frontiers | Frontiers in Materials

TYPE Original Research
PUBLISHED 17 October 2022
DOI 10.3389/fmats.2022.970970

Check for updates

OPEN ACCESS

EDITED BY
Holger Steeb,
University of Stuttgart, Germany

REVIEWED BY
Felix Fritzen,
University of Stuttgart, Germany
Ettore Barbieri,
Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Japan
Ralf Jänicke,
Technische Universitat Braunschweig, Germany

*CORRESPONDENCE
Francisco Chinesta,
Francisco.CHINESTA@ensam.eu

SPECIALTY SECTION
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

RECEIVED 16 June 2022
ACCEPTED 05 September 2022
PUBLISHED 17 October 2022

CITATION
Champaney V, Amores VJ, Garois S, Irastorza-Valera L, Ghnatios C, Montáns FJ, Cueto E and Chinesta F (2022), Modeling systems from partial observations. *Front. Mater.* 9:970970.
doi: 10.3389/fmats.2022.970970

Modeling systems from partial observations

Victor Champaney[1], Víctor J. Amores[2], Sevan Garois[1], Luis Irastorza-Valera[1,2], Chady Ghnatios[1], Francisco J. Montáns[2,3], Elías Cueto[4] and Francisco Chinesta[1,5]*

[1]PIMM Lab, UMR CNRS, Arts et Métiers Institute of Technology, Paris, France, [2]Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, Madrid, Spain, [3]Herbert College of Engineering, University of Florida, Gainesville, FL, United States, [4]Aragón Institute of Engineering Research, Universidad de Zaragoza, Zaragoza, Spain, [5]CNRS@CREATE LTD., Singapore, Singapore

Modeling systems from collected data faces two main difficulties: the first one concerns the choice of measurable variables that will define the learnt model features, which should be the ones concerned by the addressed physics, optimally neither more nor less than the essential ones. The second one is linked to accessibility to data since, generally, only limited parts of the system are accessible to perform measurements. This work revisits some aspects related to the observation, description, and modeling of systems that are only partially accessible and shows that a model can be defined when the loading in unresolved degrees of freedom remains unaltered in the different experiments.

KEYWORDS

partial observability, AI, machine learning, recurrent NN, LSTM, static condensation

1 Introduction

Q24

Simulation-based engineering (SBE) considers well-experienced physics-based models that are expected to describe the reality under scrutiny. These models must be calibrated from experiments in order to fine-tune the different parameters they involve. After that, they can be used to make predictions on the evolution of the considered system. The numerical solution of these models needs adequate numerical techniques to discretize the partial differential equations by which they are usually described, as well as powerful computing platforms to solve them efficiently.

Such a rationale was the main driving factor of last century engineering and faces two main limitations. The first handicap we must face is the fact that engineering is nowadays more concerned with performance than with the products themselves. Thus, engineering is expected to follow-up or monitor its designs all along their lives, resulting in the so-called Digital Twins, Tuegel et al. (2011); Chinesta et al. (2020); Ghanem et al. (2022); Kapteyn and Willcox (2020); Moya et al. (2022b); Argerich et al. (2020); Sancarlos et al. (2021b, a). However, they use craves for fast and accurate responses, and SBE often found difficulties to encompass real-time feedback. This limitation was alleviated by use of advanced model order reduction techniques Chinesta et al. (2011), Chinesta et al. (2013), Chinesta et al. (2014), Chinesta et al. (2015); Ibáñez et al. (2018); Borzacchiello et al. (2019); Sancarlos et al. (2021c). The second difficulty appears when the model solution exhibits a significant deviation with respect to the reality that it is expected to describe. The main reasons for the just mentioned deviations are incompleteness of models to recreate a complex reality and the intrinsic uncertainty and variability in our approximation to the physical reality.

Recent machine learning techniques enable alleviating the just referred limitations within the so-called fourth paradigm of sciences. On the one hand, accurate regressions can be constructed from the input/output data, which will later allow for collection of the output from the input in almost real-time, like model order reduction techniques (referred previously) performed on the solution of the complex mathematical models. The data manipulated by machine learning techniques can be experimental or synthetic (coming from high-fidelity simulations). On the other hand, when the considered data come from the real system, assumed free of noise, the learnt model will represent the reality in a very accurate manner, sometimes with higher accuracy than the existing physics-based models, Fasel et al. (2022).

In between the fully physics-based and the fully data-driven perspectives, an intermediate setting exists: the so-called hybrid paradigm, at the origin of the so-called Hybrid Twins Chinesta et al. (2020), that can be viewed as instances of physics-augmented learning or transfer learning, Weiss et al. (2016).

When considering machine learning techniques, the choice is very large. First, the adequate technique depends on the type of data to be manipulated. There are powerful tools to process images [e.g., convolutional neural networks, CNNs, Venkatesan and Li. (2017)], graphs [e.g., graph neural networks, GNNs, Bronstein et al. (2021) Hernandez et al. (2022)], time series [e.g., recurrent neural networks, rNNs, and long short-time memory, LSTM, Hochreiter and Schmidhuber (1997); Zhou et al. (2016)] .When data are very rich, many correlations may exist, and prior to proceeding with modeling, data-reduction seems a valuable route, with many manifold learning approaches available. Nonlinear dimensionality reduction can be efficiently performed in a nonlinear setting by using, for example, auto-encoders Goodfellow et al. (2016); Schmidhuber (2015); Hinton and Zemel (1993). Sparse autoencoders are of particular importance Ng (2011); Makhzani and Frey (2014).

When proceeding with data for modeling purposes, a recurrent issue concerns data accessibility. Sometimes, the considered system is not globally accessible, with only a small part of it being accessible to perform measurements.

The present study addresses a conceptual issue that will be discussed on an example simple enough to be fully understood, and at the same time complex enough to encompass all the modeling issues discussed in the present study.

The main question could be formulated as follows: if there is a part of a system inaccessible for observation in which a loading that we cannot either observe or measure applies, and that influences the measures performed in the observable part of the system, different questions arise:

- is there a model connecting the observable input(s) to its output(s), knowing that they are impacted by the hidden dynamics of the system? Is it unique?
- Under which conditions that model could exist? How to find it?
- How to formulate it correctly? Is it well-posed and consistent?
- How to learn it?
- What is the impact of these hidden dynamics on the learning process?

We referred previously to the use of rNN or LSTM, whose choice is guided by the knowledge gained from physics and mathematics, excellent allies of machine learning (ML). It is well-known that ML technologies such as rNN or LSTM allow us to manage, in transient problems, the hidden variables that, even if they are not observed, influence the evolution of the data in the observable regions Williams et al. (2022); Manohar et al. (2018).

This study aims at revisiting the construction of models in the domains exhibiting partial observability, in both the steady and transient cases, while following a double approach: the usual algebraic formulation and the one concerned by machine learning approaches.

2 On the existence of models relating observable features

In this section, we assume a large system, whose state is described by a number of state variables. We consider that the variables involved in the state description are well-defined. However, the model governing the state or its time evolution is assumed to be unknown, and the data describing the state are only observable and measurable on a part of the system, remaining unattainable in the rest of the system. Previous analysis on the field can be found in González et al. (2021) or Moya et al. (2022a).

For instance, in the case of the two-mass oscillator depicted in **Figure 1**, we assume that the state is perfectly defined by the position and momentum of each mass; however, only the state of the second mass is accessible (and thus, measurable). A natural question concerns the possibility of learning the model that governs the observable state (q_2,p_2) while ignoring the state of the first mass (q_1,p_1).

