasing the observation window to $[0, 350]$. However, the number of sensor observations is kept fixed at 75. The result is shown in Fig. 11 With this setup, the digital almost perfectly predicts the time evolution of the stiffness. This illustrates the importance of collecting data over a longer time-span. For all the cases discussed, the GP based digital twin fails to provide accurate prediction beyond the observation window.



(a) $N_s = 120$

(b) $N_s = 150$

Fig. 10. Digital twin performance with increase in number of sensor measurements over the observation window $[0, 150]$. The sensor data are contaminated with white Gaussian noise having $\sigma_0 = 0.025$.

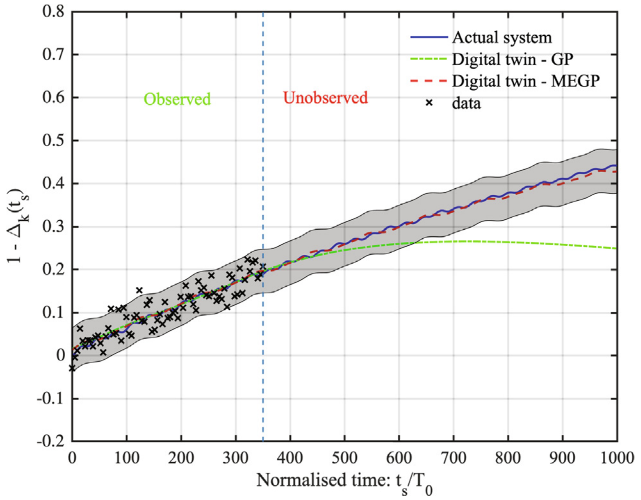


Fig. 11. Performance of digital twin for observation time-window $[0, 350]$. 75 sensor data are available within the specified time-window. The data are contaminated by white Gaussian noise with $\sigma_0 = 0.025$.

5. Conclusions

Digital twins of dynamical systems encountered in engineering and technology can benefit from the use of multiple time-scales. The solution of such systems is difficult from a computational point-of-view as we need a time-step of the order of the fastest scale. As a result, tasks such as health-monitoring, damage prognosis and remaining useful-life prediction becomes excessively difficult. To address this issue, we present a machine learning-based digital twin framework for multi-timescale dynamical systems. The proposed digital twin has two major components: (a) a physics-driven nominal model (generally represented by ordinary or partial differential equation(s)) and (b) a data-driven machine learning model. We use the physics-driven nominal model for data processing and predictions and the machine learning model for learning the time-evolution of the system parameters. We propose to use a mixture of experts (MOE) as the machine learning model of choice. As an expert within the MOE framework, we propose to use GP. The basic idea is that each of the experts will track the temporal evolution of the system parameters at a single scale. For learning the hyperparameters of the proposed model, an algorithm based on expectation-maximization and sequential Monte Carlo sampler is proposed. The proposed training algorithm is of hybrid nature where some of the parameters are treated in a Bayesian sense while the point-estimates for others are provided.

The results obtained using the proposed approach show its ability in predicting the time-evolution of the system parameters, even outside the observation time-window. However, extremely sparse and highly noisy data can affect the performance of the proposed framework. Moreover, collecting data over a longer observation window can drastically improve the performance of the proposed framework. The proposed framework being partially Bayesian quantifies the uncertainty due to limited and noisy data. In most of the cases, the predictive uncertainty is found to envelop the true solution, indicating that the uncertainty is properly captured. However, for some cases, the true solution is found to be outside the envelope. This is probably because of overfitting and employing a fully-Bayesian framework can help us overcome this issue.

Despite the several interesting findings of the current work, there are three possible extensions that need to be pursued in future. First, the illustration has been carried out using a single degree of freedom system. While this helps us in understanding the functionality of the proposed framework, further investigation on multi-degree-of-freedom (MDOF) systems is necessary. For MDOF systems, one major challenge will be in the data-processing step. This is because deriving closed-form relation between frequency measurements and system parameters is difficult, if not impossible. Secondly, we have used a damped natural frequency to be our observation. However, this is a derived quantity based on strain measurements. It is necessary to extend the framework to directly learn the evolution of mass and stiffness from time-history measurements. Last but not the least, the framework proposed in this paper doesn't utilize the physics and the data simultaneously; instead the physics and data are utilized in a decoupled way. There are methods rooted in physics-informed machine learning framework where the two can be used simultaneously. Potential of such methods within the digital twin framework needs to be investigated.

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.

Acknowledgements

SC acknowledges the support of Technology Innovation Hub (TIH) for Cobotics, IIT Delhi. SA acknowledges the financial support from The Engineering Physical Science Research Council (EPSRC) through a programme grant EP/R006768/1.

References

[1] Rasheed A, San O, Kvamsdal T. Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access 2020;8( ):21980–2012.

[2] Tao F, Zhang H, Liu A, Nee A. Digital twin in industry: state of the art. IEEE Trans Industr Inf 2018;15(4):2405–15.

[3] Digital twin towards a meaningful framework, Tech. rep., Arup, London, England (November 2019). http://www.arup.com/digitaltwinreport.

[4] Worden K, Cross E, Gardner P, Barthorpe R, Wagg D. On digital twins, mirrors and virtualisations. Model Valid Uncertain Quantif 2020;3(1):285–95.

[5] Booyse W, Wilke DN, Heyns S. Deep digital twins for detection, diagnostics and prognostics. Mech Syst Signal Process 2020;140:106612.

