Mechanical Systems and Signal Processing 233 (2025) 112767

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

State estimation in structural dynamics through RNN transfer learning

Shuo Hao, Hong-Wei Li *, Yi-Qing Ni, Weijia Zhang, Lei Yuan

Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
National Rail Transit Electrification and Automation Engineering Technology Research Center (Hong Kong Branch), Hong Kong, China

ARTICLE INFO

Communicated by D. Bernal

Dataset link: TL-RNN-StatePrediction-StructDyn

Keywords:
State estimation
Recurrent neural network (RNN)
Transfer learning
Structural dynamics
Physics-data fusion

ABSTRACT

Model construction for state estimation is a pivotal concern in structural dynamics, driven by the need for effective control and health monitoring of structures. In general, state estimation models are derived from physics-based finite element (FE) models. However, the limited capability of FE models to simulate actual structures and the complexity of the environments in which these structures operate pose a significant challenge to the accuracy of model-based state estimation. This paper proposes a novel approach that leverages recurrent neural network (RNN)-based transfer learning to construct state estimation models, aiming to enhance the accuracy of state estimation for actual structures even if the FE model is not accurate enough. A calibrated FE model is used to generate extensive response data under synthetic excitations. The data is then processed and integrated to train an RNN model specifically designed for state estimation. Considering the diverse sensors involved in real-world structure monitoring, this study innovatively utilizes the collected data in a dual-purpose manner. A portion of the data serves as input for the RNN model, while the complete dataset facilitates the transfer learning process for the RNN model. This strategy enables the RNN model to adapt to real-structure state prediction. To ensure effective convergence in transfer learning, a method is proposed where parameters within the RNN cells at the network's front end are fine-tuned, while those close to the output layers are frozen. This approach deviates from conventional transfer learning methods used for other neural networks, while it is particularly useful for RNN models designed for state estimation. Numerical and experimental studies demonstrate that the proposed RNN transfer learning approach could effectively integrate both model-generated and actual measurement data. Under the same data acquisition condition, the transfer learning-RNN models could achieve significantly higher accuracy than state estimation models that rely solely on FE models.

1. Introduction

Accurate state information (e.g., displacement, velocity) of structural systems is crucial for various structural control and health monitoring applications, such as the digital twin construction [1], virtual sensing technology [2,3], active vibration control of structures [4], and structural damage detection [5]. However, obtaining comprehensive measurements of states for real-world structures often presents significant challenges. Deploying a large number of transducers across numerous degrees of freedom (DOFs)

[*] Corresponding author at: Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China.

E-mail address: hong-wei.li@polyu.edu.hk (H.-W. Li).

https://doi.org/10.1016/j.ymssp.2025.112767
Received 24 July 2024; Received in revised form 29 November 2024; Accepted 15 April 2025

Available online 26 April 2025

can be impractical due to high costs, extensive space requirements, and increased risk of damage. As a result, the problem of state estimation, inferring a structure's complete state from limited measurements, has attracted considerable interest in the field of structural dynamics.

Most prior studies on state estimation have primarily utilized various models derived from the principle of Kalman filtering. By incorporating the equations of motion of the finite element (FE) model and embedding priors on external unknown forces and measurements, the constructed Kalman filter model can recursively output the complete system state using noisy measurements of system response. Early work focused on simplified linear structural systems. Papadimitriou et al. [5] employed the standard Kalman filter, which uses accelerations from sparsely distributed accelerometers to estimate the full state of the system, enabling the estimation of strain at locations where no measurements were taken. Lourens et al. [6] developed the augmented Kalman filter (AKF) for joint input-state estimation, where unknown forces are incorporated into the state vector. This approach allows for more effective prior assumptions regarding stochastic loads, measurement, and modeling errors compared to the standard Kalman filter solution. Consequently, the AKF can estimate the augmented vector, including both forces and states. Furthermore, Eftekhar Azam et al. [7] introduced a dual implementation of the Kalman filter algorithm, known as the dual Kalman filter (DKF), for full-state estimation. The DKF employs a sequential structure of two Kalman filters, which circumvents the numerical issues related to unobservability and rank deficiency encountered in the AKF. As a result, the DKF yields more precise state estimation results. Recent efforts have focused on developing state estimation algorithms for nonlinear structures, such as the extended Kalman filter, unscented Kalman filter, and ensemble Kalman filter approaches, as well as various variants of these filters dedicated to the corresponding nonlinear structures [8–11].

In recent years, deep learning models have been increasingly explored and applied to state estimation problems, complementing the mainstream use of Kalman filters. Unlike Kalman filter-based models, which rely on governing equations, deep learning models are developed through the training and optimization of parameters, e.g., weights and biases. The training and test datasets are generated from the responses of an FE model subjected to numerous (if necessary) synthetic excitations. Deep learning-based state estimation models are particularly useful when the FE model cannot be explicitly formulated [12] or when it includes hybrid components functioning as black boxes [13,14]. In such cases, Kalman filter methods become inapplicable, while deep learning models remain valid. Moreover, the deep learning framework for state estimation is effective for both linear and nonlinear structural systems. The performance of a deep learning model is evaluated by the training and test loss; when the training process converges to a relatively low test loss, the predictive performance of the deep learning-based model is considered sufficient for subsequent state estimation tasks. During the initial stages of exploration in this field, researchers seek to utilize the fundamental feedforward artificial neural networks (ANNs) for state estimation, of which input is the combination of sensing data from several steps before. The work by Yang and Lee [15] demonstrated the effectiveness of ANNs, where a four-step delay ANN with a single hidden layer could yield accurate estimates of composite beam tip displacement and velocity within 5% of error. However, for more complicated structures, the accuracy of delayed ANNs might not be guaranteed. With the advancement of computing power and the continuous improvement of deep learning infrastructure, various novel forms of neural networks were adopted for state estimation, e.g., recurrent neural networks (RNNs) [16,17], convolutional neural networks [18], graph neural networks [19], and transformers [20], etc. Although the researches mentioned above are not restricted to structural dynamic systems, these models have all been proven to effectively learn the mapping from monitoring data to the system state. Among all deep learning-based models for state estimation, RNNs stand out for their inherent advantages in the field of structural dynamics, namely their intrinsic capability to model sequence data. Unlike other advanced neural networks that require complex architectural adaptations to handle temporal dependencies, RNNs are inherently designed to process sequences of varying lengths and retain information over time. This ability enables RNNs to capture long-term dependencies and perform real-time processing efficiently, which is crucial for state estimation in structural dynamics. Furthermore, RNNs simplify temporal feature extraction and are supported by a strong theoretical foundation and widespread best practices, enhancing their stability and performance in state estimation tasks, comparable to Kalman filter-based methods. Consequently, RNNs are the primary focus for state estimation in this paper, given these compelling advantages.

Although Kalman filter-based and deep learning-based approaches have made significant advances in state estimation, their application in real-world structures has encountered challenges. These methods, whether based on Kalman filters or deep learning, rely on accurate dynamic models that describe the motion of structures, i.e., FE models. Kalman filter-based methods derive from the equations of motion, while deep learning-based methods are trained on data generated from the FE model. It is crucial to understand that both approaches are fundamentally physics-based models. While some sources might describe deep learning models as data-driven [21,22], in the context of state estimation problems, the training and test datasets originate from physics-based FE models. Consequently, deep learning models are still inherently rooted in physical principles. The primary issue when using these methods for real-world structure state estimation is the limitation of FE models in accurately approximating complex real-world structures. Despite the continuous development of FE technology, which enhances the theoretical accuracy of models through increased model size, a resultant issue is the escalating costs associated with modeling and model calibration. Therefore, continuously expanding the comprehensiveness of FE models may not be a cost-effective strategy for enhancing the predictive accuracy of state estimation models. Alternative approaches that balance model complexity with practical application constraints and computational efficiency warrant further investigation.

Beyond simply optimizing the accuracy of FE models, numerous studies have highlighted the benefits of integrating more information from multiple transducers deployed on the real-world structure [23–26]. For example, Zhu et al. [26] developed a multi-rate Kalman filtering method that combines measured acceleration and displacement data sampled at different frequencies to reconstruct structural responses. The main concept behind this method is that increasing the volume of monitoring data from the actual structure leads to better accuracy in the predicted quantities of interest. However, the primary problem of state estimation caused by discrepancies between FE models and actual structures is still not solved, and sometimes the increment of monitoring data might not compensate for the limitation of the FE model itself.

Enhancing the performance of FE models aims to reinforce state estimation models from a physical perspective, while employing multi-sensor information fusion techniques seeks to enhance the accuracy of predictive results from a data-driven standpoint. Striking a balance between the physical and data-driven strategies is crucial for developing a high-performing state estimation model. This paper introduces an approach using transfer learning of RNN (TL-RNN) to implement the physics-based and data-driven state estimation. Due to the simplicity and flexibility of RNN within the deep learning framework, existing deep learning libraries could easily facilitate the transfer learning (often referred to as fine-tuning) of RNN models. As described, the RNN trained on the FE model is a physics-based model. It is essential to integrate the measured data from the actual structure to ensure the accuracy of this RNN model in practical structural state estimation applications. Therefore, multi-sensor data plays a crucial role in this framework. A portion of the data serves as input for state prediction through the RNN, while the comprehensive sensor data are utilized to fine-tune the RNN model effectively. This proposed approach is feasible in structural dynamics because the states predicted from the initial data portion can be transformed into other observed quantities (e.g., strains, inter-story displacements, etc.), which are included in the second category as illustrated. These transformed quantities contribute to forming a loss function used to fine-tune the RNN. Furthermore, this paper proposes an effective fine-tuning strategy that enables the convergence of RNN transfer learning. This strategy involves freezing the parameters of the fully connected layers near the output and only fine-tuning the parameters within the RNN cells. Therefore, the RNN trained with data from the FE model, which incorporates physical information, and further trained with multi-sensor data through transfer learning, truly achieves a fusion of physics and measured data. The proposed method particularly shows its benefits when inherent discrepancies exist between the FE model and the actual structure, or when the FE model is inaccurate in some sense. A state estimation model derived from an inaccurate FE model will inherently be flawed, regardless of the volume of data collected. However, the transfer learning applied to the RNN allows for corrections to the state estimation model itself, thereby achieving an enhancement in accuracy. Moreover, the input to the RNN model post-transfer learning is less dependent on the current multi-sensor information fusion, as the additional collected data is only used once during the transfer learning. Subsequent predictions using RNN can rely solely on the quantities in the first portion of data for state estimation.

The organization of this paper is as follows: Section 2 lays the foundation by introducing the mathematical framework for state estimation in structural dynamics. Section 3 delves into the methodology of employing TL-RNN for state estimation, offering a detailed procedural understanding. In Sections 4 and 5, the methodologies described in Section 2 are applied to a numerical simulation and an experimental example, respectively. The numerical simulation in Section 4 focuses on a linear shear-type structure, which serves as an approximation to a base-isolated building where hysteresis plays a significant role in state response during seismic excitation. By utilizing limited measurements of inter-story displacement and acceleration, the proposed TL-RNN framework showcases high accuracy in state estimation, outperforming both DKF and AKF approaches that employ the same dataset. Section 5 presents an experimental example involving a two-span, continuously supported beam subjected to stochastic loads under laboratory conditions. This example underscores the significant improvement in predictive accuracy offered by transfer learning when compared to an RNN model trained exclusively on FE models. Section 6 discusses three different approaches to leveraging monitored information for state estimation, emphasizing the innovative solution provided by the TL-RNN framework. Finally, conclusions are drawn in Section 7.

2. Problem statement

Consider an n-DOF structural dynamic system formulated through spatial and temporal discretization of the governing differential equations:

$$g_1\left(\boldsymbol{x}_k, \boldsymbol{f}_k, \boldsymbol{x}_{k+1}; \boldsymbol{\theta}\right) = \mathbf{0}, \tag{1}$$

in which $\boldsymbol{x}_k \in \mathbb{R}^{2n}$ is the state vector at time instant k, containing the displacement and velocity of each DOF of the structure; $\boldsymbol{f}_k \in \mathbb{R}^{n_f}$ represents the external excitation vector; $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ denotes the parameters that completely define the system; n_f and n_p are the numbers of forces and parameters, respectively; $g_1 : \mathbb{R}^{2n} \times \mathbb{R}^{n_f} \times \mathbb{R}^{2n} \to \mathbf{0} \in \mathbb{R}^{2n}$ is the function that governs the dynamical evolution of the state vector. The observation process of this n-DOF structure subjected to external excitation is of the form:

$$g_2\left(\boldsymbol{x}_k, \boldsymbol{f}_k, \boldsymbol{y}_k; \boldsymbol{\theta}\right) = \mathbf{0}, \tag{2}$$

where $\boldsymbol{y}_k \in \mathbb{R}^{n_s}$ is the observation vector involving measurements of limited DOFs, obtained from the sensor array installed on the monitored structure; n_s denotes the number of sensors; $g_2 : \mathbb{R}^{2n} \times \mathbb{R}^{n_f} \times \mathbb{R}^{n_s} \to \mathbf{0} \in \mathbb{R}^{n_s}$ represents the observation function.

State estimation in structural dynamics typically involves developing a model that provides accurate estimates of state vectors $\boldsymbol{x}_k$, denoted as $\tilde{\boldsymbol{x}}_k$, throughout a specified time interval $[0, T]$. This estimation uses a dataset $\mathcal{D} = (t_k, \hat{\boldsymbol{y}}_k)_{k=1}^{N}$, where $t_k \in [0, T]$ is the timestamp corresponding to the measurement $\hat{\boldsymbol{y}}_k \in \mathbb{R}^{n_s}$ over structure, and T is the total duration of the estimation period. The measurement $\hat{\boldsymbol{y}}_k$ corresponds to the observation vector $\boldsymbol{y}_k$ described in Eq. (2) and may include various physical quantities measurable in real-world structures, such as accelerations from limited DOFs and strains at specific locations.

Compared to the states of the structural system, the measurements are often spatially sparse and contaminated with noise, adding complexity to the estimation process. Additionally, the functions g_1 and g_2 that govern the motion behavior of real-world structures, along with the corresponding external excitations, are generally unknown. Traditionally, an idealized FE model is constructed to simulate the motion of structure, providing approximated functions $\tilde{g}_1$ and $\tilde{g}_2$. To proceed with the estimation, prior assumptions about the properties of measurement noise and external excitations must be made. However, the FE model and assumptions could lack accuracy due to modeling simplifications and insufficient knowledge of real-world conditions, which may consequently lead to inaccuracies in state estimation results. Therefore, applying state estimations to actual structures is inherently challenging and requires advanced techniques to achieve precise estimations.

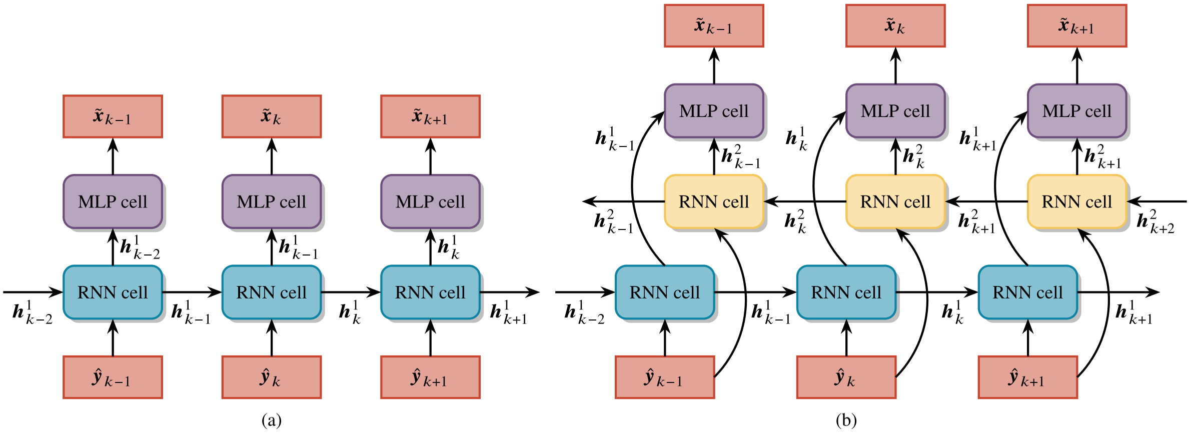

Fig. 1. The general architecture of RNN and BiRNN for state estimation.