In the following section, we address this question using a quite generic algebraic rationale in two situations: a model that

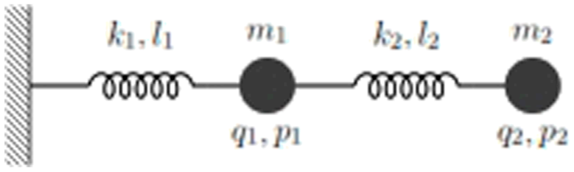

FIGURE 1
Oscillator composed of two masses, two linear springs of stiffness k_1 and k_2, reference lengths l_1 and l_2, and whose state is defined by the position and momentum of each mass (q_1, p_1, q_2, p_2).

does not depend on time and a transient problem. We will discuss the multiple-mass oscillators later. Henceforth, more generic settings are considered.

2.1 Time-independent problem

A generic linear time-independent model can be expressed from:

$$\mathbf{K}\mathbf{U} = \mathbf{F}, \tag{1}$$

which, considering the observable variables $\mathbf{U_o}$ and the internal ones $\mathbf{U_i}$, can be rewritten as follows:

$$\begin{pmatrix} \mathbf{K_{oo}} & \mathbf{K_{oi}} \\ \mathbf{K_{io}} & \mathbf{K_{ii}} \end{pmatrix} \begin{pmatrix} \mathbf{U_o} \\ \mathbf{U_i} \end{pmatrix} = \begin{pmatrix} \mathbf{F_o} \\ \mathbf{F_i} \end{pmatrix}. \tag{2}$$

Developing the last equation, we find that

$$\mathbf{K_{io}}\mathbf{U_o} + \mathbf{K_{ii}}\mathbf{U_i} = \mathbf{F_i} \rightarrow \mathbf{U_i} = \mathbf{K_{ii}^{-1}}\mathbf{F_i} - \mathbf{K_{ii}^{-1}}\mathbf{K_{io}}\mathbf{U_o}. \tag{3}$$

Also, introducing the resulting expression of $\mathbf{U}_i$ into the development of the first, we obtain (this is known as static condensation or Guyan reduction)

$$\left(\mathbf{K_{oo}} - \mathbf{K_{oi}}\mathbf{K_{ii}^{-1}}\mathbf{K_{io}}\right)\mathbf{U_o} = \mathbf{F_o} - \mathbf{K_{oi}}\mathbf{K_{ii}^{-1}}\mathbf{F_i}, \tag{4}$$

which can be rewritten as

$$\tilde{\mathbf{K}}_{\mathbf{oo}}\mathbf{U_o} = \mathbf{F_o} - \tilde{\mathbf{F}}_{\mathbf{i}}, \tag{5}$$

with

$$\left\{ \begin{array}{l} \tilde{\mathbf{K}}_{\mathbf{oo}} = \left(\mathbf{K_{oo}} - \mathbf{K_{oi}}\mathbf{K_{ii}^{-1}}\mathbf{K_{io}}\right) \\ \tilde{\mathbf{F}}_{\mathbf{i}} = \mathbf{K_{oi}}\mathbf{K_{ii}^{-1}}\mathbf{F_i} \end{array} \right. . \tag{6}$$

Remark 1:

- If $\mathbf{F_i} = \mathbf{0}$, a direct relation exists between $\mathbf{U_o}$ and $\mathbf{F_o}$.
- In the case of a 1D system in which only the borders of the interval are accessible (observable), $\mathbf{U_o}$ and $\mathbf{F_o}$ contain two components. If we apply $\mathbf{U_o}^T = (1, 0)$, the resulting $\mathbf{F_o}$ represents the first column of $\tilde{\mathbf{K}}_{\mathbf{oo}}$, and the solution $\mathbf{F_o}$ related to $\mathbf{U_o}^T = (0, 1)$ will represent the second column of $\tilde{\mathbf{K}}_{\mathbf{oo}}$.
- In the same one-dimensional system, when $\mathbf{F_i} \neq \mathbf{0}$, there are two effective internal variables, the components of $\tilde{\mathbf{F}}_{\mathbf{i}}$. Thus, all the richness of $\mathbf{F_i}$ endangers these two components, generating some sort of irreversibility: from $\mathbf{F_i}$, we can obtain $\tilde{\mathbf{F}}_{\mathbf{i}}$, but from the last one, we cannot come back to the former. The condensation of the internal degrees of freedom into the observable one produces an entropy increase in the theory of information sense: there are many micro-states $\mathbf{F_i}$ associated with the macro-state $\tilde{\mathbf{F}}_{\mathbf{i}}$.
- Computing these two effective internal variables just referred needs extra-calculation. For example, if $\mathbf{U_o} = \mathbf{0}$, then $\mathbf{F_o} = \tilde{\mathbf{F}}_{\mathbf{i}}$.

2.2 Time-dependent problem

A general linear second-order dynamical system can be expressed from

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{F}, \tag{7}$$

which, applying Fourier transform, becomes

$$-\omega^2\mathbf{M}\mathbb{U} + i\omega\mathbf{C}\mathbb{U} + \mathbf{K}\mathbb{U} = \mathbb{F}, \tag{8}$$

with $i \equiv \sqrt{-1}$ (without confusion with respect to other i subscript symbols used in the study, like the index i referring to the internal variables or the index i in vector components) and $\mathbb{U}$ and $\mathbb{F}$, the Fourier transforms of $\mathbf{U}$ and $\mathbf{F}$, respectively. The previous equation can be rewritten as follows:

$$\mathbf{K}^*\mathbb{U} = \mathbb{F}, \tag{9}$$

with $\mathbf{K}^* = -\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K}$, that can be separated in the same way considered in the time-independent case, but now, for each possible frequency (ω) involved in the loading and operating in the complex domain, leading to

$$\tilde{\mathbf{K}}^*_{\mathbf{oo}}\mathbb{U}_{\mathbf{o}} = \mathbb{F}_{\mathbf{o}} - \tilde{\mathbb{F}}_{\mathbf{i}}, \tag{10}$$

which proves that all the discussion previously addressed in the time-independent case remains valid as soon as the Fourier transform applies.

Thus, one could expect that a model relating observable variables might exist as well (and could be learnt from collected data) in the time domain, under certain constraints, as the one referred in *Remark 2* below, due to the dependence of $\tilde{\mathbb{F}}_i$ on the internal loading $\mathbb{F}_i$. This would imply the consideration of the history of the variables, which is naturally implicit in the Fourier transform. We will discuss this point later.

Remark 2:

The just-described rationale applies in the forced regime, i.e., far from the transient effects induced by the initial condition. In order to address transient regimes, the Laplace transform could be employed instead of the Fourier one. However, it is well-known that the Laplace inverse transform is less simple from the numerical point of view than Fourier's. It is also important to note that the Fourier transform of the internal loading considered in the training stage should remain invariant to ensure the validity of the learnt model.

2.3 Neural network-based modeling

In many cases, artificial intelligence, and more concretely machine learning, aims at extracting the model that relates measured inputs to the corresponding outputs Brunton and Kutz (2019); Liu and Tegmark (2021). In general, the measured output depends on the whole internal state. For instance, in a structural dynamics problem where the loading (evolving in time) constitutes the problem's input, the corresponding response is the displacement at each point and time, whereas the corresponding output data are the measured displacement in a certain observable point of the structure.

In physics-based structural mechanics, the internal response (displacement at any location and time instant) is obtained by discretization of the continuum mechanics model, consisting of the momentum balance and the constitutive equations; from this internal state, the output of interest is directly extracted at each time instant. Alternatively, machine learning looks for the direct relation between observables, the input action, and the measured response that, as just mentioned, can depend on the present and past values of a series of non-observed internal variables Lee and Carlberg (2020).