[6] Wang J, Ye L, Gao R, Li C, Zhang L. Digital twin for rotating machinery fault diagnosis in smart manufacturing. Int J Prod Res 2019;57(12):3920–34.

[7] Tao F, Sui F, Liu A, Qi Q, Zhang M, Song B, et al. Digital twin-driven product design framework. Int J Prod Res 2019;57(12):3935–53.

[8] Millwater H, Ocampo J, Crosby N. Probabilistic methods for risk assessment of airframe digital twin structures. Eng Fract Mech 2019;221:106674.

[9] Zhou M, Yan J, Feng D. Digital twin framework and its application to power grid online analysis. CSEE J Power Energy Syst 2019;5(3):391–8.

[10] Debroy T, Zhang W, Turner J, Babu S. Building digital twins of 3d printing machines. Scripta Mater 2017;135:119–24.

[11] Haag S, Anderl R. Digital twin - proof of concept. Manuf Lett 2018;15:64–6.

[12] Lu Y, Liu C, Kevin I, Wang K, Huang H, Xu X. Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robot Comput Integr Manuf 2020;61:101837.

[13] Park KT, Lee D, Noh SD. Operation procedures of a work-center-level digital twin for sustainable and smart manufacturing. Int J Precision Eng Manuf-Green Technol 2020;7:791–814.

[14] He B, Bai K-J. Digital twin-based sustainable intelligent manufacturing: a review. Adv Manuf 2019:1–21.

[15] Li C, Mahadevan S, Ling Y, Choze S, Wang L. Dynamic bayesian network for aircraft health monitoring digital twin. AIAA J 2017;55(3):930–41.

[16] Kapteyn MG, Knezevic DJ, Willcox K. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In: AIAA Scitech 2020 Forum; 2020. p. 0418.

[17] Tuegel E, Ingraffea A, Eason T, Spottswood S. Reengineering aircraft structural life prediction using a digital twin. Int J Aerosp Eng 2011;2011 [Art. 154768].

[18] Hoodorozhkov S, Krasilnikov A, Digital twin of wheel tractor with automatic gearbox. In: E3S Web of Conferences, vol. 164, EDP Sciences, 2020. p. 03032.

[19] Adhikari S, Bhattacharya S. Dynamic analysis of wind turbine towers on flexible foundations. Shock Vib 2012;19(1):37–56.

[20] Ganguli R, Adhikari S. The digital twin of discrete dynamic systems: Initial approaches and future challenges. Appl Math Model 2020;77:1110–28.

[21] Chakraborty S, Adhikari S, Ganguli R. The role of surrogate models in the development of digital twins of dynamic systems. Appl Math Model 2021;90:662–81.

[22] Chakraborty S. Simulation free reliability analysis: A physics-informed deep learning based approach, arXiv preprint arXiv:2005.01302; 2020.

[23] Goswami S, Anitescu C, Chakraborty S, Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theoret Appl Fract Mech 2020;106:102447.

[24] Chakraborty S. Transfer learning based multi-fidelity physics informed deep neural network, arXiv preprint arXiv:2005.10614; 2020.

[25] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput Methods Appl Mech Eng 2020;362:112790.

[26] Bilionis I, Zabaras N, Konomi BA, Lin G. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. J Comput Phys 2013;241:212–39.

[27] Bilionis I, Zabaras N. Multi-output local Gaussian process regression: Applications to uncertainty quantification. J Comput Phys 2012;231(17):5718–46.

[28] Nayek R, Chakraborty S, Narasimhan S. A gaussian process latent force model for joint input-state estimation in linear structural systems. Mech Syst Sig Process 2019;128:497–530.

[29] Said SE, Dickey DA. Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika 1984;71(3):599–607.

[30] Box GE, Pierce DA. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J Am Stat Assoc 1970;65 (332):1509–26.

[31] Kaur MJ, Mishra VP, Maheshwari P. The convergence of digital twin, iot, and machine learning: Transforming data into action. In: Digital Twin Technologies and Smart Cities. Springer; 2020. p. 3–17.

[32] Koutsourelakis P-S, Bilionis E. Scalable bayesian reduced-order models for simulating high-dimensional multiscale dynamical systems. Multiscale Model Simul 2011;9(1):449–85.

[33] Chakraborty S, Zabaras N. Efficient data-driven reduced-order models for high-dimensional multiscale dynamical systems. Comput Phys Commun 2018. https://doi.org/10.1016/j.cpc.2018.04.007.

[34] Feng D, Feng M, Ozer E, Fakuda Y. A vision-based sensor for noncontact structural displacement measurement. Sensors 2015;15(7):16557–75.

[35] Wang Q, Huang J, Liu Q, Zhou Z. Dynamic strain measurement of hydraulic system pipeline using fibre bragg grating sensors. Adv Mech Eng 2016;8(4). 1687814016645069.

[36] Del Moral P, Doucet A, Jasra A. Sequential monte carlo samplers. J Roy Stat Soc Ser B (Stat Methodol) 2006;68(3):411–36.

[37] Murphy KP. Machine learning: a probabilistic perspective. MIT Press; 2012.

[38] Doucet A, Godsill S, Andrieu C. On sequential monte carlo sampling methods for bayesian filtering. Stat Comput 2000;10(3):197–208.

[39] Wan J, Zabaras N. A probabilistic graphical model approach to stochastic multiscale partial differential equations. J Comput Phys 2013;250:477–510.

[40] Liu JS, Chen R. Blind deconvolution via sequential imputations. J Am Stat Assoc 1995;90(430):567–76.