3. Methodology

3.1. RNN for state estimation

RNNs constitute a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This architecture allows the RNN to exhibit temporal dynamic behavior and enables the network to act as a form of memory. As a result, RNNs could be exceptionally well-suited for applications in state estimation, where the states of a structural dynamic system are in a state of flux, continuously influenced by their historical states and external excitation. By assimilating the data from practical measurements over a structure, a well-trained RNN is expected to output the states of the system accurately.

In the present study, two general architectures of RNNs for state estimation are investigated, i.e., the vanilla RNN and bidirectional RNN (BiRNN). Although there have been advancements in RNN design, such as the long short-term memory network and the gated recurrent unit, the simulation results indicate that the simplest form of RNN achieves outstanding performance in structural state estimation. Therefore, these more complex RNN structures are not adopted in this paper. Fig. 1 displays the general architectures of RNN and BiRNN. It is pertinent to note that the RNN facilitates real-time state computation using measurement data. In contrast, computations of BiRNN are offline but often yield higher prediction accuracy than RNN by leveraging additional information from future measurements in the data sequence.

In Fig. 1, the RNN cell in the ith layer processes the measurements from the structure, denoted as $\hat{\boldsymbol{y}}_k$, and the previous hidden state, $\boldsymbol{h}^i_{k-1}$, to produce the current hidden state, $\boldsymbol{h}^i_k$. The computation at each time step follows the equation:

$$\boldsymbol{h}^i_k = \tanh\left(\mathbf{W}^i_1 \hat{\boldsymbol{y}}_k + \mathbf{W}^i_2 \boldsymbol{h}^i_{k-1} + \boldsymbol{b}^i\right). \tag{3}$$

Here, k ranges from 1 to N, where N is the total number of time steps in the state estimation process. The matrices $\mathbf{W}^i_1$ and $\mathbf{W}^i_2$ represent the weights respectively for measurements and hidden states, while the vector $\boldsymbol{b}^i$ includes the corresponding biases. In an RNN, the hidden state output from the RNN cell at each step is the direct input to a multilayer perceptron (MLP) cell. In contrast, for a BiRNN, the hidden states from forward and backward RNN cells are concatenated before being input into the MLP cell. The MLP cell, composed of densely connected neurons with nonlinear activation functions, ultimately provides the estimated state at each timestamp. It is suggested to set all bias vectors in both RNN and MLP cells to zero when performing state estimation tasks. This ensures that the RNN model outputs are primarily influenced by the inputs and hidden states, aligning with the expectation of zero dynamic states when measurements are zero.

As a data-driven approach, the training and test datasets for the RNN model are constructed from the vibration responses of an idealized FE model. Both the input features and target outputs for the RNN model correspond to the response variables obtained from the simulations. To create label pairs for the RNN model, it is necessary first to apply synthesized external excitations to the idealized FE model. Subsequently, the calculated structural responses are combined with noise patterns that have been manually determined, which essentially reflect the estimated contamination from factors including measurement and modeling. The optimization of parameters is carried out by minimizing the loss function across all time steps, as demonstrated in Eq. (4), where $\mathcal{L}_1(\cdot)$ denotes the loss function for the first training of RNN. The test loss generated from the test dataset reflects the expected performance of the trained RNN model on the state estimation task.

$$\min \sum_{k=1}^{n} \mathcal{L}_1\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{x}_k\right). \tag{4}$$

It is important to note that the artificial specification of excitation and noise in the RNN responses is similar to the prior assumptions made in Kalman filter-based methods. Ultimately, these specified features are incorporated into the parameters of either the RNN or Kalman filter models. The key difference lies in the number of parameters: Kalman filter models involve few, albeit statistically meaningful, parameters, whereas RNNs have thousands of parameters that are not individually interpretable. This discrepancy arises because Kalman filtering is derived based on rigorous but idealized assumptions; as long as real-world conditions adhere to these simplified assumptions, Kalman filtering performs well. In contrast, the construction of RNNs allows these assumptions to be more complex, such as by specifying non-stationary external excitations or more intricate noise models. The inherent complexity of RNN models grants them greater flexibility, potentially enabling more accurate state estimation under complex conditions. However, this flexibility also introduces a common criticism in the field of deep learning: the performance of RNNs cannot be mathematically guaranteed and must be validated empirically using test datasets, which is considered acceptable in practice.

3.2. Transfer learning of RNN

All models in structural dynamics are simplifications of actual structures they attempt to represent. These approximations, by their nature, cannot capture every nuance of their real-world counterparts. The inaccuracies can be particularly stark when structures are subjected to extreme operational conditions. The divergence between the model predictions and actual behavior is often due to a range of factors, including but not limited to, oversimplified assumptions, parameter uncertainties, ignored physical mechanisms, and the constraints of computational techniques [27]. Even though the calibration of the FE model through sensitivity analysis, uncertainty quantification, and model updating techniques improve its reliability [28,29], discrepancies with the actual structure could not be thoroughly eliminated. Moreover, the accuracy of state estimation in real-world structures relies heavily on the fidelity of the FE models. Since the mathematical model for state estimation is typically derived from the equations of motion or trained using datasets created from the FE model, any discrepancies in the simplified model can significantly impact the precision of state estimations. In situations where the structure experiences extreme conditions which are not accounted for in the idealized FE model, the errors in state estimation are likely to be amplified. Consequently, the reliability of the state estimation is inherently linked to the representativeness of the idealized FE model: an imprecise idealized FE model can lead to poor state estimation outcomes.

Acknowledging the impracticality of creating an exhaustive model for complex structures due to computational and modeling constraints, this paper advances the field by proposing an approach referred to as TL-RNN that harnesses transfer learning in conjunction with RNN models for state estimation. Transfer learning leverages the power of pre-trained models by applying knowledge acquired from previous tasks to new, similar tasks [30]. This approach significantly reduces the need for extensive data typically required to achieve high performance in new learning tasks, addressing the data-hungry problem ubiquitous in deep learning.

In the context of this study, the pre-trained RNN is initially trained using data from an FE model. The subsequent transfer learning task involves improving this model using data from the actual structure. In practice, generating full input–output label pairs for the RNN based on the actual structure is challenging, as it is usually infeasible to observe all states of the structure. Moreover, the volume of observation data available from the actual structure is substantially less than what can be simulated using the idealized FE model, thereby providing an insufficient basis for training an accurate RNN model for state estimation. A practical strategy is that beyond observing the first portion of data (e.g., accelerations) that would be used as input to RNN, measurements from other transducers on the structure together with the acceleration data would be useful to fine-tune the pre-trained RNN model. For example, in multi-story buildings, various techniques can be implemented to effectively measure inter-story displacements and velocities [31]. Similarly, for certain structural elements, strain measurements are readily available. These measurements are compiled into a vector, denoted as $\hat{\mathbf{z}}_k \in \mathbb{R}^{n_e}$ at step k, where n_e denotes the number of extra transducers rather than those used for $\hat{\mathbf{y}}_k$. The measurement vector $\hat{\mathbf{z}}_k$ is intrinsically linked to the structural states predicted by the RNN. It can be derived from the state vector through a specific linear transformation, represented here by the matrix $\mathbf{B} \in \mathbb{R}^{n_e \times 2n}$. As a result, a novel loss function can be introduced, minimizing the discrepancies between $\mathbf{B}\tilde{\mathbf{x}}_k$ and $\hat{\mathbf{z}}_k$, as depicted in Eq. (5):

$$\min \sum_{k=1}^{n} \mathcal{L}_2\left(\mathbf{B}\tilde{\mathbf{x}}_k, \hat{\mathbf{z}}_k\right), \tag{5}$$

where $\mathcal{L}_2(\cdot)$ represents the specified loss function for transfer learning. The minimization process aligns the pre-trained RNN model, derived from the idealized model, with the high-fidelity data of the actual structure, thus effectively fine-tuning the model to improve state estimation accuracy.

To preserve the knowledge learned from the idealized model, as shown in Fig. 2, the parameters in the MLP layers are kept constant, i.e., the MLP layers are "frozen". The re-training process is focused exclusively on the parameters in the RNN cells, utilizing the new loss function as described in Eq. (5). This selective training approach contrasts with that commonly employed in transfer learning for convolutional neural networks, where it is standard practice to re-train the final few layers while the initial layers, which capture more generic features, remain unchanged. This distinction highlights a tailored transfer learning strategy for RNN in the context of state estimation in structure dynamics. By re-training only the RNN cells, the model is fine-tuned to capture the temporal dynamics characteristic of the actual structure, while the knowledge embedded in the MLP layers, likely representing more universal aspects of structural behavior, is retained without modification. This allows the TL-RNN to adapt to the specificities of the field-measured data from the actual structure without needing to re-learn fundamental patterns and relationships already established during the initial training phase on the idealized FE model. Moreover, from the practice of optimization based on Eq. (5), the execution of freezing MLP layers and re-training RNN cells leads to the convergence of the algorithm.

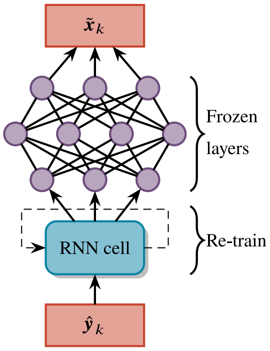

Fig. 2. Transfer learning of RNN.

3.3. State estimation procedures using TL-RNN

The workflow of the proposed TL-RNN is shown in Fig. 3. For the sake of convenience, the full monitoring data are divided into accelerations and additional measurements, with accelerations serving as the primary information for state estimation. The major steps are summarized as follows:

Step 1: Idealized model construction. Construct an idealized, simplified parametric FE model to approximate the dynamic of the actual structure. Incorporate existing information to preliminarily determine the model parameters as accurately as possible.

Step 2: System identification and model updating. Calculate the vibrational characteristics of the actual structure through field-monitoring data. Refine the parameters of the idealized FE model to align the model results with the measurement, thereby further reducing the discrepancies between the model and the actual structure.

Step 3: Data preparation. Delineate a multitude of external excitation time histories as input for the calibrated FE model. Compute the resultant responses to these excitations and incorporate corresponding noise into the responses. For each timestamp, the acceleration responses of designated DOFs are extracted from the global responses. These selected acceleration responses are utilized as input to the RNN, while the system's state vector is designated as the output. These corresponding pairs of input and output are methodically compiled to constitute the training and test datasets.

Step 4: RNN construction, training, and validation. Build the RNN or BiRNN model based on the general architecture shown in Fig. 1. Begin by selecting suitable hyperparameters, which should include, but are not limited to, the learning rate, number of training epochs, type of loss function and optimizer, regularization techniques and associated parameters, dimension of hidden state in the RNN cell, and the configuration of the MLP cell. Then, train the RNN with various hyperparameter combinations to minimize the training loss. Concurrently, monitor the RNN performance on the test dataset to guard against overfitting. The aim is to identify the set of hyperparameters that results in the minimum loss on the test dataset after training has converged. This optimal hyperparameter set should be considered as the one that enhances the model's generalization ability on unseen state estimation tasks.

Step 5: Transfer learning of RNN. Divide the multi-sensor measurements into two sets at each time step. For example, if there are four strain gauges on the structure, one set (the training dataset) includes the measurements from three of these gauges, while the second set (the test dataset) contains data from the remaining gauge. The corresponding acceleration data, which acts as the input to the RNN, is prepared in alignment with the strain measurements of both datasets. Then, a loss function is formulated for both datasets according to Eq. (5). The pre-trained RNN is fine-tuned on the training dataset following the regulation as shown in Fig. 2, during which the hyper-parameters are adjusted. The RNN model that achieves the best performance on the test dataset is chosen as the fine-tuned model. This model is expected to fit well to measured data while retaining the essential knowledge from the pre-training phase and is then deployed for state estimation on the actual structure. The fine-tuned model should now be capable of making accurate predictions of the structural states based on the input acceleration data, informed by the multi-sensor measurements.

4. Numerical example

4.1. Base-isolated building structure and simplified model

A 13-story base-isolated shear-type building shown in Fig. 4(a) is utilized to represent the high-fidelity real-world structure, where the superstructure is simulated as a linear shear-type structure, and New Zealand (N-Z) bearings are adopted between the

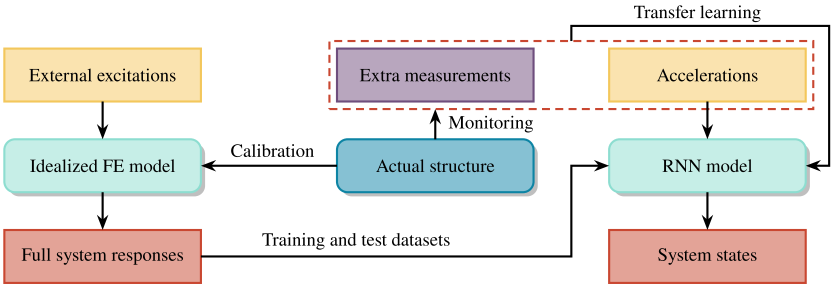

Fig. 3. Workflow of the RNN transfer learning for state estimation.

Table 1
Parameters for defining the base-isolated building structure.

Parameter	Symbol	Value	Unit	Parameter	Symbol	Value	Unit
Mass of 1st∼13th story	$m_1 \sim m_{13}$	1.20×10^5	kg	Yield displacement	q	0.01	m
Stiffness of 1st story	k_1	1.56×10^5	kN/m	Yield strength	F_y	1.20×10^3	kN
Stiffness of 2nd∼4th story	$k_2 \sim k_4$	1.44×10^5	kN/m	Ratio of post- to pre-yield stiffness	α	0.70	–
Stiffness of 5th∼9th story	$k_5 \sim k_9$	9.60×10^4	kN/m	Bearing dimensionless quantity	β	0.50	–
Stiffness of 10th∼13th story	$k_{10} \sim k_{13}$	6.00×10^4	kN/m	Bearing dimensionless quantity	γ	0.50	–
Damping factor of 1st∼4th story	$c_1 \sim c_4$	3.60×10^2	kNs/m	Bearing dimensionless quantity	n	2	–
Damping factor of 5th∼9th story	$c_5 \sim c_9$	2.88×10^2	kNs/m	Bearing dimensionless quantity	A	1.00	–
Damping factor of 10th∼13th story	$c_{10} \sim c_{13}$	1.80×10^2	kNs/m				

foundation and superstructure [32]. As shown in Fig. 4(b), the dominant feature of the N-Z bearing is the parallel action of a linear spring, damper, and hysterics. Therefore, the Bouc–Wen model is well-suited to characterize the force–deformation relationship [33]. The restoring force F_b provided by the N-Z bearing is given by:

$$F_b = \alpha k_1 u_1 + c_1 \dot{u}_1 + (1-\alpha) F_y \nu, \tag{6}$$

where F_y is the yield strength of the bearing; u_1 and $\dot{u}_1$ are displacement and velocity of the base floor, respectively; α denotes the ratio of post- to pre-yielding stiffness; k_1 and c_1 represent the stiffness and viscous damping factor of the base story, respectively; ν is the dimensionless hysteric displacement component that satisfies the following nonlinear first order differential equation:

$$q\dot{\nu} = A\dot{u}_1 + \beta|\dot{u}_1||\nu|^{n-1}\nu - \gamma \dot{u}_1 |\nu|^n, \tag{7}$$

where q denotes the yield displacement; A, β, γ, and n are dimensionless quantities, among which n must be a positive integer to control the smoothness of transition from elastic to plastic response.