Recurrent neural networks (rNNs) and their long-short time memory counterparts (LSTM) address such situations by trying to model the time evolution of the internal state at the same time it constructs the model relating the observable input and output (action and response). For the sake of completeness, revisits both rNN and LSTM neural networks.

2.4 Addressing time-dependent problems in the time domain

Finally, to reinforce the main conclusions of **Section 2.2**, we are briefly discussing time-dependent problems modeling but directly operating in the time domain, instead of operating in the Fourier domain as was considered before. For simplicity, we contemplate the first-order dynamical system

$$\mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{F}, \tag{11}$$

whose implicit time discretization reads

$$\mathbf{C}\mathbf{U}^n + \Delta t\mathbf{K}\mathbf{U}^n = \Delta t\mathbf{F}^n + \mathbf{C}\mathbf{U}^{n-1}, \tag{12}$$

with Δt being the considered time step. This equation can be rewritten in the more compact form

$$\mathbf{K}^*\mathbf{U}^n = \mathbf{F}^{*,n} + \mathbf{C}\mathbf{U}^{n-1}, \tag{13}$$

with $\mathbf{K}^* = \mathbf{C} + \Delta t\mathbf{K}$ and $\mathbf{F}^{*,n} = \Delta t\mathbf{F}^n$.

The sequencing of these equations can be written, inspired by the dynamic model decomposition, in the matrix form

$$\mathbf{K}^*\left[\mathbf{U}^n, \ldots, \mathbf{U}^1\right] = \left[\mathbf{F}^{*,n}, \ldots, \mathbf{F}^{*,1}\right] + \mathbf{C}\left[\mathbf{U}^{n-1}, \ldots, \mathbf{U}^0\right], \tag{14}$$

and by defining the extended vectors $\mathcal{U}$ and $\mathcal{F}$,

$$\begin{cases} \mathcal{U}^T = \left[\mathbf{U}^{n^T}, \mathbf{U}^{n-1^T}, \ldots, \mathbf{U}^{0^T}\right] \\ \mathcal{F}^T = \left[\mathbf{F}^{n^T}, \mathbf{F}^{n-1^T}, \ldots, \mathbf{F}^{1^T}\right] \end{cases}, \tag{15}$$

and the extended matrix $\mathcal{K}$

$$\mathcal{K} = \begin{pmatrix} \mathbf{K}^* & -\mathbf{C} & 0 & \ldots & \ldots \\ 0 & \mathbf{K}^* & -\mathbf{C} & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}, \tag{16}$$

The previous system reads

$$\mathcal{K}\mathcal{U} = \mathcal{F}, \tag{17}$$

where the solution $\mathcal{U}$ is, in general, computed from the $\mathcal{K}$ matrix pseudo-inverse.

This algebraic system can be addressed by using the same rationale that was applied before, but this time, the model will explicitly involve the time evolution of the input(s) and output(s), reinforcing the result already obtained when using the Fourier transform.

Another alternative formulation, more aligned with the use of machine learning techniques that will be presented afterward, consists of writing the explicit integration

$$\mathbf{C}\mathbf{U}^n = \Delta t\mathbf{F}^n - \Delta t\mathbf{K}\mathbf{U}^{n-1} + \mathbf{C}\mathbf{U}^{n-1}, \tag{18}$$

that can be reformulated as

$$\mathbf{U}^n = \mathbf{A}\mathbf{F}^n + \mathbf{B}\mathbf{U}^{n-1}, \tag{19}$$

perfectly expressible within the rNN architecture illustrated in Figure 12. When the model concerns only a part of the state (the observable part), rNN and/or LSTM seem especially appealing to carry out the task.

3 Results

This section addresses, as indicated in the introduction, some numerical examples, simple enough to be perfectly understood, but complex enough to underline all the issues and methodological aspects just discussed.

One could think that working with a three-mass dynamical system while observing only the state of one of them is too simple.

It is in fact very simple to visualize, and this was the primary objective: being easy to reproduce because such a model is quite simple to understand and replicate and check all the discussions that we are discussing in the present section.

However, this simplicity is only apparent. Forces are being applied to the internal masses, unknown and unobserved by the modeler, who, furthermore, totally ignores how many hidden masses are involved in the system. We consider three in the present example, but they could come in any number, from one to thousands.

When introducing all the system's degrees of freedom—in our case, the state of the three masses—in a model, the last one becomes larger but finally simpler to interpret and to learn since all the needed data for properly describing the system are there, fully available. On the contrary, when considering only the data associated to one mass, while ignoring all the data related to all the other masses, the model seems simpler from its size, but very intricate nonetheless.

For this reason, and this was our motivation, the simplicity is only apparent and allows for a more fruitful discussion on the issues and the conceptual questions previously addressed.

For completeness, the different programs elaborated and used in the numerical examples addressed in the present section, in particular the rNN and LSTM neural network architectures (with their associated hyper-parameters), are available at https://github.com/cghnatios/LSTM-rNN-for-Modeling-systems-from-partial-observations-.git.

3.1 Learning in the Fourier space

We consider the linear N-mass dynamical system, including inertia, elastic, and damping behaviors, illustrated in **Figure 2**.

The state of each mass is represented by $\mathbf{z}_i = (q_i, p_i)$, with q_i and p_i being the i-th mass position and momentum, respectively. We define the system state from the extended vector $\mathbf{Z}^T = (\mathbf{z}_1^T, \ldots, \mathbf{z}_N^T)$.

The usual model, coming from Newton's equation, can be expressed by

$$\dot{\mathbf{Z}} = \mathbf{TZ} + \mathbf{J} + \mathbf{F}, \tag{20}$$

where matrix $\mathbf{T}$ includes the system properties, masses, spring stiffness, and viscosity of the dampers. On the other hand, $\mathbf{J}$ is a constant vector (in the linear case addressed below) and $\mathbf{F}$ contains the external forces applied to the different masses, appearing at odd positions in vector $\mathbf{F}$ (an explicit form of that matrix, and those vectors will be given later).

In the forced regime, the Fourier transformation becomes a valuable route. The dynamical model in the Fourier domain reads:

$$(-\mathbf{T} + i\omega\mathbf{I})\,\mathbb{Z} = \mathbb{J} + \mathbb{F}. \tag{21}$$

By defining the effective loading $\mathbb{S} = \mathbb{J} + \mathbb{F}$ and $\tilde{\mathbf{T}} = -\mathbf{T} + i\omega\mathbf{I}$, we can write the matrix form that separates the degrees of freedom related to the measurable position (noted by q) and the derived momentum (p):

$$\begin{pmatrix} \tilde{\mathbf{T}}_{qq}(\omega) & \mathbf{T}_{qp} \\ \mathbf{T}_{pq} & \tilde{\mathbf{T}}_{pp}(\omega) \end{pmatrix} \begin{pmatrix} \mathbb{Z}_q(\omega) \\ \mathbb{Z}_p(\omega) \end{pmatrix} = \begin{pmatrix} \mathbb{S}_q(\omega) \\ \mathbb{S}_p(\omega) \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbb{S}_p(\omega) \end{pmatrix}. \tag{22}$$

Being $\mathbb{S}_q(\omega) = 0$, $\mathbb{Z}_p(\omega)$ can be expressed in terms of $\mathbb{Z}_q(\omega)$:

$$\tilde{\mathbf{T}}_{qq}(\omega)\,\mathbb{Z}_q(\omega) + \mathbf{T}_{qp}\mathbb{Z}_p(\omega) = 0 \rightarrow \mathbb{Z}_p(\omega) = -\mathbf{T}_{qp}^{-1}\tilde{\mathbf{T}}_{qq}(\omega)\,\mathbb{Z}_q(\omega), \tag{23}$$

that, introduced into the second equation, leads to

$$\left(\mathbf{T}_{pq} - \tilde{\mathbf{T}}_{pp}(\omega)\,\mathbf{T}_{qp}^{-1}\tilde{\mathbf{T}}_{qq}(\omega)\right)\mathbb{Z}_q(\omega) = \mathbb{S}_p(\omega), \tag{24}$$

which can be reshaped into a more compact form:

$$\mathbf{A}(\omega)\,\mathbb{Q}(\omega) = \mathbb{R}(\omega), \tag{25}$$

with $\mathbf{A}(\omega) = \mathbf{T}_{pq} - \tilde{\mathbf{T}}_{pp}(\omega)\,\mathbf{T}_{qp}^{-1}\tilde{\mathbf{T}}_{qq}(\omega)$, $\mathbb{Q}(\omega) \equiv \mathbb{Z}_q(\omega)$, and $\mathbb{R}(\omega) \equiv \mathbb{S}_p(\omega)$.

This way, we have removed the momentum from the state variables since it derives directly from the measurable position.

Now, the partition between the internal and the observable degrees of freedom can be enforced:

$$\begin{pmatrix} \mathbf{A_{oo}}(\omega) & \mathbf{A_{oi}}(\omega) \\ \mathbf{A_{io}}(\omega) & \mathbf{A_{ii}}(\omega) \end{pmatrix} \begin{pmatrix} \mathbb{Q}_\mathbf{o}(\omega) \\ \mathbb{Q}_\mathbf{i}(\omega) \end{pmatrix} = \begin{pmatrix} \mathbb{R}_\mathbf{o}(\omega) \\ \mathbb{R}_\mathbf{i}(\omega) \end{pmatrix} \tag{26}$$

such that following the aforementioned rationale leads to

$$\tilde{\mathbf{A}}_\mathbf{oo}(\omega)\,\mathbb{Q}_\mathbf{o}(\omega) = \mathbb{R}_\mathbf{o}(\omega) - \tilde{\mathbb{R}}_\mathbf{i}(\omega), \tag{27}$$

with $\tilde{\mathbb{R}}_\mathbf{i}(\omega) = \mathbf{A_{oi}}(\omega)\,\mathbf{A_{ii}}^{-1}(\omega)\,\mathbb{R}_\mathbf{i}(\omega)$ and $\tilde{\mathbf{A}}_\mathbf{oo}(\omega) = \mathbf{A_{oo}}(\omega) - \mathbf{A_{oi}}(\omega)\,\mathbf{A_{ii}}^{-1}(\omega)\,\mathbf{A_{io}}(\omega)$, where the same remarks that were previously discussed apply.

As a particular example, we consider a system composed of three identical masses ($m_1 = m_2 = m_3 = m$), springs ($k_1 = k_2 = k_3 = k$), and dampers ($c_1 = c_2 = c_3 = c$), with the springs having a reference length also identical ($l_1 = l_2 = l_3 = l$). Forces can be applied on both the internal masses (the first two) and on the observable one, the third. The following values are considered: $m = 0.5$ Kg, $c = 0.8$ N/m, $k = 1$ N/m, and $l = 1$ m.

The dynamical model reads

$$\begin{pmatrix} \dot{q}_1 \\ \dot{p}_1 \\ \dot{q}_2 \\ \dot{p}_2 \\ \dot{q}_3 \\ \dot{p}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1/m & 0 & 0 & 0 & 0 \\ -2k & -2c/m & k & c/m & 0 & 0 \\ 0 & 0 & 0 & 1/m & 0 & 0 \\ k & c/m & -2k & -2c/m & k & c/m \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & k & c/m & -k & -c/m \end{pmatrix} \times \begin{pmatrix} q_1 \\ p_1 \\ q_2 \\ p_2 \\ q_3 \\ p_3 \end{pmatrix} + \begin{pmatrix} 0 \\ k_1 l_1 - k_2 l_2 \\ 0 \\ k_2 l_2 - k_3 l_3 \\ 0 \\ k_3 l_3 \end{pmatrix} + \begin{pmatrix} 0 \\ F_1(t) \\ 0 \\ F_2(t) \\ 0 \\ F_3(t) \end{pmatrix}, \tag{28}$$

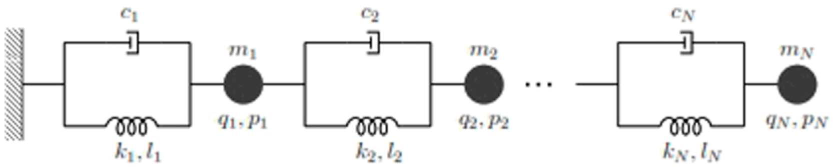

FIGURE 2
N-mass dynamical system.

which, in the linear case and taking into account that $k_1 = k_2 = k_3 = k$ and $l_1 = l_2 = l_3 = l$, after applying Fourier transform, leads to

$$\begin{pmatrix} i\omega & -1/m & 0 & 0 & 0 & 0 \\ 2k & i\omega + 2c/m & -k & -c/m & 0 & 0 \\ 0 & 0 & i\omega & -1/m & 0 & 0 \\ -k & -c/m & 2k & i\omega + 2c/m & -k & -c/m \\ 0 & 0 & 0 & 0 & i\omega & -1/m \\ 0 & 0 & -k & -c/m & k & i\omega + c/m \end{pmatrix} \times \begin{pmatrix} \hat{q}_1 \\ \hat{p}_1 \\ \hat{q}_2 \\ \hat{p}_2 \\ \hat{q}_3 \\ \hat{p}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ kl \end{pmatrix} \delta(\omega) + \begin{pmatrix} 0 \\ \hat{F}_1 \\ 0 \\ \hat{F}_2 \\ 0 \\ \hat{F}_3 \end{pmatrix}, \tag{29}$$

where the *hat* operator, $\hat{\bullet}$, refers to the Fourier transform of the mass positions, momentum, and applied forces.

It is important to note that, in the nonlinear case described later on, since the spring stiffnesses depend on the spring elongation and the latter will obviously be different for each node—unlike here in the linear case— the first vector of the right hand member will contain three non-vanishing spring contributions: $k_1 l_1 - k_2 l_2$, $k_2 l_2 - k_3 l_3$, and $k_3 l_3$.