The equations of motion for the 13-story base-isolated structure subjected to external excitation are given by:

$$\begin{aligned} &\mathbf{M}_s \ddot{\boldsymbol{u}}_s + \mathbf{C}_s \dot{\boldsymbol{u}}_s + \mathbf{K}_s \boldsymbol{u}_s = -\mathbf{M}_s \boldsymbol{r} \ddot{u}_1 + \boldsymbol{f}_s, \\ &m_1 \ddot{u}_1 + F_b - c_2 \dot{u}_2 - k_2 u_2 = f_b, \end{aligned} \tag{8}$$

where $\mathbf{M}_s$, $\mathbf{C}_s$, and $\mathbf{K}_s \in \mathbb{R}^{12\times 12}$ are mass, damping, and stiffness matrices of the superstructure, respectively; $\boldsymbol{u}_s$, $\dot{\boldsymbol{u}}_s$, and $\ddot{\boldsymbol{u}}_s \in \mathbb{R}^{12}$ are relative floor displacement, velocity, and acceleration vectors with respect to the base story, respectively; $\boldsymbol{r} \in \mathbb{R}^{12}$ is a vector with all elements equal to 1; $\boldsymbol{f}_s \in \mathbb{R}^{12}$ is the external excitation vector, and $f_b \in \mathbb{R}$ denotes the external load applied to the base story. The complete parameters that define the base-isolated structural dynamic system are listed in Table 1 by referring to [32,34]. The governing equations in Eq. (8) could be explicitly formulated with the determination of these parameters. Upon defining the time-varying external forces acting on the base-isolated building, the responses of the whole system are numerically calculated step by step based on the Newmark-β method ($\beta_{\text{Newmark}} = 1/6$, $\gamma_{\text{Newmark}} = 1/2$).

The linear shear-type structure depicted in Fig. 4(c) serves as a simplified model to approximate the vibration of the base-isolated building shown in Fig. 4(a). Each floor of the simplified model is assigned a mass equivalent to its base-isolated building counterpart. However, replication of the base-isolated building's stiffness and damping characteristics necessitates a process of system identification and subsequent model refinement. Despite these efforts to calibrate the parameters of the simplified model, discrepancies between the model and the base-isolated building are inevitable. Consequently, reliance on the simplified model for state estimation is likely to introduce a measure of inaccuracy.

4.2. System identification and model updating from ambient vibration

Stationary white Gaussian noise processes as external excitation are applied to each story of the base-isolated building to simulate the ambient vibration. These processes are synthesized by first establishing one-sided, flat power spectral density with a constant value of 1.44×10^4 N^2/Hz ranging from 0 to 10 Hz, then the time histories are generated by applying the inverse Fourier transform

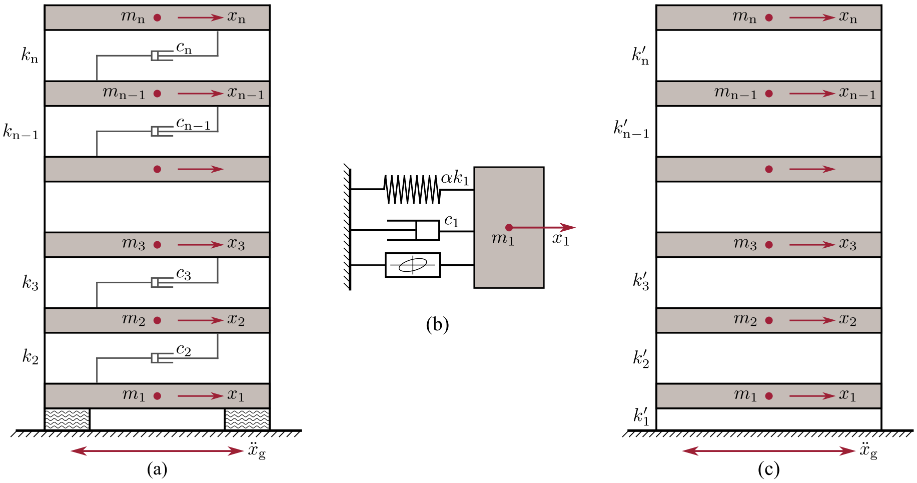

Fig. 4. A base-isolated building structure and simplified shear-type structure.

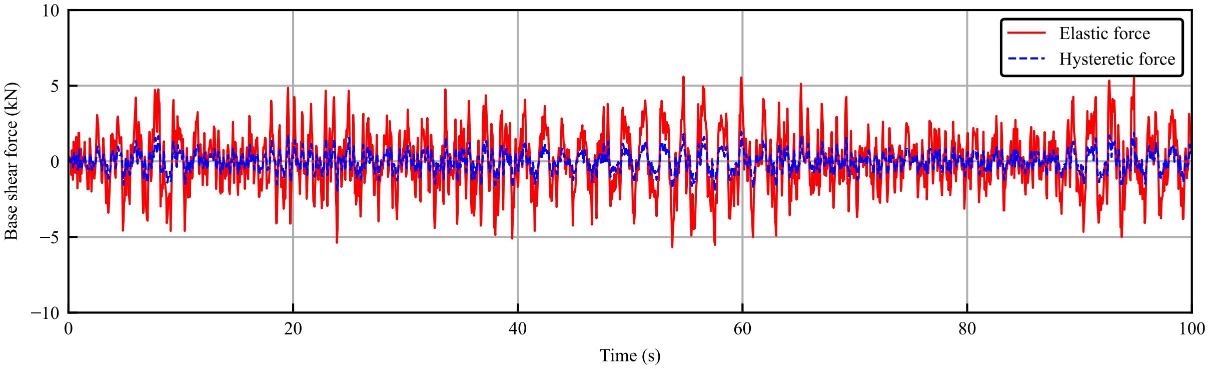

Fig. 5. Comparison of elastic force and hysteretic force from the bearing under ambient vibration.

on the spectral components derived from the flat power spectral density function. The sampling frequency and total duration of the white Gaussian noise process are 20 Hz and 1000 s, respectively. The system responses of the base-isolated building are computed accordingly with the excitation input to the governing equation in Eq. (8). It is worth noting that although simulated ambient excitation could only result in the relatively low amplitude of structure vibration, e.g., the peak acceleration at the top story during the whole loading process is merely 0.065 $\mathrm{m/s^2}$, the hysteresis from the bearing still occupy a certain role in contributing the base shear force. Fig. 5 presents a comparison between the elastic force and hysteretic force in the first 100 s, indicating that the hysteretic force is not negligible even under the ambient vibration condition. Thus, there could be certain deviations between the calibrated shear-type structure and the base-isolated building.

The output-only system identification is virtually performed on the base-isolated building. This process involves equipping each story with an accelerometer to capture the lateral acceleration. As a result, ambient vibration data for 1000 s from every floor is collected. The collected data undergoes frequency domain decomposition [35] to extract the building's modal properties including modal frequency, mode shape, and damping ratio for each vibration mode. Fig. 6 presents the various orders of singular values, where the black line denotes the predominant singular value at every frequency. Modal frequencies and damping ratios are estimated by curve fitting the single DOF spectral bell over the black line. Mode shapes are obtained from singular vectors corresponding to the peaks in the bells. The identified first five modal frequencies and mode shapes are presented in Fig. 7(b) and (c), and the corresponding damping ratios are respectively 2.14%, 2.25%, 3.05%, 4.51%, and 4.41%.

The linear shear-type structure's stiffness parameters are refined by aligning them with the modal frequencies and mode shapes obtained from measurements. To achieve the closest possible correlation between the simplified shear-type structure and the base-isolated building, a goodness-of-fit function $J(\boldsymbol{\tau})$ is minimized [36], which is given by:

$$J(\boldsymbol{\tau}) = \sum_{i=1}^{r}\left[\left(\frac{\hat{f}_i - f_i(\boldsymbol{\tau})}{\hat{f}_i}\right)^2 + \left(1 - \frac{|\hat{\boldsymbol{\phi}}_i^{\mathsf{T}}\boldsymbol{\phi}_i(\boldsymbol{\tau})|^2}{\left(\hat{\boldsymbol{\phi}}_i^{\mathsf{T}}\hat{\boldsymbol{\phi}}_i\right)\left(\boldsymbol{\phi}_i^{\mathsf{T}}(\boldsymbol{\tau})\boldsymbol{\phi}_i(\boldsymbol{\tau})\right)}\right)\right], \tag{9}$$

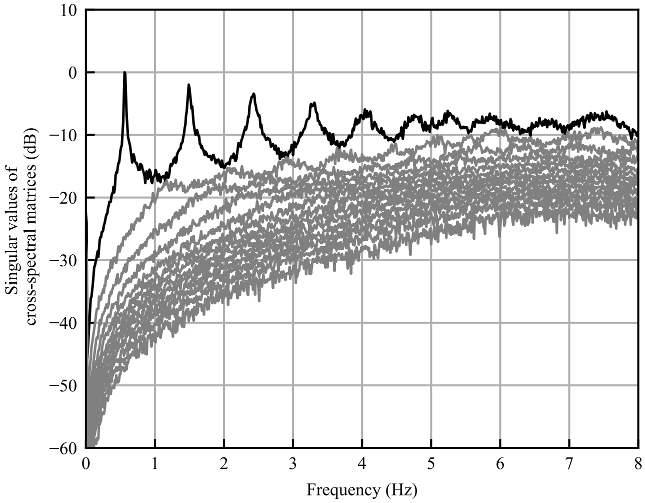

Fig. 6. Singular values of cross-spectral matrices.

where $r = 5$ is the total order of modal information used for model updating; $\hat{f}_i$ and $\hat{\boldsymbol{\phi}}_i$ are the measured ith modal frequency and mode shape, respectively; $f_i(\boldsymbol{\tau})$ and $\boldsymbol{\phi}_i(\boldsymbol{\tau})$ are the corresponding model prediction results under the stiffness parameter set $\boldsymbol{\tau}$. The optimization process employs the standardized L-BFGS-B algorithm [37], a limited-memory variant of the Broyden–Fletcher–Goldfarb–Shanno method. Initial values of stiffness parameters for the optimization are based on the parameters $k_1 \sim k_{13}$ listed in Table 1. The optimal stiffness parameters are displayed in Fig. 7(a). Although at the model level, the shear-type structure and the base-isolated structure differ only in the bearings at the base, this distinction is manifested in the global stiffness parameters identified during the parameter identification process. The updated parameters are observed to be slightly lower than the true parameters. This discrepancy can be attributed to the presence of the hysteretic behavior in the bearings of the base-isolated structure, which absorbs a portion of the total energy. Consequently, the amount of elastic energy is reduced, leading to a decrease in the stiffness of the base-isolated structure when subjected to dynamic loads. Fig. 7(b) and (c) illustrate the comparisons of modal frequencies and mode shapes. The red patterns represent those calculated from the updated linear shear-type structure model, while the blue patterns are obtained from the measured acceleration data on the base-isolated building. The differences between the predicted and measured modal frequencies are all smaller than 3%, and the correlation between the predicted and measured mode shapes are all higher than 99.98%. Therefore, the updated simplified model demonstrates a high degree of accuracy in replicating the dynamic behavior of the base-isolated structure.

For the damping characteristics of the simplified model, the Rayleigh damping approach is used. In this method, the damping matrix for the shear-type structural system is derived from a linear combination of the mass matrix and the updated stiffness matrix. The Rayleigh damping coefficients, $\alpha_{\mathbf{M}}$ for mass and $\beta_{\mathbf{K}}$ for stiffness, are determined using the equations below:

$$\begin{aligned} \alpha_{\mathbf{M}} &= 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}, \\ \beta_{\mathbf{K}} &= \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi\left(f_2^2 - f_1^2\right)}, \end{aligned} \tag{10}$$

where ζ_1 and ζ_2 are the damping ratios for the first two modes of vibration, while f_1 and f_2 are the corresponding measured modal frequencies in Hz.

4.3. Pre-training of RNN based on the calibrated shear-type structure model

Training and test datasets for the RNN and BiRNN are created based on the calibrated shear-type structure model. To begin, 100 robust ground acceleration time histories are synthesized to act as external excitations for the shear-type structure. Earthquake-induced ground acceleration is inherently a non-stationary process. This research synthesizes the time histories by combining stationary white Gaussian noise processes with a time-varying envelope function. These stationary processes stem from a one-side flat power spectral density from 0 to 10 Hz with random phases, set to a constant value of 0.003 g^2/Hz, where g is equivalent to 9.8 $\mathrm{m/s}^2$. This setting ensures that the generated random excitations have a standard deviation of approximately 0.17 g, aligning with the procedure in Kalman filter-based methods where a prior assumption about the external excitations must be determined. Each stationary process is sampled at a frequency of 20 Hz and spans a total duration of 40 s. The envelope function $I(t)$ that imparts

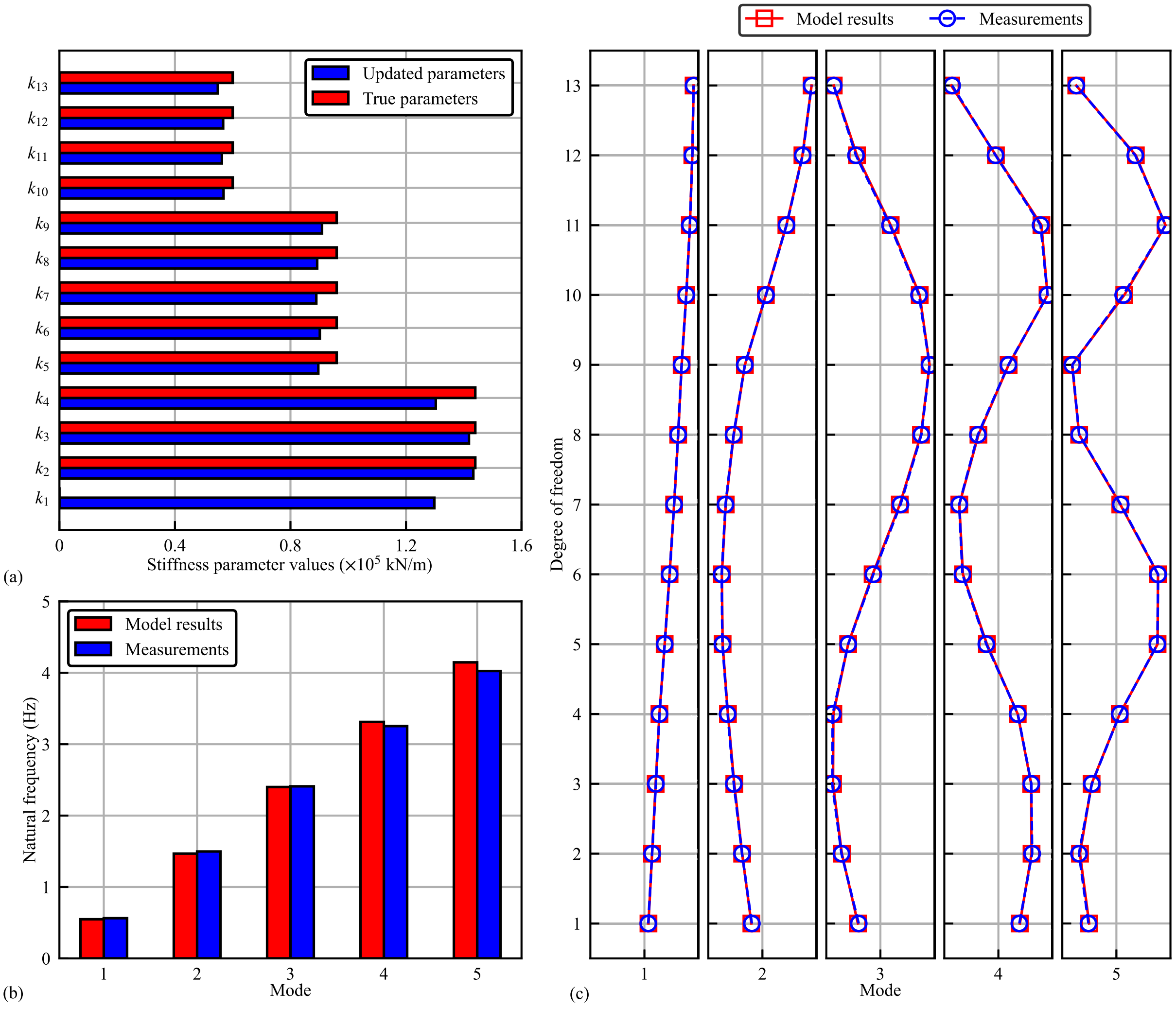

Fig. 7. Optimal stiffness parameter values and comparison of model results with the measurements.

non-stationarity is given by the following equation:

$$I(t) = \begin{cases} t/\left(b_1 t_{\max}\right) & t/t_{\max} \le b_1 \\ 1 & b_1 < t/t_{\max} \le b_2 \\ \exp\left[-\kappa\left(t/t_{\max} - b_2\right)\right] & t/t_{\max} > b_2 \end{cases}, \tag{11}$$

where b_1, b_2, and κ are parameters drawn from uniform distributions with $b_1 \in [0.1, 0.2]$, $b_2 \in [0.4, 0.6]$, and $\kappa \in [3, 5]$. The variable $t_{\max}$ represents the total duration of each time history, i.e., 40 s. Fig. 8(a) illustrates a representative synthesized ground acceleration alongside the corresponding window function. Fig. 8(b) displays the continuous wavelet transform results of a sample, confirming the non-stationary nature of the synthesized time history.

The acceleration, velocity, and displacement responses of the idealized model subjected to 100 synthesized ground motions are calculated and retained. Assuming that accelerometers are mounted on stories 1–5, acceleration responses from these stories are extracted from the overall response. The strong observability of the system under this accelerometer deployment scheme is established using the procedures illustrated in [38], thereby justifying its rationality. The selected acceleration responses serve as input data for RNN and BiRNN, with the system state vector defined as the output target. These input–output pairs are compiled to create the training and test datasets. Specifically, the training dataset is composed of 90 batches of sequences, while the test dataset includes the remaining 10 batches.

RNN and BiRNN models are established according to the architecture shown in Fig. 1. Multiple training iterations are performed for each model to determine the optimal hyper-parameters that yield the best test dataset performance. The mean square error function is employed as the loss function for both models, with the Adam optimizer used for optimization. The hidden state

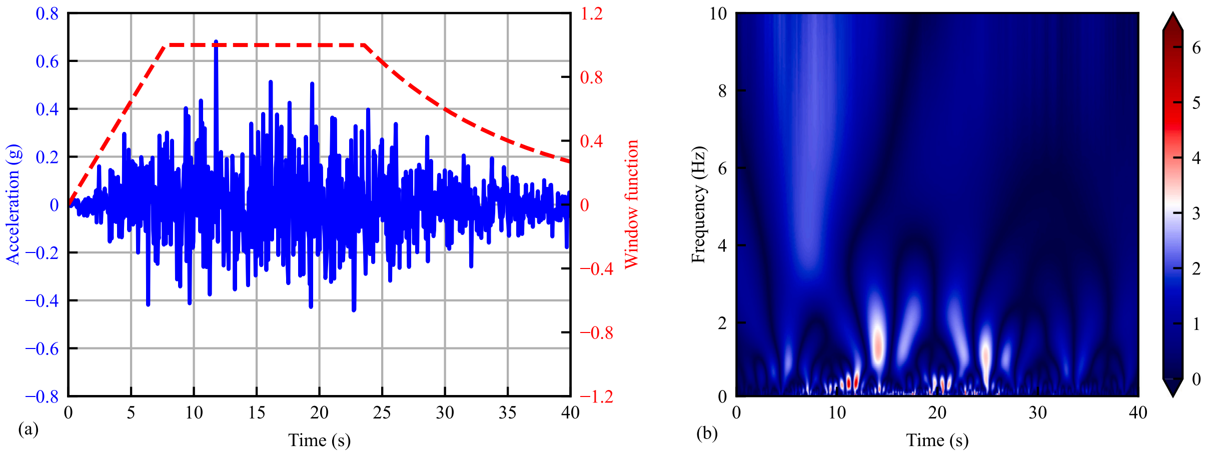

Fig. 8. A sample of synthesized ground acceleration, window function, and continuous wavelet transformation of the ground acceleration.

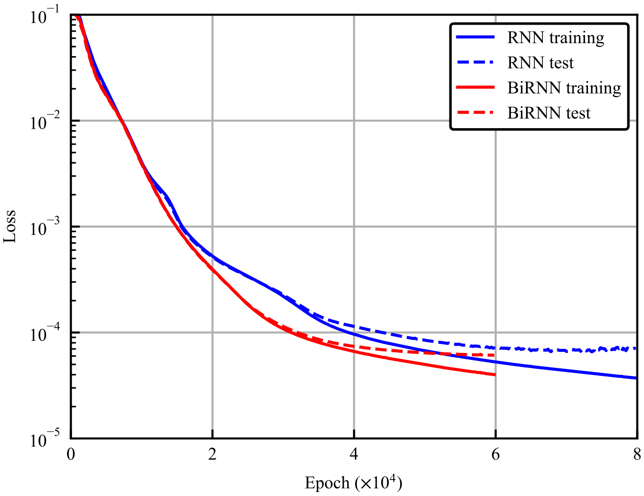

Fig. 9. Loss curves from the pre-training of RNN and BiRNN for state estimation of the shear-type structure.

dimension for both models is set to 30. Within the MLP cell, there are 4 layers, each hidden layer being equipped with 26 neurons, and the Tanh activation function is utilized. The RNN and BiRNN models contain 3858 and 5688 trainable parameters, respectively. For optimal performance, the training process involves 80,000 epochs for the RNN and 60,000 epochs for the BiRNN. Additionally, their training configurations differ in the learning rates: the RNN is set at 1×10^{-5} and the BiRNN at 8×10^{-6}. The training of RNN and BiRNN is based on a personal computing system with an AMD Ryzen 3700X 8-Core Processor, a NVIDIA GeForce GTX 1660 Super graphical card, and 32 GB of RAM. On the software front, the models are implemented using Python 3.10.14 with Pytorch 2.3.0 serving as the deep learning framework. The training durations are recorded to be 364.0 s for the RNN and 334.1 s for the BiRNN. Fig. 9 shows the loss curve from the training of both the RNN and the BiRNN, illustrating that by the end of the training, the BiRNN's performance is better than that of the RNN.

The performance of the trained RNN and BiRNN models can be assessed by visualizing their state estimation outputs on the test dataset and comparing these outputs with the reference. For comparison purposes, two established methods, AKF [6] and DKF [7], are employed to address the same state estimation problem. The hyperparameters in the DKF and AKF algorithms, which include parameters for process noise, measurement noise, and forces modeled as zero-mean Gaussian processes, are tuned using the training dataset to ensure optimal performance. When fed with identical acceleration data from the test dataset, both DKF and AKF are tasked with performing state estimations comparable to those of the RNN models. The outcomes of this comparative analysis are depicted in Fig. 10, highlighting the estimated displacement at the 9th story and velocity at the 13th story. The preliminary visualization of the state estimation data indicates that the four approaches, RNN, BiRNN, DKF, and AKF, deliver acceptable results when applied using the calibrated idealized model.

The normalized root mean square error (NRMSE) is utilized to quantitatively compare the prediction accuracy to the actual reference values. Eq. (12) illustrates the NRMSE formula, where $\tilde{x}_i$ represents the predicted state sequence (displacement or velocity)

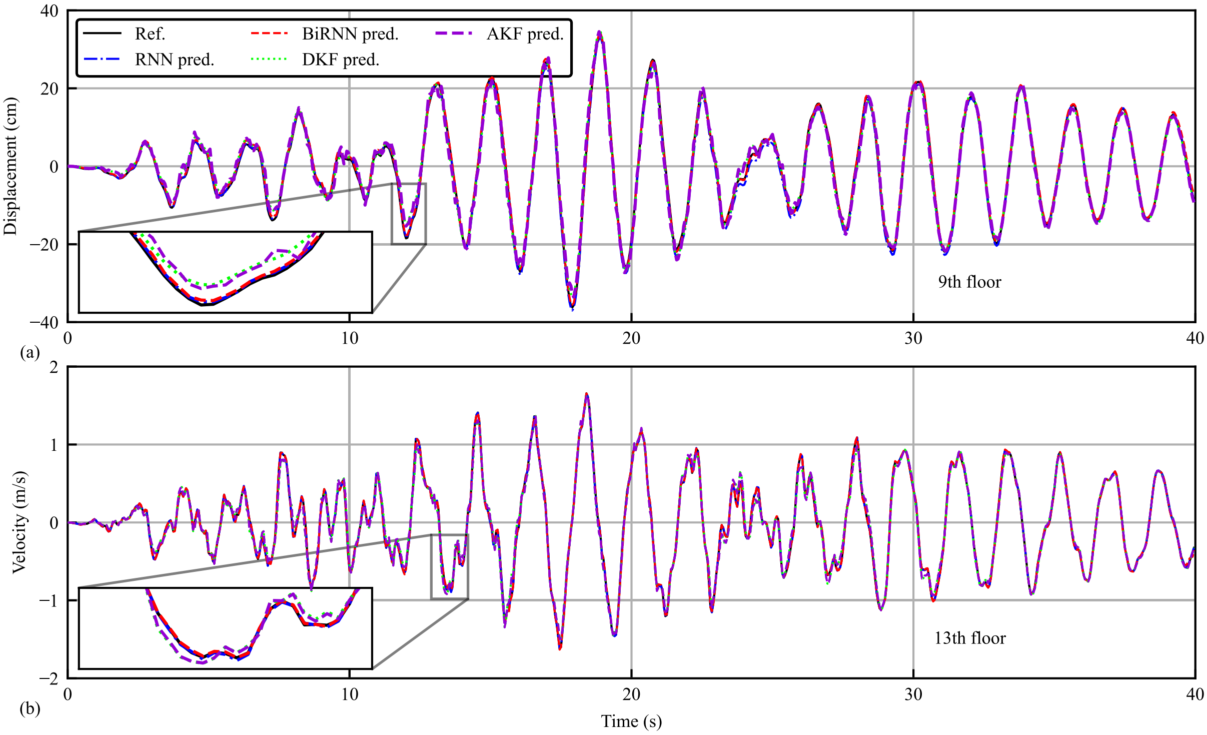

Fig. 10. Comparison of state prediction samples from RNN, BiRNN, DKF, and AKF models; these predictions are based on acceleration inputs from a shear-type structural model and are benchmarked against the reference output of the same model.

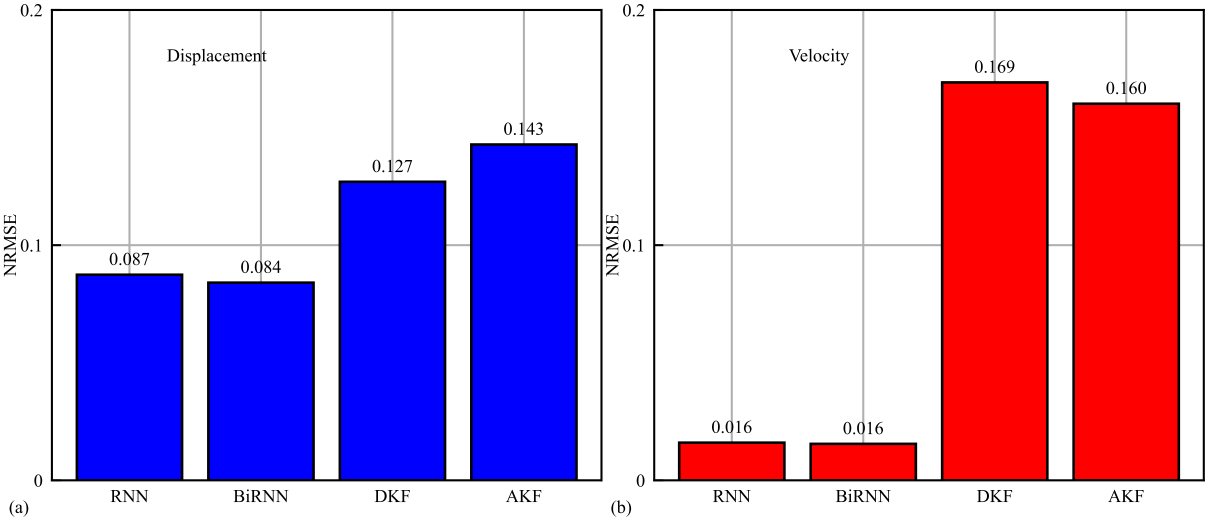

Fig. 11. Averaged NRMSEs of RNN, BiRNN, DKF, and AKF models against the shear-type structural model for displacement and velocity predictions.

at the ith story and $\boldsymbol{x}_i$ is the reference sequence. The notation $\|\cdot\|$ signifies the 2-norm, or Euclidean norm. A lower NRMSE value indicates a higher prediction accuracy.

$$\mathrm{NRMSE}(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) = \frac{\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|}{\|\boldsymbol{x}_i - \mathrm{mean}(\boldsymbol{x}_i)\|}. \tag{12}$$

Within the test dataset, there are 10 sequence batches, with each containing 13 displacement and 13 velocity sequences. An NRMSE score is computed for each sequence. The average NRMSE scores for displacement and velocity predictions are calculated separately. The results, presented in Fig. 11, indicate that the average NRMSEs for both the RNN and BiRNN models are lower than those for two

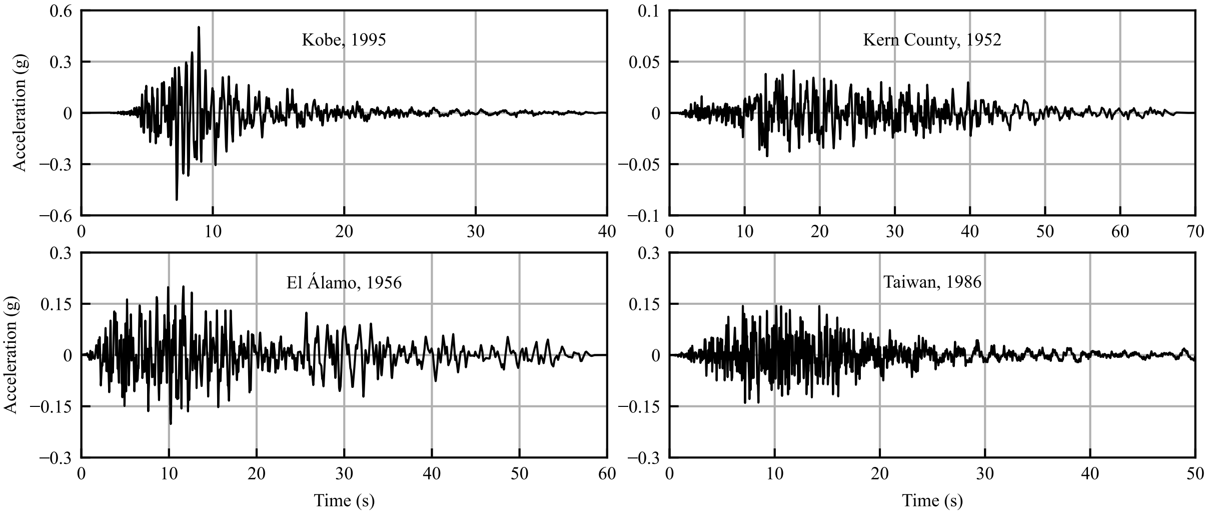

Fig. 12. Ground motion records from Kobe, Kern County, El Álamo, and Taiwan earthquakes.