By reordering the previous system, the position and momentum degrees of freedom can be grouped:

$$\begin{pmatrix} \tilde{\mathbf{T}}_{qq}(\omega) & \mathbf{T}_{qp} \\ \mathbf{T}_{pq} & \tilde{\mathbf{T}}_{pp}(\omega) \end{pmatrix} \left(\begin{array}{c} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hline \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \end{array} \right) = \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ kl \end{array} \right) \delta(\omega) + \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \hat{F}_1 \\ \hat{F}_2 \\ \hat{F}_3 \end{array} \right), \tag{30}$$

and then, the momentum degrees of freedom $\hat{p}_i$ condensed into the ones related to the mass positions $\hat{q}_i$, as previously discussed:

$$\left(\mathbf{T}_{pq} - \tilde{\mathbf{T}}_{pp}(\omega) \mathbf{T}_{qp}^{-1} \tilde{\mathbf{T}}_{qq}(\omega) \right) \mathbb{Z}_q(\omega) = \mathbb{S}_p(\omega). \tag{31}$$

that, after separating the internal and observable degrees of freedom, reads

$$\left(\begin{array}{cc|c} A_{11}(\omega) & A_{12}(\omega) & A_{13}(\omega) \\ A_{21}(\omega) & A_{22}(\omega) & A_{23}(\omega) \\ \hline A_{31}(\omega) & A_{32}(\omega) & A_{33}(\omega) \end{array} \right) \left(\begin{array}{c} \hat{q}_1 \\ \hat{q}_2 \\ \hline \hat{q}_3 \end{array} \right) = \left(\begin{array}{c} 0 \\ 0 \\ \hline kl \end{array} \right) \delta(\omega) + \left(\begin{array}{c} \hat{F}_1 \\ \hat{F}_2 \\ \hline \hat{F}_3 \end{array} \right) \tag{32}$$

which allows making the model involving the observable degree of freedom, $\hat{q}_3$, explicit:

$$\begin{aligned} &\left\{ A_{33}(\omega) - \begin{pmatrix} A_{31}(\omega) & A_{32}(\omega) \end{pmatrix} \right. \\ &\quad \left. \times \begin{pmatrix} A_{11}(\omega) & A_{12}(\omega) \\ A_{21}(\omega) & A_{22}(\omega) \end{pmatrix}^{-1} \begin{pmatrix} A_{13}(\omega) \\ A_{23}(\omega) \end{pmatrix} \right\} \hat{q}_3(\omega) \\ &\quad = kl\delta(\omega) + \hat{F}_3 + \begin{pmatrix} A_{31}(\omega) & A_{32}(\omega) \end{pmatrix} \\ &\quad \times \begin{pmatrix} A_{11}(\omega) & A_{12}(\omega) \\ A_{21}(\omega) & A_{22}(\omega) \end{pmatrix}^{-1} \begin{pmatrix} \hat{F}_1(\omega) \\ \hat{F}_2(\omega) \end{pmatrix}, \end{aligned} \tag{33}$$

that, arranged in a more compact manner, reads

$$\tilde{A}_{33}(\omega) \hat{q}_3(\omega) = kl\delta(\omega) + \hat{F}_3(\omega) + \hat{F}_{i3}(\omega), \tag{34}$$

which represents the system transfer function.

Now, the final point concerns the data-driven model identification, that is, how to extract from the given data the different model components: $\tilde{A}_{33}$ and $\hat{F}_{i3}$, for each involved frequency ω. In the last equation, the index i associated with $\hat{F}_{i3}$ reflects all the effects coming from the unresolved degrees of freedom (internal unobserved masses).

Conceptually, the system identification could proceed as follows:

1. The free response associated with $F_3 = 0$ (only the loads on the internal masses apply), $q_3^f(t)$, is obtained (measured), and the superscript $\bullet^f$ refers to the fact that the observable mass remains free of loading.
2. Then, for a non-null applied (and measurable) loading on the observable mass, $F_3 \neq 0$, the system response $q_3(t)$ is recorded, which now is a consequence of all the loading terms involved in the right-hand member of the previous equation.
3. The difference between the forced and free displacement can be obtained from $\Delta q_3(t) = q_3(t) - q_3^f(t)$, allowing for computation of its Fourier transform $\widehat{\Delta q}_3(\omega)$.
4. Finally, by means of the just calculated $\widehat{\Delta q}_3$ and the Fourier transform of the measurable force $\hat{F}_3(\omega)$, the model coefficient $\tilde{A}_{33}(\omega)$ is learnt from

$$\tilde{A}_{33}(\omega) = \frac{\hat{F}_3(\omega)}{\widehat{\Delta q}_3(\omega)}. \tag{35}$$

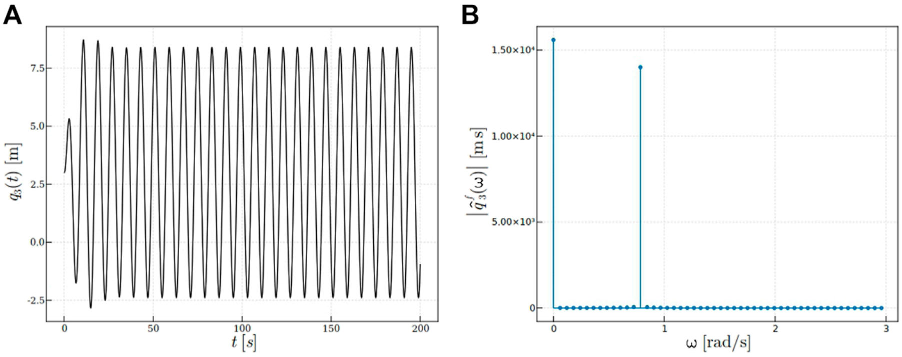

FIGURE 3
Free response ($F_3(t) = 0$): **(A)** $q_3^f(t)$; and **(B)** $\hat{q}_3^f(\omega)$.

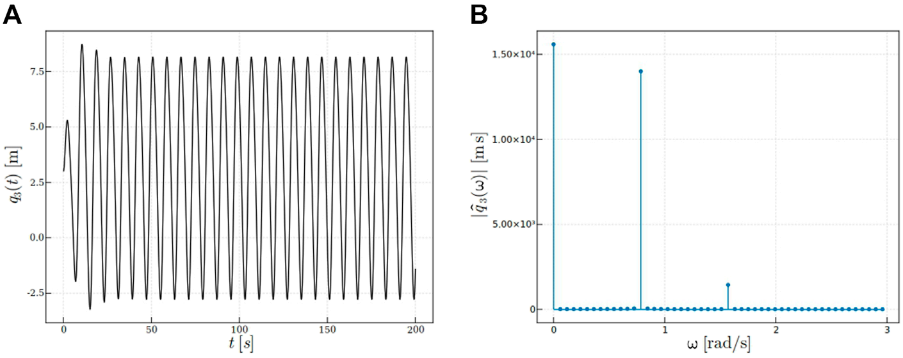

FIGURE 4
Response: **(A)** $q_3(t)$; and **(B)** $\hat{q}_3(\omega)$.

When applying a single-frequency loading, we have:

$$\begin{cases} F_1(t) = 2\cos(2\pi t) \\ F_2(t) = 2\cos\left(\frac{\pi}{4}t\right) \\ F_3(t) = 2\cos\left(\frac{\pi}{2}t\right) \end{cases}, \tag{36}$$

The free and forced responses and their Fourier transforms are depicted, respectively, in Figures 3, 4. This loading is used to generate the synthetic data that will serve to identify the model's output $q_3(t)$ later on as a function of the observed load $F_3(t)$. During the training process of that model, $F_2(t)$ and $F_3(t)$ are fully ignored.

Figure 5 shows the response difference $\Delta q_3(t) = q_3(t) - q_3^f(t)$ and its Fourier transform $\widehat{\Delta q_3}(\omega)$ on the domain in which the difference $\Delta q_3(t)$ becomes almost stabilized, meaning the transient component almost vanishes.

Now, when comparing the reference solution, obtained by the reference analytical model $\tilde{A}_{33} = -1.4848 + 1.0221i$, to the one obtained from the learnt model $\tilde{A}_{33} = -1.5242 + 0.9823i$ (at principal frequency), an excellent accuracy can be noticed.

3.2 rNN and LSTM time simulations in both the linear and the nonlinear settings

In this section, we consider again the 3-mass dynamical system. The dynamical problem is integrated numerically to obtain the ground truth, that is, the reference solution. The computed data will be used for training the different neural networks, the rNN and the LSTM.