Kalman filter-based methods. Furthermore, the BiRNN model exhibits a marginally better performance compared to the RNN model. Notably, for both RNN-based models, the predictions for velocity are more accurate than those for displacement, which contrasts with the results from the Kalman filter-based methods. This slight advantage may stem from the fact that the DKF and AKF employed in this study adhere to the standard procedures outlined in [6,7], with only three parameters regulating the Kalman filter estimation process. However, for the 13-DOF structure examined here, subject to non-stationary seismic excitation, the assumption through three parameters appears overly simplistic. It should be clarified that if the Kalman filter's assumptions are made more complex by incorporating specific conditions, the NRMSE values of the estimated states might be lower than those of the RNN models. Since comparing the inherent strengths and weaknesses of state estimation models is not the primary concern of this paper, this paper will not discuss it further. In essence, based on the performance of the four models discussed above, it can be concluded that all of them satisfy the practical application requirements under conditions where FE model errors are negligible.

4.4. TL-RNN for state estimation of base-isolated building under seismic excitation

Initially, the calibrated shear-type structure-derived RNN and BiRNN models are employed for state estimation of the base-isolated building under seismic excitation. However, their performance is not expected to be optimal due to the differences in dynamic behavior between the shear-type structure and the base-isolated building during such strong excitations. To begin, ground motion histories from four distinct regions are collected and normalized using specific scaling factors, as illustrated in Fig. 12. These ground motions are much stronger than the synthesized ambient excitations, and in such cases, they can excite a higher portion of nonlinear dynamic responses of the base-isolated building which might not be reflected in the linear shear-type structure. Moreover, the records display variations from synthesized ground acceleration in peak values, duration, and power spectral densities. Nevertheless, the bulk of the energy is concentrated below 10 Hz, which is similar to that of the synthesized acceleration for training of RNN. Subsequently, the responses of the base-isolated building to these real-world strong ground motions are computed. Accelerations from the 1st to the 5th floors are extracted from the responses to serve as input to the pre-trained RNN and BiRNN. The states from these calculated responses provide a benchmark for assessing the performance of the neural networks. Concurrently, the DKF and AKF given in the previous subsection are applied as well, utilizing the same acceleration data to conduct state estimation for the building.

A sample of the velocity estimation results is depicted in Fig. 13, illustrating that all four methods yield satisfactory performance. For displacement estimation, a sample of results from the four methods is presented in Fig. 15, revealing that each method exhibits varying degrees of deviation in displacement prediction. The BiRNN demonstrates the least deviation, while the RNN shows the most. This discrepancy in prediction is attributed to the inherent differences between the simplified shear-type structure and the base-isolated building structure, even though the simplified structure has been calibrated with the monitoring data. The RNN and BiRNN are trained and tested using data derived from the simplified structure, while the DKF and AKF are directly deduced from the motion equations of the simplified structure. Therefore, the introduction of the TL-RNN framework is essential to enhance the precision of state estimation.

In this study, inter-story displacements [31] between adjacent floors from the 1st to 5th floors/DOF are assumed to be monitored during the earthquake. Therefore, there are extra measurements regarding 4 inter-story displacements beyond the acceleration measurements. The extra measurements are divided into two sets at each time step, one set (the training dataset) contains the information on 1–2, 2–3, and 3–4 inter-story displacements, while the second set (the test dataset) contains the 4–5 inter-story displacement. The RNN and BiRNN are fine-tuned based on the training and test datasets. The parameters in the MLP cells are kept

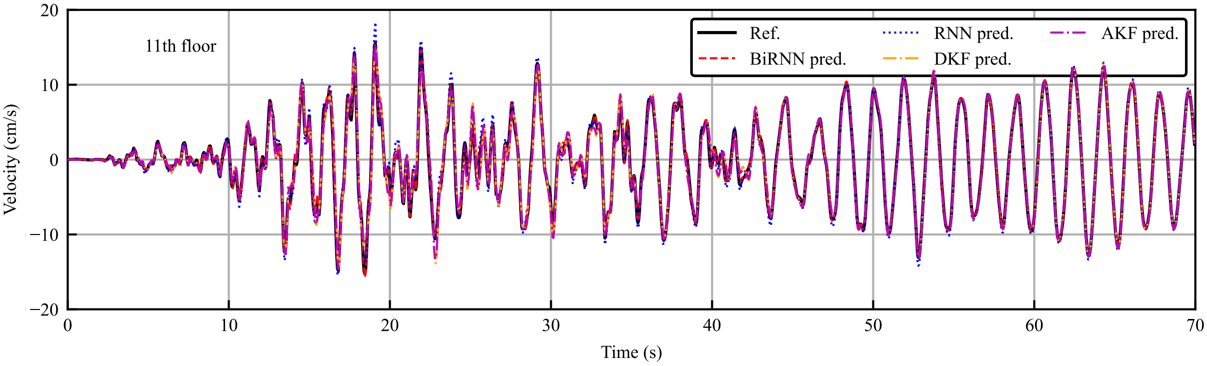

Fig. 13. Velocity estimation results of the 11th floor of the base-isolated structure subject to the Kern County earthquake.

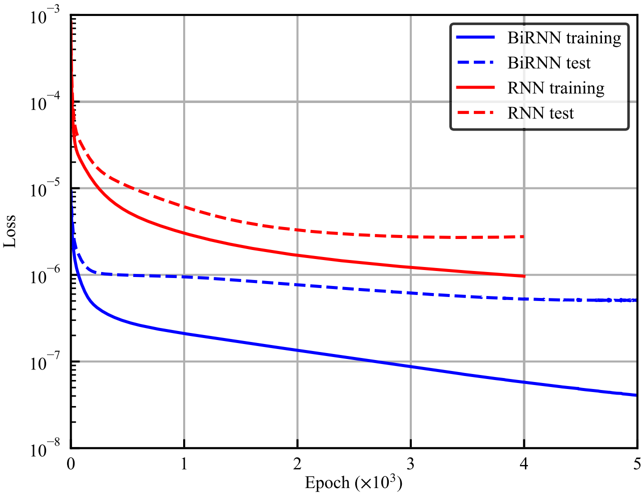

Fig. 14. Loss curves from the transfer learning of RNN and BiRNN under the Kobe earthquake.

constant, and only those on the RNN cells are optimized based on Eq. (5). In total, there are 1050 parameters for RNN and 2100 parameters for BiRNN to be re-trained. The learning rate is set as 1×10^{-5}. The training and test loss curves corresponding to the ground motion in the case of Kobe earthquake are shown in Fig. 14, where the test loss from BiRNN is lower than that of RNN by the end of transfer earning. In comparison with the training of RNN and BiRNN, the fine-tuning process takes much fewer epochs to converge. Under the same computing environment, for the RNN model, the time cost for transfer learning is 43.9 s, 105.0 s, 68.1 s, and 70.1 s for the four ground motion cases, respectively; for the BiRNN model, the time cost is 61.6 s, 160.2 s, 88.67 s, and 56.47 s, respectively.

For comparison purposes, the same dataset, including the inter-story displacements and accelerations, is offered to the DKF and AKF for state estimation, resulting in the designation of integrated DKF and integrated AKF to distinguish them from their traditional counterparts. The procedures for implementing integrated DKF and integrated AKF are similar to those of the traditional DKF and AKF. It is important to note that TR-RNN and integrated prediction methods fundamentally differ: the TL-RNN modifies the pre-trained state estimation RNN model by using the extra measurements, whereas the integrated prediction methods provide the state estimation model with additional reference data for forecasting.

The displacement prediction samples for the base-isolated building using the extra measurements are shown in Fig. 15. Due to the inherent limitation of the DKF algorithm, the inclusion of extra information does not alter the DKF predictions, whereas the other three methods experience varying degrees of precision enhancement. The TL-BiRNN outperforms the other methods, followed by the TL-RNN, and then the Integrated AKF. Even though the extra measurements have enhanced the predictive capability of the AKF, this enhancement is often evident in corrections to the overall trend. During strong seismic events, its state predictions still exhibit significant discrepancies when compared with the reference data. In contrast, the TL-RNN and TL-BiRNN maintain a certain level of predictive accuracy throughout the entire seismic event. This may be due to the pronounced nonlinearity exhibited by the base-isolated building under earthquake conditions, particularly in the interval between 10 s and 30 s, as observed in the Kern County earthquake. Such nonlinearity cannot be represented by the calibrated shear-type structure. Therefore, the refinement of the state prediction model itself (i.e., the transfer learning) based on additional measurement data is necessary to achieve better predictive

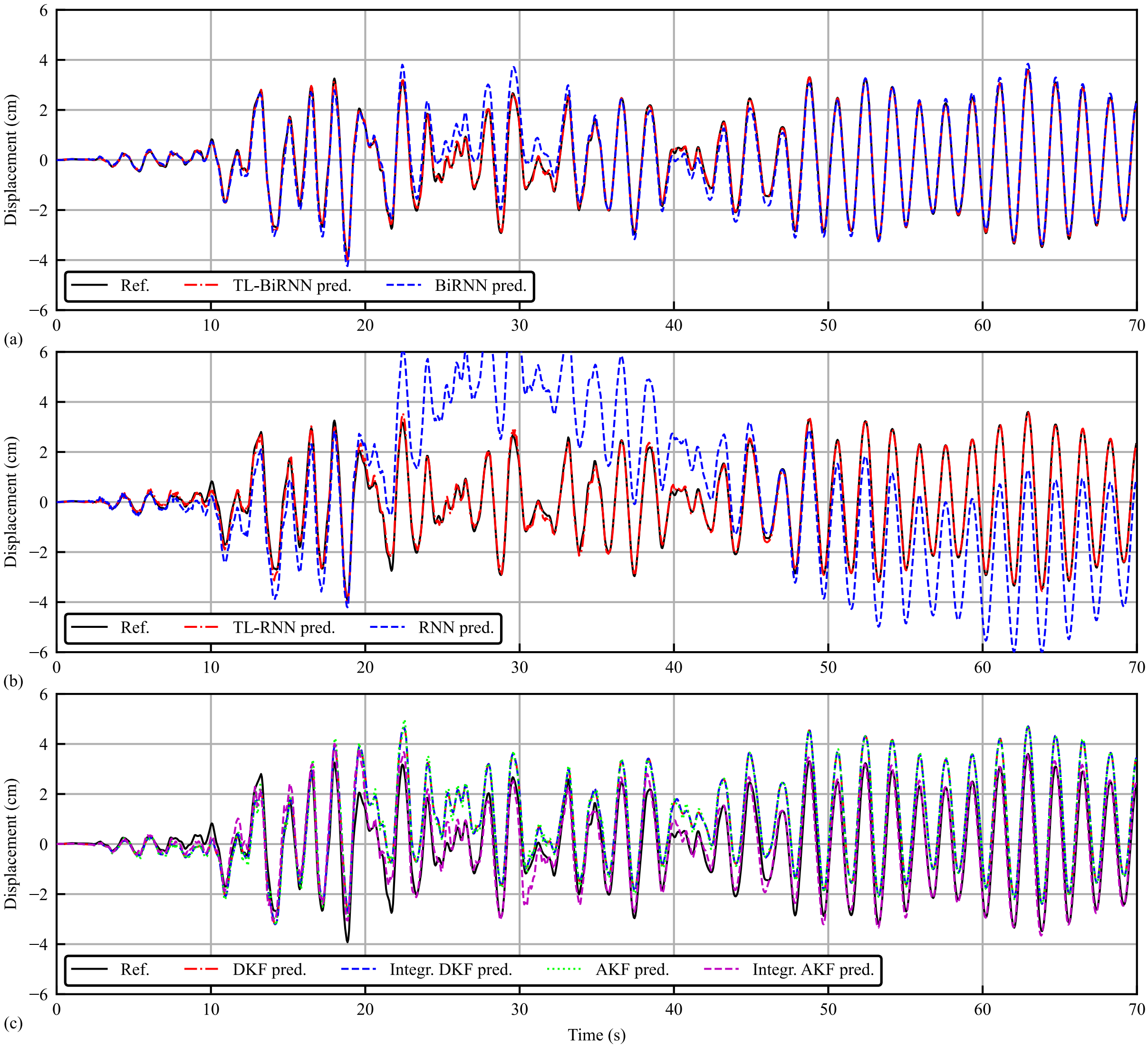

Fig. 15. Displacement estimation results of the 11th floor of the base-isolated structure subject to the Kern County earthquake.

accuracy. In summary, when the structural model does not align with the actual conditions, the accuracy of state estimation can be significantly improved by fine-tuning the RNN model using real-world measurements. This method of refinement yields better results than simply incorporating extra measurements into the existing model as additional inputs for prediction.

Quantitative evaluations of model performance with the inclusion of extra measurements are carried out using the NRMSE metric. In this context, the assessment of the integrated DKF predictive results is omitted, as additional measurements do not impact its predictions. For each of the four strong ground motion cases, NRMSEs for all predicted displacement sequences against their corresponding references are calculated. These NRMSE values are then averaged for each case, and the results are presented in Fig. 16. In cases where the simplified model diverges from the actual structure, the proposed TL-RNN framework has proven to be effective in ensuring the accuracy of state estimation. Following the application of transfer learning, the performance of the BiRNN is outstanding across all four cases, while the TL-RNN also demonstrates a commendable level of accuracy.

On the other hand, when using the same full dataset, integrated AKF does improve state estimation accuracy; however, its performance is not as robust as that of the TL-RNN and TL-BiRNN models. This limitation arises because integrated AKF utilizes the entire dataset (including both inter-story displacements and accelerations) as input for state estimation, leaving the primary issue of the deviation between shear-type structures and base-isolated buildings unresolved. Some may argue that the AKF could be tuned by adopting an information utilization procedure similar to that of RNN transfer learning, such as adjusting its process noise based on the full dataset. In the standard setting of AKF, the process noise is assumed to be Gaussian, governed by a single variance parameter. It is important to note that, in this case study, the process noise is not merely a simple probabilistic distribution but

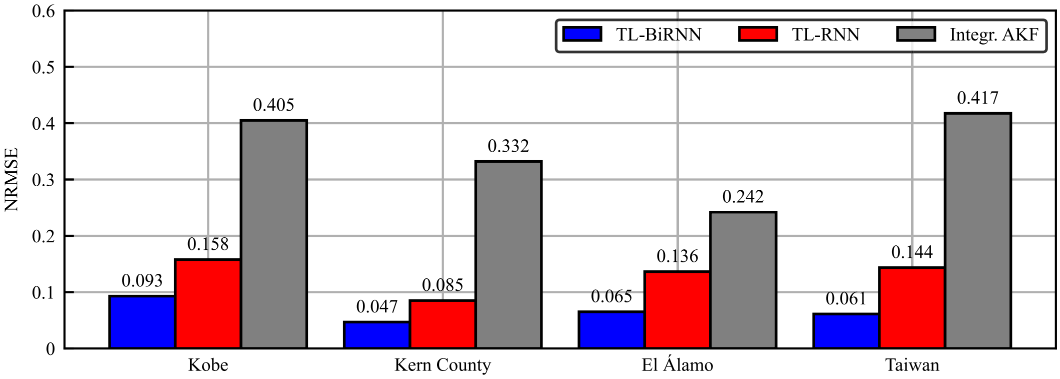

Fig. 16. Averaged NRMSEs of TL-BiRNN, TL-RNN, and Integrated AKF for displacement prediction under the four earthquakes.