In both cases, the input data consist of the force F_3 and position q_3 in the previous time steps, which results in the surrogate $\mathcal{H}$:

$$\tilde{q}_3^i = \mathcal{H}\left(\begin{pmatrix} F_3^i \\ F_3^{i-1} \\ \vdots \\ F_3^{i-n+1} \end{pmatrix}, \begin{pmatrix} q_3^{i-1} \\ q_3^{i-2} \\ \vdots \\ q_3^{i-n} \end{pmatrix}\right), \tag{37}$$

where $\tilde{q}_3^i$ is the prediction of q_3 at time step i.

As Eq. 37 reflects, different memory lengths from the use of the positive integer n, ($n \geq 0$), are taken into account. For $n \neq 0$, an initialization issue occurs.

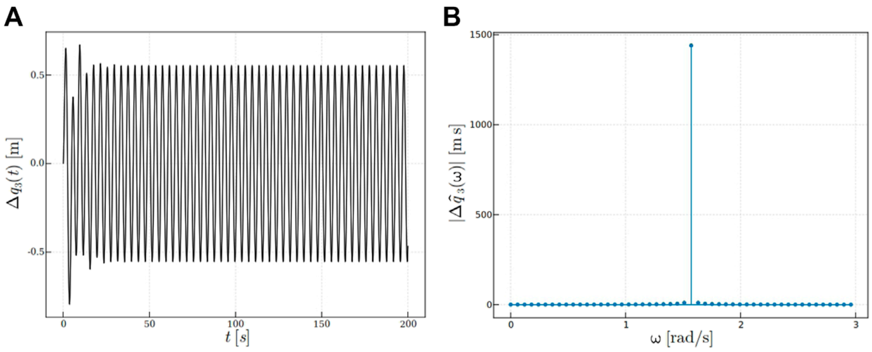

FIGURE 5
Response difference: **(A)** $\Delta q_3(t)$; and **(B)** $\widehat{\Delta q_3}(\omega)$.

In the case considered here, the larger memory is the toll ignoring the internal forces take, whose consequences on the observed variables are learnt from the time evolution of the last.

The initialization can be carried out following two routes:

- If we are interested in the forced regime, the long-time solution does not depend on the initialization.
- Should we prefer obtaining the transient solution, one could consider a coarser model that updates the state from the just previous state until completing the first n values. Then, the LSTM can take over.

In the present study, as previously indicated, we are focused on proving under which conditions a model relating observable inputs and outputs exists, despite the existence of hidden dynamics, resulting in a noticeable larger memory. For that reason, in the simulations considered in the present study, we assumed the first n values known.

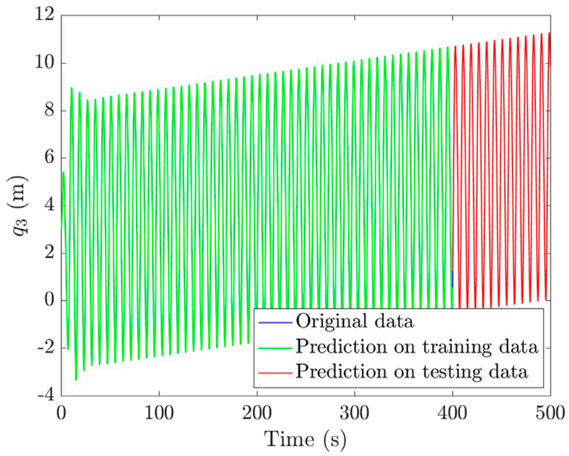

FIGURE 6
Prediction of the observable position $\tilde{q}_3(t)$ computed from a trained rNN with $n = 2$ (colors green and red mark the training and testing sets, respectively). It can be noted that the blue curve is not visible because it is almost exactly under the green and red curves.

3.2.1 Using a simple recurrent neural network

First, we consider a rNN surrogate model with $n = 2$, with respect to **Eq. 37**. The considered data for training come from the integration of the dynamical system, in both the linear and nonlinear cases.

The data consist of 10,000 states of the observable variables (coming as indicated from the standard integration of the dynamical system). These data are divided into two sets, the training and the testing ones, the former containing 80% of the points and the latter the remaining 20%.

The rNN consists of a single layer with one output $\tilde{q}_3^i$, in reference to **Eq. 37**. The network parameters and the initialization choices are the ones reported in Glorot and Bengio (2010). The algorithm is trained during 1,500 epochs, even though the use of fewer epochs leads to similar results.

The linear problem considers, once more: $m_1 = m_2 = m_3 = 0.5Kg$, $c = 0.8\ N/m$, $k_1 = k_2 = k_3 = 1\ N/m$, and $l_1 = l_2 = l_3 = 1$ m, expressing the applied loading the following way:

$$\begin{cases} F_1(t) = 2\cos(2\pi t) \\ F_2(t) = 2\cos\left(\dfrac{\pi}{4}t\right) \\ F_3(t) = \dfrac{t}{t_{max}} + \cos\left(\dfrac{\pi}{2}t\right) \end{cases}, \tag{38}$$

with $t_{max} = 500s$. This loading is used to generate the synthetic data that will serve afterward to identify the model $q_3(t)$ as a function of the observed load $F_3(t)$. During the training process of that model, $F_2(t)$ and $F_3(t)$ are again completely neglected.

The computed results from the trained network are given in **Figure 6**, being the mean absolute percentage errors (MAPE) 1.38% on the training set and 2.18% in the testing set.

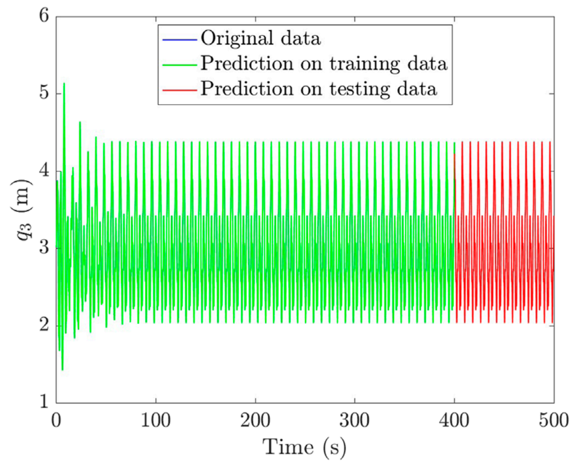

FIGURE 7
Prediction of the observable position $\tilde{q}_3(t)$ in the nonlinear case, computed by a trained rNN with $n = 3$ (again, colors refer to the training and testing sets). It can be noted that the blue curve is not visible because it is almost exactly under the green and red curves.

The same rNN (now with $n = 3$ in reference to Eq. 37) was employed to tackle a nonlinear dynamical system, with similar parameters to the ones considered in the linear case, except in what concerns the stiffnesses of the springs, now given by

$$\begin{cases} k_1 = k_{01}\left(1 + \alpha \Delta l_1\right) \\ k_2 = k_{02}\left(1 + \alpha \Delta l_2\right) \\ k_3 = k_{03}\left(1 + \alpha \Delta l_3\right) \end{cases}, \tag{39}$$

with $k_{01} = k_{02} = k_{03} = 10\ N/kg$ and $\alpha = 10^{-4}\ m^{-1}$ (arbitrary, though carefully tuned to maintain the stability of the simulation) and where $\Delta l_\bullet$ is the elongation of the corresponding spring, i.e., $\Delta l_2 = q_2 - q_1 - f_2$, $\Delta l_3 = q_3 - q_2 - f_3$, and $\Delta l_1 = q_1 - l_1$.