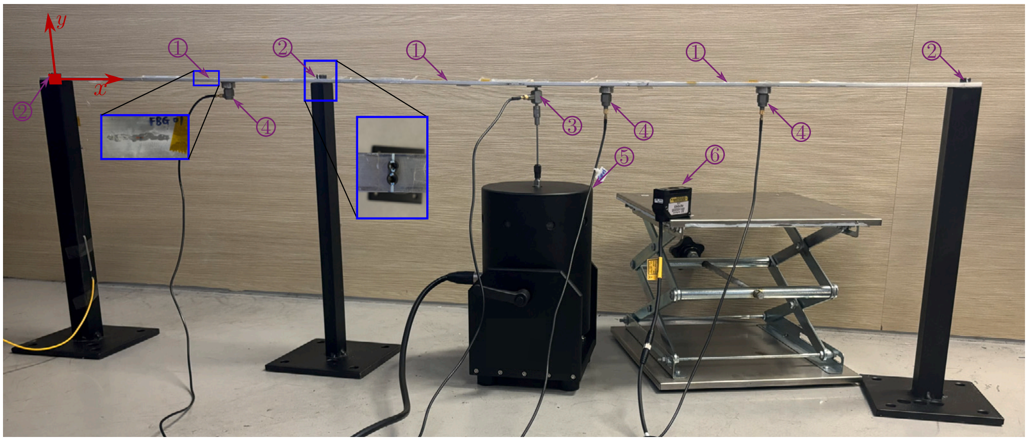

Fig. 17. Overview of the experimental setup where key components are highlighted and numbered: ① fiber Bragg gratings, ② bearings with bolted connections, ③ load cell, ④ accelerometers, ⑤ shaker, and ⑥ laser distance sensor.

is correlated with the structural response and exhibits heteroscedasticity. Thus, simply adjusting the parameter for process noise would not be as effective as the TL-RNN, which provides a sufficiently generic framework to make the RNNs adaptable to actual structures. Additionally, to the authors' best knowledge, since there are many parameters and coefficients in a constructed Kalman filter model, there is no general rule for effectively tuning each specific Kalman filter model. Consequently, this paper does not pursue further investigation into this issue.

5. Experimental example

5.1. Measurement setup

In the experiment, the investigation focuses on a 1.26 m two-span aluminum beam that is continuously supported. Fig. 17 presents an overview of the experimental setup. The primary components of this setup are numbered from ① to ⑥, including fiber Bragg gratings (FBGs), bearings with bolted connections, a load cell, accelerometers, a shaker, and a laser distance sensor. From left to right, the three FBGs and three accelerometers are abbreviated as FBG1-FBG3 and A1-A3. Specifically, FBG1-FBG3 have wavelengths of 1563 nm, 1555 nm, and 1527 nm respectively, with a grating length of 5 mm, a bandwidth of less than 0.4 nm, a side-mode suppression ratio greater than 12 dB, and a reflectivity of over 75%. The accelerometers (A1-A3), load cell, and laser distance sensor are all piezoelectric types that convert collected voltage signals into data. The shaker, connected to a power amplifier and a waveform generator, is controlled by digitally-generated voltage signals from a computer to deliver continuous stochastic excitations at a specific location on the beam. Concurrently, the sensor array is designed to continuously record original data at their respective locations with a 5000 Hz sampling frequency. As a result, at each time step, the system captures measurements from the three FBGs and three accelerometers, along with load measurement from the load cell and deflection measurement from the laser distance sensor. This array of sensors ensures comprehensive monitoring and accurate state estimation of the beam.

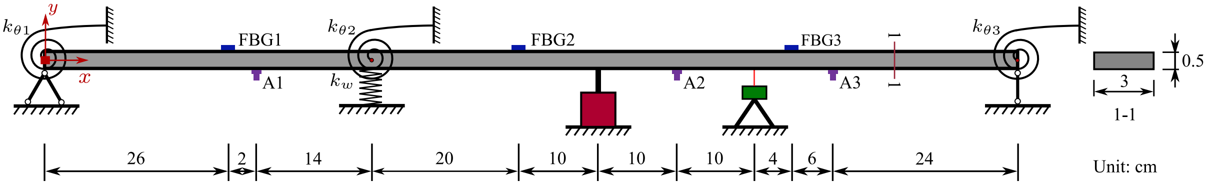

Fig. 18. Idealized model for the beam with instruments positioning.

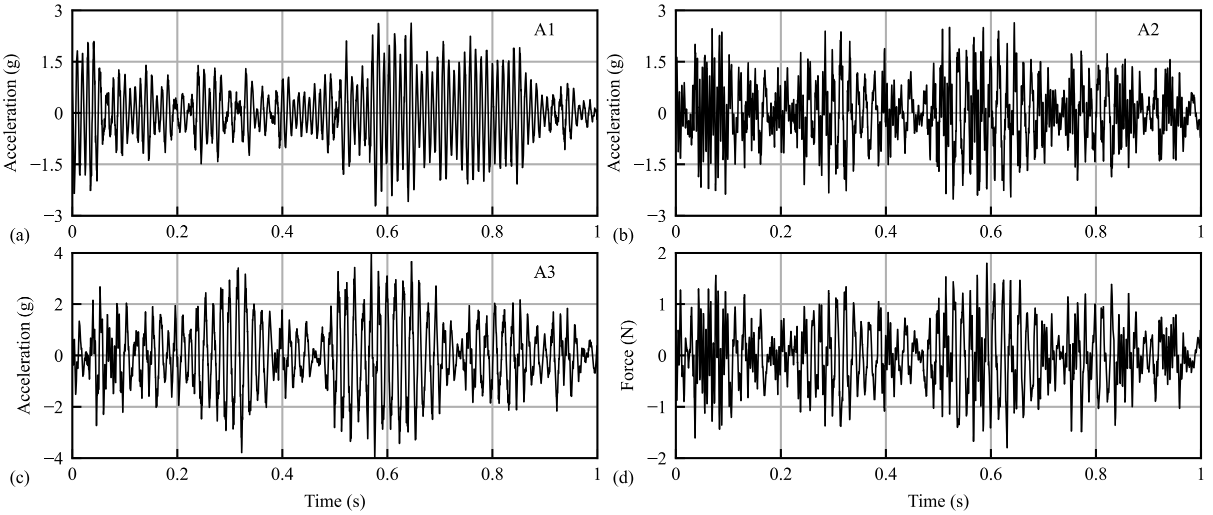

Fig. 19. Measured acceleration and force data in experimental modal testing.

5.2. Model calibration

In this experiment, the bolts at both ends of the beam are securely tightened, while those in the middle are deliberately left somewhat looser. This arrangement introduces controlled imperfections in the boundary conditions of the beam. Fig. 18 illustrates the idealized model of the continuous beam, detailing the structure and instrument configurations. As shown in Fig. 18, the two ends of the beam are transversely fixed, while the rotations of the three supports are constrained by rotational springs with stiffness values $k_{\theta1}$, $k_{\theta2}$, and $k_{\theta3}$. Additionally, a linear transversal spring is added to the middle support, denoted by its stiffness k_w. In this setup, the Euler–Bernoulli beam element is utilized for the FE model, incorporating two DOFs at each node, i.e., the transversal deflection and rotation. Each element is 2 cm long, resulting in a total of 63 elements, 64 nodes, and 126 DOFs for the FE model. To fully define the FE model, except for the geometric parameters, the elastic modulus E, density ρ, rotational spring parameters $k_{\theta1}$-$k_{\theta3}$, and transversal spring parameter k_w must be determined based on the monitoring data from the beam. Moreover, the damping model employed is Rayleigh damping, with the assumption that the first and second modes share the same damping ratio. Hence, the single parameter damping ratio ζ for these two modes needs to be calibrated as well.

An experimental modal test is conducted on a beam. A digital voltage waveform, set to a 2 V peak-to-peak amplitude, is generated by a computer and input into a waveform generator. The resultant output is a band-limited white noise signal, confined to a frequency range of 10 Hz to 100 Hz. The duration of this signal is fixed at 25 s. The power amplifier settings are adjusted so that the standard deviation of the output force is approximated to around 1 N. Concurrently, accelerations at points A1 to A3 on the beam are recorded. Fig. 19 displays the measured acceleration and force data over a 1 s interval.

With all the collected data over the 25 s, averaged Fourier transforms of excitation and acceleration signals are computed using a sliding window of 5 s per block. Subsequently, the frequency response functions (FRFs), denoted as $\hat{H}_i(f)$ where $i = 1, 2, 3$ and $f = 0.2k$ (with $k \in \mathbb{N}_0$ and $k < 1.25 \times 10^4$), are evaluated via the ratio of the Fourier transform results of acceleration to excitation. Only the FRF data ranging from 10 Hz to 100 Hz is utilized for updating the parameters of the FE model [39]. The optimization problem is formulated as follows:

$$\min \sum_{k=50}^{500} \sum_{i=1}^{3} \left[\left(\log_{10} \frac{|\hat{H}_i(0.2k)|}{|H_i(0.2k; \boldsymbol{\eta})|} \right)^2 + \left(\angle \hat{H}_i(0.2k) - \angle H_i(0.2k; \boldsymbol{\eta}) \right)^2 \right]. \tag{13}$$

Here, $\boldsymbol{\eta} = \{E, \rho, k_{\theta1}, k_{\theta2}, k_{\theta3}, k_w, \zeta\}$ represents the set of parameters to be updated; $|\cdot|$ and $\angle$ denote the magnitude and phase operators, respectively; $H_i(0.2k; \boldsymbol{\eta})$ is the ith FRF predicted from the FE model given $\boldsymbol{\eta}$. The L-BFGS-B algorithm is employed for

Table 2
Nominal values and scaling factors for parameters in the beam model.

Parameter	Symbol	Nominal value	Scaling factor	Unit
Elastic modulus	E	6.85×10^{10}	0.957	N/m^2
Density	ρ	2.70×10^{3}	1.278	kg/m^3
Rotational stiffness of left support	$k_{\theta 1}$	1.00×10^{4}	1.487	N/rad
Rotational stiffness of middle support	$k_{\theta 2}$	1.00×10^{1}	0.313	N/rad
Rotational stiffness of right support	$k_{\theta 3}$	1.00×10^{4}	0.902	N/rad
Transversal stiffness of middle support	k_w	1.00×10^{5}	2.951	N/m
Damping ratio of 1st and 2nd mode	ζ	8.00×10^{-3}	1.415	–

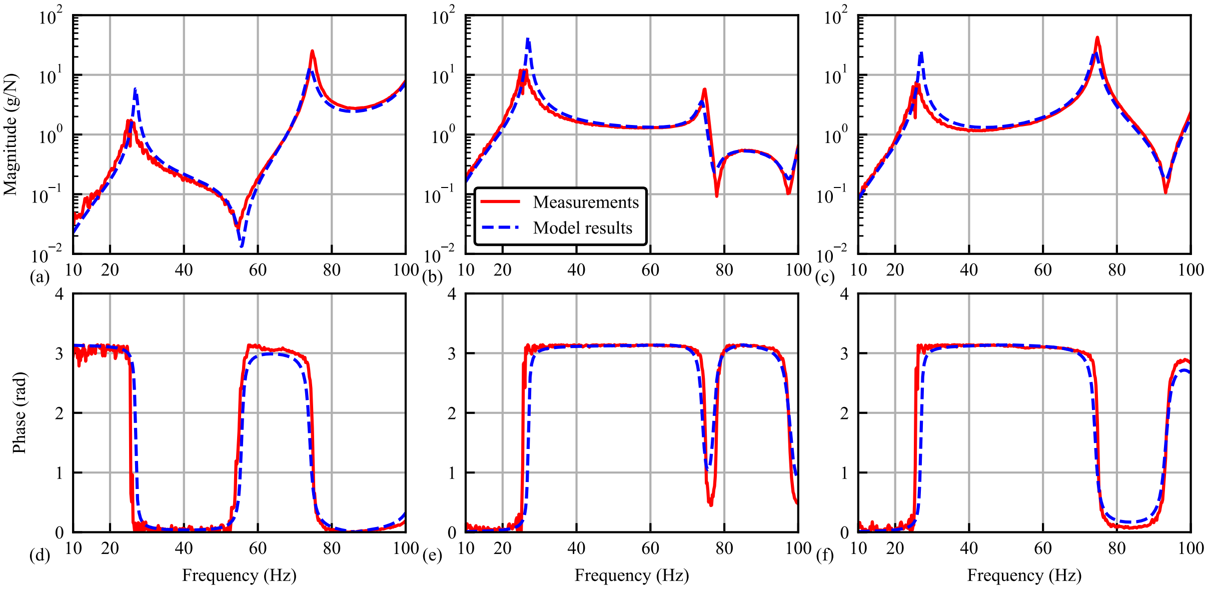

Fig. 20. Measured and calibrated model output single-sided FRFs.

optimization, and it converges with the function values in Eq. (13) being 0.148. The nominal values and optimal scaling factors for each element in $\boldsymbol{\eta}$ are presented in Table 2. Visualizations comparing the single-sided FRFs from the updated model and measurements are shown in Fig. 20. The results indicate that the principal trends between the model and measurements are almost identical, although some discrepancies exist, e.g., the first modal frequency being higher than measured, and the second being lower. Nevertheless, the calibrated model closely approximates the real structure, and it will be used for further state estimation tasks in the following subsections.

5.3. Pre-training of RNN based on the calibrated beam model

The training of the RNN and BiRNN follows the procedure outlined in Fig. 21. Initially, a stochastic force generator is programmed to produce white noise force histories within the frequency range of 10 Hz to 410 Hz, based on a flat power spectral density. In this process, only two force histories are used: one for the training dataset and one for the test dataset, both with a standard deviation of 0.5 N. These force histories are then fed into the calibrated FE model, which computes the system response, including acceleration, velocity, and displacement at each DOF. The duration for each batch of response is 4 s, with a sampling frequency of 5000 Hz. That is, there are 20,000 × 3 acceleration data fed into the RNN model, and the model will output $20,000 \times 252$ data of structural state. To enhance the convergence of RNN training, all response data are normalized. The normalized displacement and velocity data are treated as labeled states at each node. Noise is subsequently added to the acceleration data at locations A1 to A3, which is similar to priors embedded in noise as in Kalman filter-based methods. In this case study, the noise-to-signal ratio is presumed to be 10%. The noisy, normalized acceleration data are used as input to the RNN models, enabling the RNN to predict states at each node of the FE model. Accordingly, the mean square error loss function is formulated based on the predicted state and the labeled state, and the Adam optimizer is utilized to efficiently tune parameters in both RNN and BiRNN models.

The hidden state dimension is uniformly set at 36 for both the RNN and BiRNN models. Each model includes an MLP component with three layers. The RNN configuration features 36 neurons in each of the first two layers and 126 neurons in the third layer. In contrast, the BiRNN is equipped with 72 neurons in each of the first two layers and 126 neurons in the third layer. Biases in the propagation paths of both the RNN and BiRNN have been removed. In total, there are 8532 parameters in RNN and 22,248 parameters in BiRNN to be trained. Both RNN and BiRNN models are trained over 50,000 epochs using a consistent learning rate of

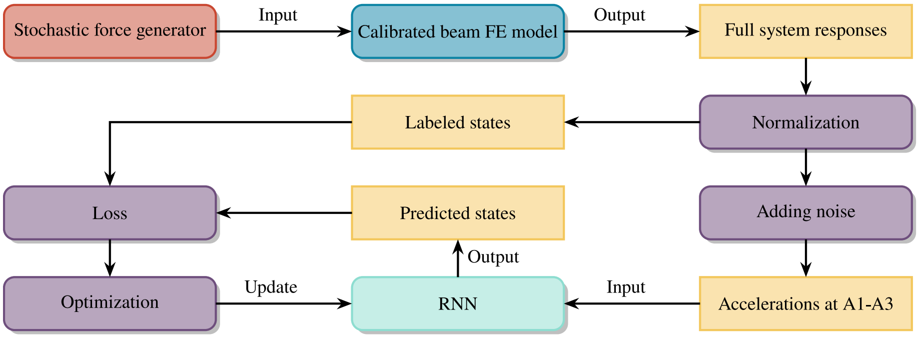

Fig. 21. Training procedures of RNN and BiRNN for the beam.