The considered loading reads

$$\begin{cases} F_1(t) = 2\cos(2\pi t) \\ F_2(t) = 2\cos\left(\frac{\pi}{4} t\right) \\ F_3(t) = \cos\left(\frac{\pi}{2} t\right) \end{cases}. \tag{40}$$

The results concerning the nonlinear dynamical system are reported in Figure 7, and, for the sake of clarity, the associated absolute error is reported in Figure 8, with a mean absolute percentage error (MAPE) of 1.34% in the training set and 1.29% in the testing set. The error is slightly larger in the training set, probably due to the larger transient phase presenting higher peaks.

3.2.2 Using an LSTM recurrent neural network

The same linear and nonlinear dynamical systems are now processed by LSTM cells, with the same network parameters and initializations used for the rNN.

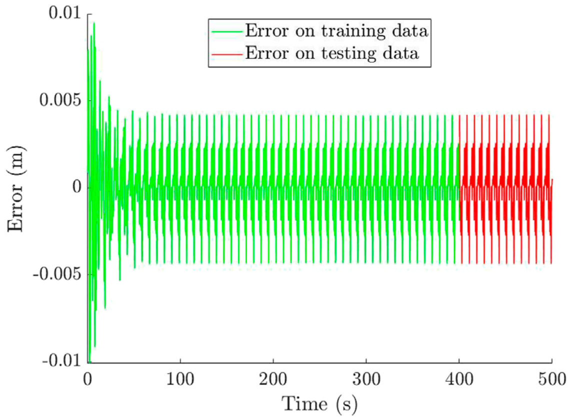

FIGURE 8
Error in the prediction of the observable position $\tilde{q}_3(t)$ in the nonlinear case, computed by a trained rNN with $n = 2$ (the same color code applies).

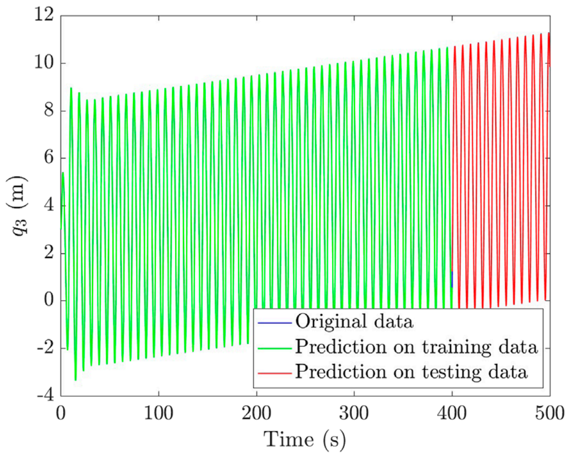

FIGURE 9
Prediction of the observable position $\tilde{q}_3(t)$ computed by a trained LSTM with $n = 2$ (the same color code is employed). It can be noted that the blue curve is not visible because it is almost exactly under the green and red curves.

When addressing the linear case, the computed results are given in Figure 9, with an MAPE of 0.84% in the training set and 1.33% in the testing set. The results in the nonlinear case are reported in Figure 10, and again, for the sake of clarity, the associated absolute error is presented in Figure 11, presenting an MAPE of 0.15% in the training set and 0.14% in the testing set. The error is again slightly larger in the training set for the same reasons given before.

As expected, LSTM outperforms the rNN for a large number of epochs. It was noticed that by reducing the number of epochs, the rNN outperforms LSTMs because convergence is more easily achieved using a lower number of parameters. The error in the

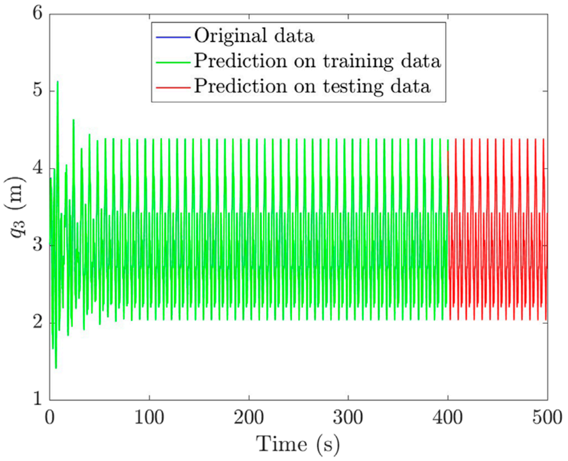

FIGURE 10
Prediction of the observable position $\tilde{q}_3(t)$ in the nonlinear case, computed by a trained LSTM with $n = 3$ (same color code). It can be noted that the blue curve is not visible because it is almost exactly under the green and red curves.

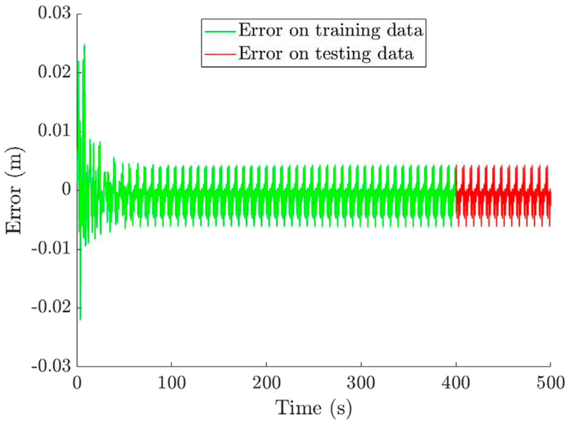

FIGURE 11
Error in the prediction of the observable position $\tilde{q}_3(t)$ in the nonlinear case, computed by a trained LSTM with $n = 3$ (same color code).

linear case was larger, possibly due to the fact that it involves close to zero values which negatively impacts the error calculation.

It must be noted that several experiments with various number of elements, different damping coefficients, stiffnesses, lengths, and masses have been carried out with similarly satisfactory results (MSE error always below 0.07 for both training and testing).

4 Conclusion

The issue related to the circumstances enabling the construction of a model relating the input and output of observed quantities, in the frame of a larger system involving a hidden state that affects the observed variables, was revisited in the present work.

We proved that for time-independent models, such a model exists, and the learnt model is the one where the hidden variables are condensed into the observed ones. As soon as the loading in the internal unresolved degrees of freedom does not change, the computed model can be reused for any other prediction with different loading in the observed region concerned by the learnt model.

In the transient case, the Fourier transform, applicable in the linear case far away from the transient regime, allowed to prove that such a model can be learnt, but in this case, the model involves the recent history of the considered variables (present and recent past) and remains valid for any loading in the observed region, as soon as the forces applied on the hidden part have the same Fourier transforms.

When operating in the time domain, the rNN and LSTM are demonstrably the most natural choices for performing the learning task, and it is also expected that they allow for addressing nonlinear dynamical systems. Regarding time-integration, the performances of both neural networks, rNN and LSTM, remain similar and prove that, as expected from the developments given in Section 2, the model relating observable inputs and outputs can be learnt as soon as past values of them are considered in the model construction.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/cghnatios/LSTM-rNN-for-Modeling-systems-from-partial-observations-.git.

Author contributions

VC contributed to the development of the analytical methodologies, VA, CG, LI-V, and SG performed the implementation of learning procedures, while FM, EC, and FC contributed to the global methodology and research development.

Funding

This research is also part of the DesCartes program and is supported by the National Centre for Scientific Research (CNRS), Prime Minister Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. This study has received funding from the European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 956401 (Project XS-Meta).

Acknowledgments

Authors acknowledge the contribution and support of the ESI-ENSAM research chair CREATE-ID. The support by the ESI Group through the ESI Chair at ENSAM Arts et Métiers Institute of Technology, and through the project 2019-0060 "Simulated Reality" at the University of Zaragoza is also acknowledged. The support of the Spanish Ministry of Science and Innovation, AEI /10.13039/501100011033, through Grant number PID2020-113463RB-C31 and by the Regional Government of Aragón, grant T24-20R, and the European Social Fund, are also gratefully acknowledged.

Conflict of interest

FC was employed by the company CNRS@CREATE LTD.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Argerich, C., Carazo, A., Sainges, O., Petiot, E., Barasinski, A., Piana, M., et al. (2020). Empowering design based on hybrid twin: Application to acoustic resonators. *Designs* 4, 44. doi:10.3390/designs4040044

Benabou, L. (2021). Development of lstm networks for predicting viscoplasticity with effects of deformation, strain rate, and temperature history. *J. Appl. Mech.* 88, 1–30. doi:10.1115/1.4051115

Borzacchiello, D., Aguado, J., and Chinesta, F. (2019). Non-intrusive sparse subspace learning for parametrized problems. *Arch. Comput. Methods Eng.* 26, 303–326. doi:10.1007/s11831-017-9241-4

Bronstein, M. M., Bruna, J., Cohen, T., and Velivckovic̀, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. ArXiv abs/2104.13478.

Brunton, S. L., and Kutz, J. N. (2019). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press. doi:10.1017/9781108380690

Chinesta, F., Ladeveze, P., and Cueto, E. (2011). A short review on model order reduction based on proper generalized decomposition. *Arch. Comput. Methods Eng.* 18, 395–404. doi:10.1007/s11831-011-9064-7

Chinesta, F., Leygue, A., Bordeu, F., Aguado, J. V., Cueto, E., Gonzalez, D., et al. (2013). Pgd-based computational vademecum for efficient design, optimization and control. *Arch. Comput. Methods Eng.* 20, 31–59. doi:10.1007/s11831-013-9080-x

Chinesta, F., Keunings, R., and Leygue, A. (2014). *The proper generalized decomposition for advanced numerical simulations: A primer*. Springer. doi:10.1007/978-3-319-02865-1

Chinesta, F., Huerta, A., Rozza, G., and Willcox, K. (2015). The encyclopedia of computational mechanics. in *Chap. Model order reduction* (John Wiley & Sons), 1–36.

Chinesta, F., Cueto, E., Abisset-Chavanne, E., Duval, J. L., and Khaldi, F. E. (2020). Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. *Arch. Comput. Methods Eng.* 27, 105–134. doi:10.1007/s11831-018-9301-4

Fasel, U., Kutz, J. N., Brunton, B. W., and Brunton, S. L. (2022). Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. Math. Phys. Eng. Sci.* 478, 20210904. doi:10.1098/rspa.2021.0904

Ghanem, R., Soize, C., Mehrez, L., and Aitharaju, V. (2022). Probabilistic learning and updating of a digital twin for composite material systems. *Int. J. Numer. Methods Eng.* 123, 3004–3020. doi:10.1002/nme.6430

Glorot, X., and Bengio, Y. (2010). "Understanding the difficulty of training deep feedforward neural networks,". *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. Editors Y. W. Teh, and M. Titterington (Chia Laguna Resort, Sardinia, Italy: Proceedings of Machine Learning Research), 9, 249–256.

González, D., Chinesta, F., and Cueto, E. (2021). Learning non-markovian physics from data. *J. Comput. Phys.* 428, 109982. doi:10.1016/j.jcp.2020.109982

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press. Available at: http://www.deeplearningbook.org.

Hernandez, Q., Badias, A., Chinesta, F., and Cueto, E. (2022). Thermodynamics-informed graph neural networks. *IEEE Trans. Artif. Intell.*, 1. doi:10.1109/TAI.2022.3179681

Hinton, G., and Zemel, R. (1993). "Autoencoders, minimum description length and helmholtz free energy," in Advances in neural information processing systems, Denver, CO, USA (Morgan-Kaufmann.), 6, 3–10.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735

Ibáñez, R., Abisset-Chavanne, E., Ammar, A., González, D., Cueto, E., Huerta, A., et al. (2018). A multidimensional data-driven sparse identification technique: The sparse proper generalized decomposition. *Complexity* 2018, 1–11. doi:10.1155/2018/5608286

Kapteyn, M. G., and Willcox, K. E. (2020). From physics-based models to predictive digital twins via interpretable machine learning. ArXiv abs/2004.11356. doi:10.48550/ARXIV.2004.11356

Lee, K., and Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* 404, 108973. doi:10.1016/j.jcp.2019.108973

Liu, Z., and Tegmark, M. (2021). Machine learning conservation laws from trajectories. *Phys. Rev. Lett.* 126, 180604. doi:10.1103/PhysRevLett.126.180604

Luo, H., Huang, M., and Zhou, Z. (2018). Integration of multi-Gaussian fitting and lstm neural networks for health monitoring of an automotive suspension component. *J. Sound Vib.* 428, 87–103. doi:10.1016/j.jsv.2018.05.007

Makhzani, A., and Frey, B. J. (2014). k-sparse autoencoders. CoRR abs/1312.5663. doi:10.48550/ARXIV.1312.5663

Manohar, K., Kutz, J. N., and Brunton, S. L. (2018). Optimal sensor and actuator placement using balanced model reduction. ArXiv abs/1812.01574.

Moya, B., Badías, A., Alfaro, I., Chinesta, F., and Cueto, E. (2022b). Digital twins that learn and correct themselves. *Int. J. Numer. Methods Eng.* 123, 3034–3044. doi:10.1002/nme.6535

Moya, B., Badias, A., González, D., Chinesta, F., and Cueto, E. (2022a). Physics-informed reinforcement learning for perception and reasoning about fluids. ArXiv abs/2203.05775. doi:10.48550/ARXIV.2203.05775

Ng, A. (2011). *Sparse autoencoder. CS294A Lecture notes. Tech. rep.* Stanford university.

Sancarlos, A., Cameron, M., Abel, A., Cueto, E., Duval, J.-L., and Chinesta, F. (2021a). From rom of electrochemistry to ai-based battery digital and hybrid twin. *Arch. Comput. Methods Eng.* 28, 979–1015. doi:10.1007/s11831-020-09404-6

Sancarlos, A., Cameron, M., Le Peuvedic, J.-M., Groulier, J., Duval, J.-L., Cueto, E., et al. (2021b). Learning stable reduced-order models for hybrid twins. *Data-Centric Eng.* 2, e10. doi:10.1017/dce.2021.16

Sancarlos, A., Champaney, V., Duval, J. L., Cueto, E., and Chinesta, F. (2021c). Pgd-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit. ArXiv abs/2103.05358.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Netw.* 61, 85–117. doi:10.1016/j.neunet.2014.09.003

Tuegel, E. J., Ingraffea, A. R., Eason, T. G., and Spottswood, S. M. (2011). Reengineering aircraft structural life prediction using a digital twin. *Int. J. Aerosp. Eng.* 2011, 1–14. doi:10.1155/2011/154798

Venkatesan, R., and Li, B. (2017). *Convolutional neural networks in visual computing: A concise guide*. London, United Kingdom: CRC Press.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *J. Big Data* 3, 9. doi:10.1186/s40537-016-0043-6

Williams, J., Zahn, O., and Kutz, J. N. (2022). Data-driven sensor placement with shallow decoder networks. arXiv. doi:10.48550/ARXIV.2202.05330

Zhou, G.-B., Wu, J., Zhang, C.-L., and Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks. *Int. J. Autom. Comput.* 13, 226–234. doi:10.1007/s11633-016-1006-2