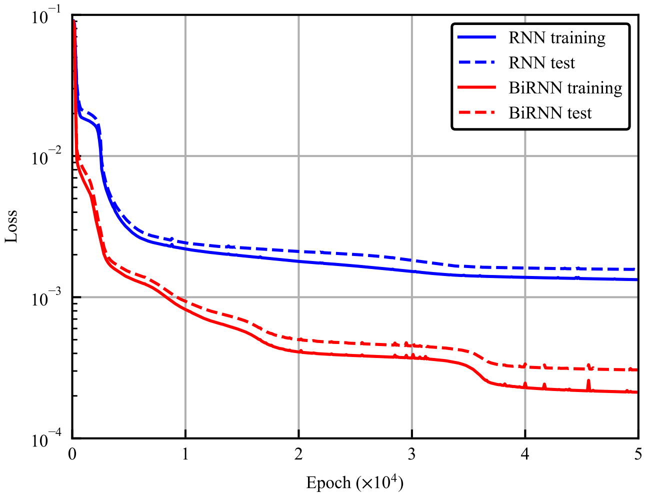

Fig. 22. Loss curve from the training of RNN and BiRNN for the beam.

3×10^{-5}. The training and test loss curves, presented in Fig. 22, demonstrate that the RNN model's test loss converges at 1.33×10^{-3}, while the BiRNN model's test loss converges at 2.12×10^{-4}. These results indicate that, with the simple test dataset generated from the calibrated model, the BiRNN significantly outperforms the RNN in terms of state estimation accuracy.

To visualize the models' performance, the predicted displacement and velocity fields over 0.4 s are displayed for the test dataset in Fig. 23. The horizontal axis represents time, while the vertical axis corresponds to the spatial coordinates of the beam, illustrating the full range of states. Both models yield satisfactory state estimations. Fig. 24 further illustrates the predicted deflection at 0.76 m and rotational speed at 0.52 m during the initial 2 s of the test period. While the RNN model shows slight deviations in amplitude from the reference data, the BiRNN model displays superior accuracy and closer adherence to the expected outcomes.

The acceleration data measured from sensors A1 to A3, as displayed in Fig. 25, are fed into the well-trained RNN and BiRNN models. It is important to note that the acceleration data presented are not as ideal as those in the training and test datasets; the noise, in this case, can be categorized into two distinct types: noise originating from the voltage signal and noise resulting from the discrepancy between the model and the actual structure. Furthermore, Fig. 25 demonstrates that there could be some outliers distorting the signal [40]. These imperfections present a significant challenge to accurate prediction of the beam's system state by the RNN and BiRNN models. To quantitatively assess the performance, the deflection measured by a laser distance sensor at a distance of 92 cm is used for comparison with the results from the RNN and BiRNN models. A preliminary laboratory test revealed that this laser distance sensor could only measure vibrations up to approximately 35–40 Hz. Consequently, a second-order low-pass Butterworth filter with a cutoff frequency of 38 Hz has been implemented to filter the outputs from the RNN and BiRNN. Performance is evaluated using NRMSE, detailed in Eq. (12). The NRMSE for the RNN is 0.545, while for the BiRNN, it is 0.221, showing that the BiRNN model achieves superior prediction accuracy over the RNN model. Such differences have been reflected during the training of RNN and BiRNN models using the training set. When the input acceleration data is subject to noise interference, the bidirectional propagation of information and computation of the state can effectively reduce such interference. In contrast, the unidirectional approach appears to be less effective. However, the unidirectional propagation allows the RNN model to perform online state estimation, which is not achievable with the BiRNN.

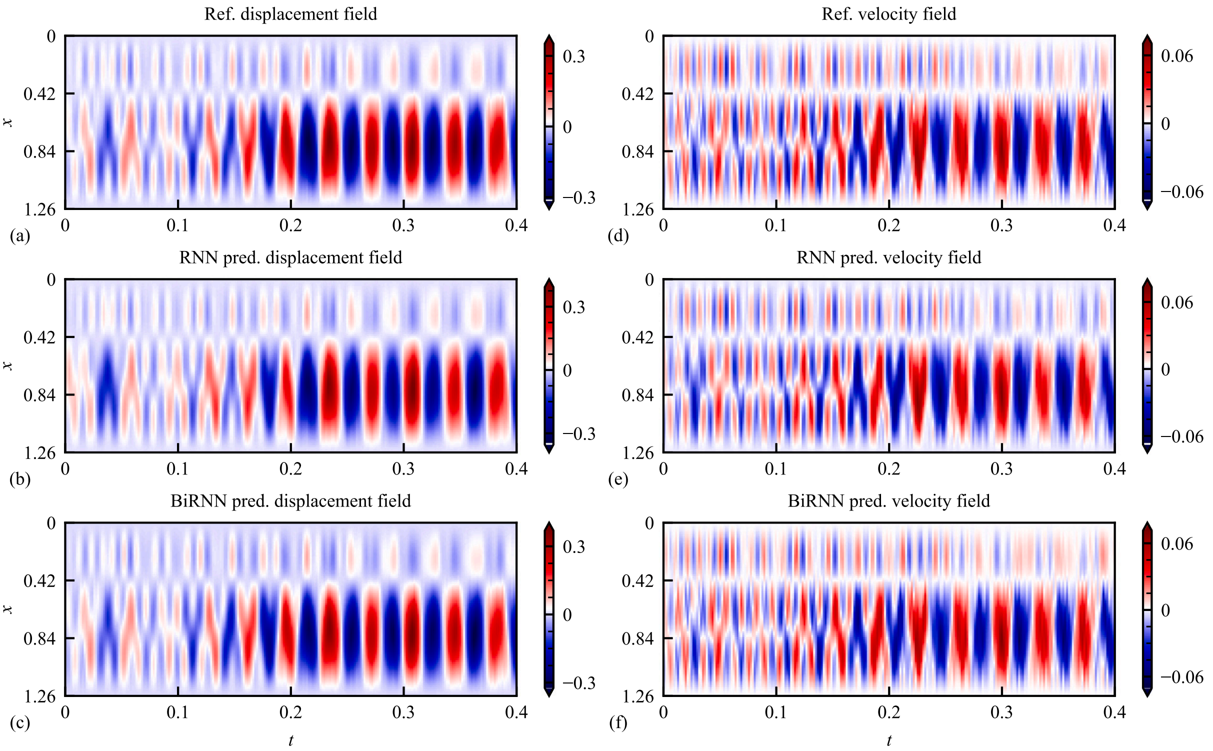

Fig. 23. Predicted displacement and velocity field over the beam in 0.4 s.

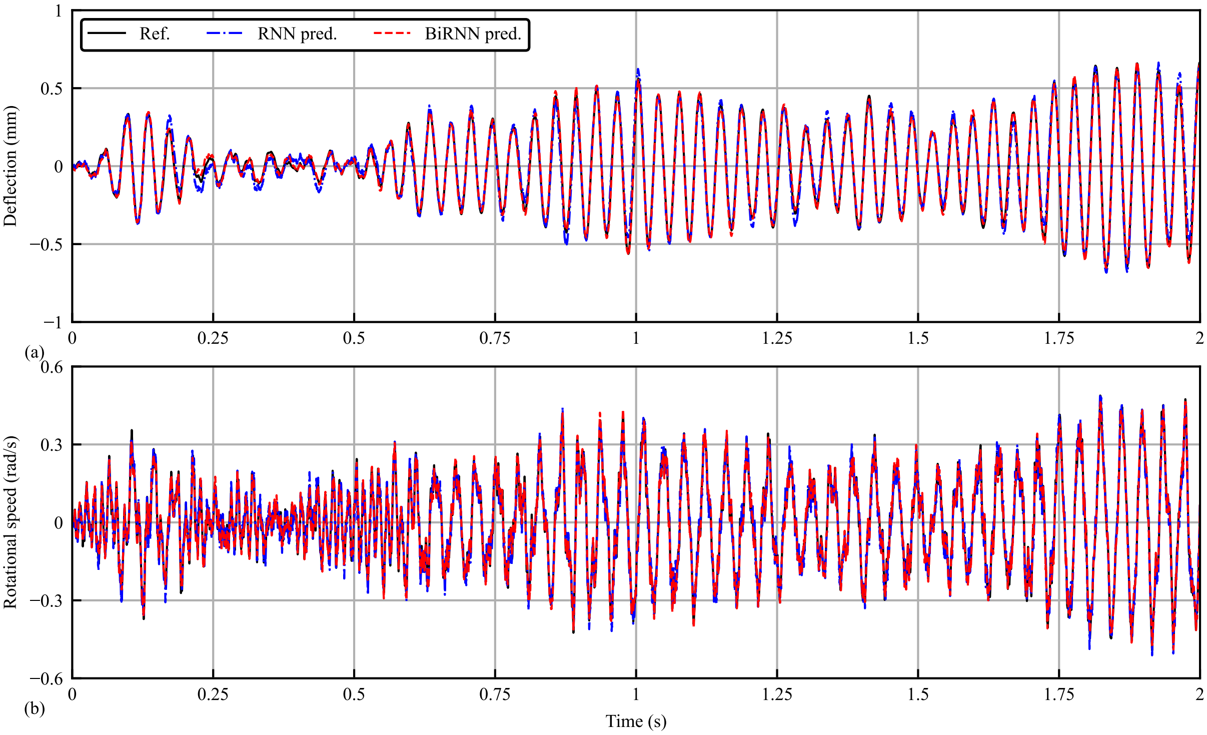

Fig. 24. Prediction results of deflection at 0.76 m and rotational speed 0.52 m of the beam.

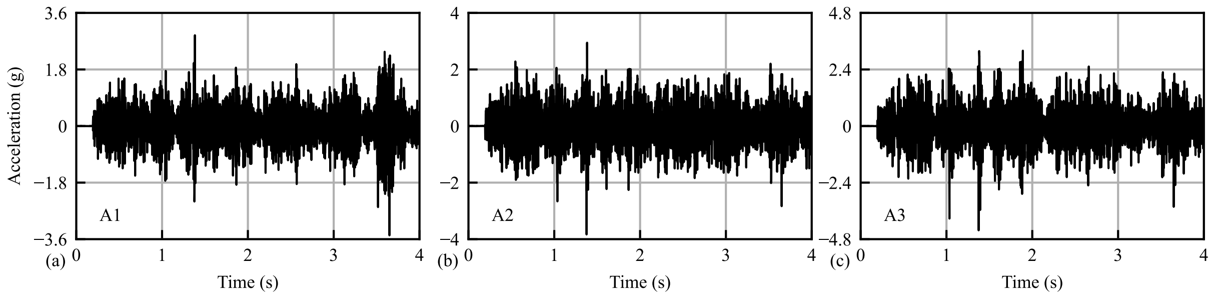

Fig. 25. Measured acceleration data from A1 to A3.

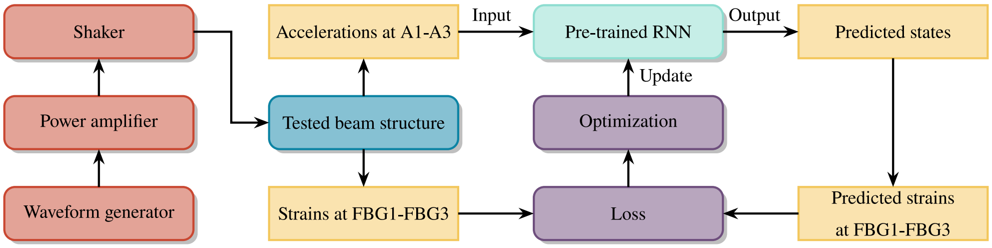

Fig. 26. Detailed procedures for RNN transfer learning to enhance the performance of beam state estimation.

5.4. Using transfer learning to enhance the model performance

The sensor array, as illustrated in Fig. 17, captures comprehensive data during the vibration process. This includes not only acceleration and deflection measurements but also strain data from three specific locations on the beam. The strain information, in conjunction with the measured acceleration data, is valuable for the transfer learning process discussed in this subsection, enabling the pre-trained RNN to adapt to the actual tested beam structure.

Let $\hat{\epsilon}_k \in \mathbb{R}^2$ denote the measured strain from FBG1-FBG2, and $\tilde{\boldsymbol{x}}_k \in \mathbb{R}^{252}$ represent predicted state by the pre-trained RNN, where $k = 1, 2, \ldots, N$, and N is the total number of timestamps. The predicted state vector $\tilde{\boldsymbol{x}}_k$ can be linearly transformed into the predicted strain vector $\tilde{\epsilon}_k$ using the linear operator $\mathbf{B}_\epsilon \in \mathbb{R}^{2\times 252}$. The operator, which is derived from the second derivatives of the four shape functions associated with the Euler–Bernoulli beam element, relates solely to the geometry of the beam and the location of FBGs. The loss function for transfer learning can be formulated as:

$$\min \sum_{k=1}^{N} \mathcal{L}\left(\mathbf{B}_\epsilon \tilde{\boldsymbol{x}}_k, \hat{\epsilon}_k\right). \tag{14}$$

During the training process, the parameters of the pre-trained RNN are fine-tuned. The training dataset incorporates measured strain data from FBG1 and FBG2, while strain data from FBG3 is reserved for the test dataset to prevent overfitting in the transfer learning process. Fig. 26 illustrates the procedures for implementing RNN transfer learning to enhance the performance of RNN prediction.

As detailed in Section 3.2, during the transfer learning process, all parameters except those within the RNN cells are kept constant. Consequently, there are 1404 parameters in the RNN and 2808 in the BiRNN that require retraining using the measured strain data. This represents that 16.46% of the parameters in the RNN and 12.62% in the BiRNN will undergo transfer learning. The learning rate is set at a very low 1×10^{-6} because the success of transfer learning in this scenario is highly sensitive to the learning rate. In this problem, it is found that stability in the optimization process can only be achieved when the learning rate is significantly lower than that used in the initial training phase of the RNN and BiRNN. The transfer learning for RNN and BiRNN converge after 1420 epochs and 2420 epochs of training, respectively. NRMSEs from the TL-RNN and TL-BiRNN predicted deflection at 92 cm of the beam are calculated as well, and comparisons are made with those from the RNN and BiRNN in the previous subsection, as presented in Fig. 27. The results indicate that through the use of two FBGs' information, the transfer learning-based RNN and BiRNN models significantly outperform the vanilla RNN and BiRNN. For the RNN, the NRMSE decreased from 0.545 to 0.273, a reduction of 49.9%. In contrast, the BiRNN started with higher initial accuracy, with an NRMSE of 0.221. After the transfer learning, the NRMSE is reduced to 0.168, providing a decrease of 24.0%. Although the accuracy of the TL-RNN does not surpass that of the FE model-based training of BiRNN, it likely constitutes the achievable upper boundary for models designed to conduct online state estimation using only three accelerometer inputs. Finally, having demonstrated that the TL-RNN method proposed in this paper indeed enhances the performance of RNN models, the deflection results at 92 cm for both TL-RNN and TL-BiRNN are presented in Fig. 28. Comparison with the measured values shows superior accuracy of TL-BiRNN in state estimation problems. At the same time, the results confirm that the performance of TL-RNN is acceptable and can be applied to online monitoring and digital twin systems, despite minor discrepancies between predicted and actual values at some time steps.

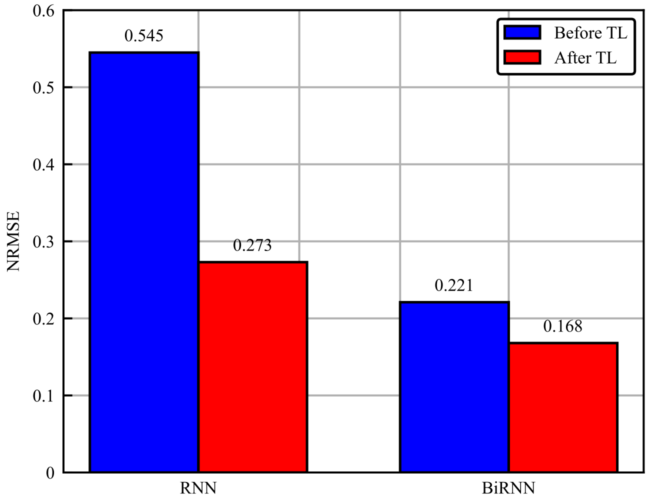

Fig. 27. NRMSE values of RNN, BiRNN, TL-RNN, and TL-BiRNN models for the beam deflection prediction.

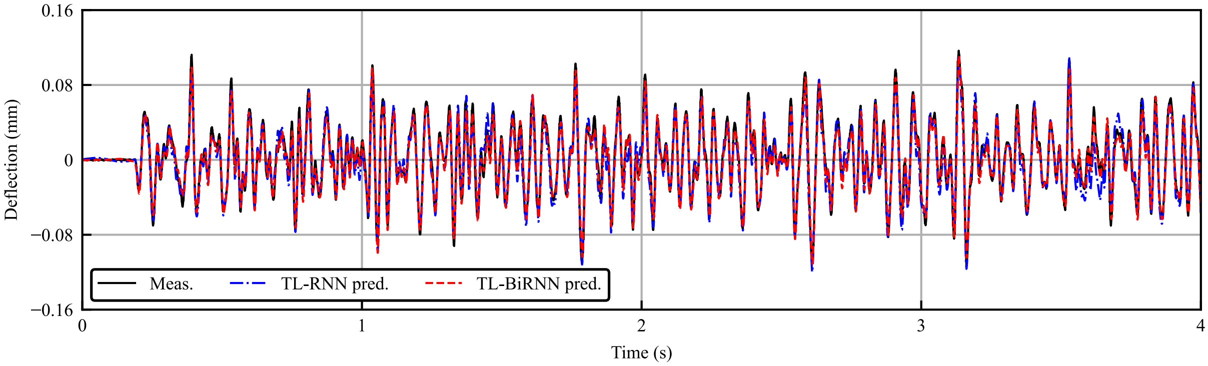

Fig. 28. Comparison of measured and predicted beam deflections using TL-RNN and TL-BiRNN models.

6. Discussion

The research presented in this paper demonstrates the efficacy of RNN transfer learning for state estimation in structural dynamics, providing a commendable alternative to conventional methods. The improved state estimation performance of TL-RNN can be intuitively understood by analyzing how it leverages information compared to traditional methods. Fig. 29 illustrates three approaches for state estimation by using different types of information and their combinations, where Type A represents commonly used acceleration data, Type B encompasses additional measurement data from the structure, and Type C pertains to the estimated state information. Models $\mathcal{M}_1$ to $\mathcal{M}_3$, despite their differences, output identical results, Type C. Traditionally, the approach depicted in Fig. 29(a) is most prevalent, with extensive research focusing on effectively constructing the $\mathcal{M}_1$ model to ensure accurate state prediction. Some methods originate directly from the equations of motion formulated by the FE model, while others derive from the data generated by the FE model.

The interest of this study lies in optimizing the use of increased monitoring data available through sensor arrays in monitoring systems, a scenario that aligns with real-world applications where diverse sensors often collaborate in monitoring tasks. In the numerical example illustrated above, both approaches depicted in Fig. 29(b) and (c) are investigated. The implementations of the two approaches require the same dataset that includes both Types A and B data. The present work indicates that employing all available information as per the approach in Fig. 29(c) substantially outperforms that in Fig. 29(b). This is primarily because models $\mathcal{M}_2$ and $\mathcal{M}_3$ initially require calibration using an FE model, which, even when calibrated, still differs from the actual structure. Such discrepancies can be exacerbated under specific conditions, leading to inaccuracies in the initial outputs of $\mathcal{M}_2$ or $\mathcal{M}_3$. This is evidenced in the numerical example where a multi-story linear model is calibrated using ambient vibration data from the base-isolated building. During an earthquake, the sliding mechanism at the base of the building causes its response to become predominantly nonlinear, rendering the initial prediction of $\mathcal{M}_2$ and $\mathcal{M}_3$ inaccurate. Despite additional data from Type B, the results displayed in Fig. 15 still show low prediction accuracy. Contrarily, using the mode depicted in Fig. 29(c), where the input remains simply Type A but is fine-tuned using both Types A and B data, results in accurate prediction of Type C. This

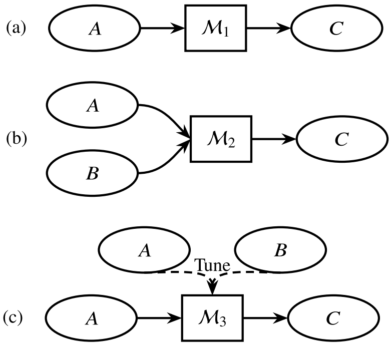

Fig. 29. Three approaches on using information for state estimation in structural dynamics.

suggests that incorporating additional information into model adjustments, rather than directly into model predictions, might yield more precise results. In the experimental validation, the approach shown in Fig. 29(c) is replicated. The FE model of the continuous beam is calibrated using FRF data obtained from experimental modal analysis. The training and test datasets are generated based on the random vibration of the calibrated FE model; and both RNN and BiRNN models for state estimation are trained with those generated structural random vibration data, establishing the information flow from Types A to C as shown in both Fig. 29(a) and (c). Subsequently, the strain at three locations is calculated based on the measured wavelength changes in FBG1 to FBG3, constituting Type B data, and RNN and BiRNN models are fine-tuned accordingly. This results in model $\mathcal{M}_3$, which, despite having the same input information as $\mathcal{M}_1$, demonstrates superior performance, thereby confirming the practical efficacy of $\mathcal{M}_3$ over $\mathcal{M}_1$ in future state prediction tasks.

While the advantages of the TL-RNN model are evident, it remains essential to evaluate its appropriateness for specific problems in comparison with traditional methods. Without introducing additional measurements, a comparison between traditional Kalman filter-based approaches and RNNs highlights the greater flexibility of RNNs. This flexibility comes with both advantages and challenges: notably, RNN models involve the training of significantly more parameters and the adjustment of numerous hyperparameters compared to the Kalman filter-based models used for state estimation. This increased complexity demands additional time and effort to develop an effective state estimation model. Additionally, a common criticism of deep learning is that the effectiveness of RNNs for state estimation cannot be mathematically guaranteed; instead, like most neural networks, their effectiveness is demonstrated through empirical tests. In contrast, the Kalman filter is founded on rigorous mathematical principles, providing clear theoretical guarantees for its performance. However, for practitioners with extensive experience in deep learning, a well-constructed RNN model has the potential to achieve excellent estimation performance. This is because reliance on rigorous mathematical models often necessitates certain simplifications, whereas RNNs, being inherently more complex, offer greater flexibility for intricate state estimations. Moreover, a state estimation model based on the proposed TL-RNN framework may be better suited to complex structures where FE models, based on existing physical laws, fail to capture the dynamics accurately. In scenarios where model assumptions align closely with actual conditions, TL-RNN may not necessarily outperform existing methods. Conversely, in cases where there is a significant mismatch between model assumptions and real-world conditions, a common occurrence given the complexity of real-world structures, TL-RNN can be a highly effective choice, potentially leading to substantial improvements in prediction accuracy.

Looking ahead, open-source libraries like PyTorch, TensorFlow, and JAX are continuously evolving, providing robust platforms that ease the development, training, and application of neural networks. These tools abstract complex algorithms into user-friendly interfaces, making RNN implementation more accessible to researchers and engineers in structural engineering. Concurrently, hardware improvements, particularly GPU technology, are enhancing computational power for training deep neural networks. This progression not only accelerates the training process but also democratizes access to these advanced facilities. As a result, building and training RNN models, as well as fine-tuning them using the transfer learning approach proposed in this article, are becoming more convenient and effective for state estimation in structural dynamics. This integration of advanced computational tools and methodologies significantly contributes to the precision and efficiency of dynamic state estimation, aligning with the latest trends in data-centric engineering research.

7. Conclusion

In this paper, the role of RNN transfer learning is explored in terms of state estimation. Two original innovations are introduced in this study. First, when multiple types of monitoring information are available, a new paradigm for information integration and utilization in constructing the state estimation model is proposed, which truly achieves the physics-data fusion. Second, contrary to the traditional approach of freezing the latter several layers of the model, the fine-tuning of the RNN model in state estimation requires locking the parameters in the earlier RNN cells. The innovations presented in this study, along with the acknowledgment that discrepancies between FE models and actual structures are inevitable, suggest that integrating the constructed physics-based model with measured data is the optimal approach for state estimation. This concept is exemplified by the TL-RNN and TL-BiRNN models developed herein. The effectiveness of these innovations in improving prediction accuracy is demonstrated through two state estimation examples: a numerical example using a linearized shear-type structure as an approximation of a base-isolated building, and an experimental example employing a simplified beam model to represent the laboratory structure. Despite the inherent discrepancies between the FE model and the actual structure, the state estimation results from both numerical and experimental case studies, based on the TL-RNN framework, prove the efficacy of the proposed innovations in enhancing prediction accuracy.

CRediT authorship contribution statement

Shuo Hao: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Hong-Wei Li:** Writing – review & editing, Validation, Supervision, Software, Conceptualization. **Yi-Qing Ni:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Weijia Zhang:** Validation, Investigation. **Lei Yuan:** Investigation, Data curation.

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.

Acknowledgments

The authors gratefully acknowledge Dr. Mujib Adeagbo for his invaluable discussions on system identification and model updating problems. The authors also thank Zhengyang Li for his insightful suggestions during the preparation of this manuscript. The work described in this paper was supported by the start-up fund for research assistant professors under the strategic hiring scheme of the Hong Kong Polytechnic University, Hong Kong (Grant No. P0046770), and a grant from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region (SAR) (Grant No. PolyU 152308/22E). The authors also appreciate the funding support by the Innovation and Technology Commission of the Hong Kong SAR Government to the Hong Kong Branch of the National Engineering Research Center on Rail Transit Electrification and Automation (Grant No. K-BBY1).

Data availability

The datasets generated and analyzed during the current study are available in the TL-RNN-StatePrediction-StructDyn. This repository contains all the necessary code, data, and instructions needed to reproduce the results presented in this paper.

References

[1] M. Torzoni, M. Tezzele, S. Mariani, A. Manzoni, K.E. Willcox, A digital twin framework for civil engineering structures, Comput. Methods Appl. Mech. Engrg. 418 (2024) 116584.

[2] M. Song, S. Christensen, B. Moaveni, A. Brandt, E. Hines, Joint parameter-input estimation for virtual sensing on an offshore platform using output-only measurements, Mech. Syst. Signal Process. 170 (2022) 108814.

[3] D. Teymouri, O. Sedehi, L.S. Katafygiotis, C. Papadimitriou, A Bayesian Expectation-Maximization (BEM) methodology for joint input-state estimation and virtual sensing of structures, Mech. Syst. Signal Process. 169 (2022) 108602.

[4] S. Moradi, S. Eftekhar Azam, M. Mofid, On Bayesian active vibration control of structures subjected to moving inertial loads, Eng. Struct. 239 (2021) 112313.

[5] C. Papadimitriou, C.-P. Fritzen, P. Kraemer, E. Ntotsios, Fatigue predictions in entire body of metallic structures from a limited number of vibration sensors using Kalman filtering, Struct. Control. Heal. Monit. 18 (2011) 554–573.

[6] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, G. Lombaert, An augmented Kalman filter for force identification in structural dynamics, Mech. Syst. Signal Process. 27 (2012) 446–460.

[7] S. Eftekhar Azam, E. Chatzi, C. Papadimitriou, A dual Kalman filter approach for state estimation via output-only acceleration measurements, Mech. Syst. Signal Process. 60–61 (2015) 866–886.

[8] K. Erazo, E.M. Hernandez, Uncertainty quantification of state estimation in nonlinear structural systems with application to seismic response in buildings, ASCE-ASME J. Risk Uncertain. Eng. Syst. Part A: Civ. Eng. 2 (2015) B5015001.

[9] M. Roohi, K. Erazo, D. Rosowsky, E.M. Hernandez, An extended model-based observer for state estimation in nonlinear hysteretic structural systems, Mech. Syst. Signal Process. 146 (2021) 107015.

[10] P.K. Paul, A. Dutta, S.K. Deb, Comparison of the performance of nonlinear Kalman filter based algorithms for state-parameter identification of base isolated structures, Struct. Control. Heal. Monit. 29 (2022) e3029.

[11] M. Impraimakis, A.W. Smyth, An unscented Kalman filter method for real time input-parameter-state estimation, Mech. Syst. Signal Process. 162 (2022) 108026.

[12] X. Lv, X.Y. Huang, X.G. H., X.H. Su, A matrix-free implicit unstructured multigrid finite volume method for simulating structural dynamics and fluid-structure interaction, J. Comput. Phys. 225 (2007) 120–144.

[13] H.-W. Li, Y.-Q. Ni, Y.-W. Wang, Z.-W. Chen, E.-Z. Rui, Z.-D. Xu, Modeling of forced-vibration systems using continuous-time state-space neural network, Eng. Struct. 302 (2024) 117329.

[14] H.-W. Li, S. Hao, Y.-Q. Ni, Y.-W. Wang, Z.-D. Xu, Hybrid structural analysis integrating physical model and continuous-time state-space neural network model, Comput.-Aided Civ. Infrastruct. Eng. 40 (2024) 166–180.

[15] S.M. Yang, G.S. Lee, Vibration control of smart structures by using neural networks, J. Dyn. Syst. Meas. Control. 119 (1997) 34–39.

[16] L. Wang, Q. Zhou, S. Jin, Physics-guided deep learning for power system state estimation, J. Mod. Power Syst. Clean Energy 8 (2020) 607–615.

[17] Y. Kumar, P. Bahl, S. Chakraborty, State estimation with limited sensors–A deep learning based approach, J. Comput. Phys. 457 (2022) 111081.

[18] T. Kawai, Y. Kawamura, T. Okada, T. Mitsuyuki, X. Chen, Sea state estimation using monitoring data by convolutional neural network (CNN), J. Mar. Sci. Technol. 26 (2021) 947–962.

[19] R. Madbhavi, B. Natarajan, B. Srinivasan, Graph neural network-based distribution system state estimators, IEEE Trans. Ind. Inform. 19 (2023) 11630–11639.

[20] Z. Shi, Incorporating Transformer and LSTM to Kalman Filter with EM algorithm for state estimation, 2021, arXiv preprint arXiv:2105.00250.

[21] P. Seventekidis, D. Giagopoulos, A. Arailopoulos, O. Markogiannaki, Structural Health Monitoring using deep learning with optimal finite element model generated data, Mech. Syst. Signal Process. 145 (2020) 106972.

[22] S. Kim, H. Shin, Deep learning framework for multiscale finite element analysis based on data-driven mechanics and data augmentation, Comput. Methods Appl. Mech. Engrg. 414 (2023) 116131.

[23] H. Zhu, K. Gao, Y. Xia, F. Gao, S. Weng, Y. Sun, Q. Hu, Multi-rate data fusion for dynamic displacement measurement of beam-like supertall structures using acceleration and strain sensors, Struct. Heal. Monit. 19 (2020) 520–536.

[24] A. Warsewa, M. Böhm, F. Guerra, J.L. Wagner, T. Haist, C. Tarín, O. Sawodny, Self-tuning state estimation for adaptive truss structures using strain gauges and camera-based position measurements, Mech. Syst. Signal Process. 143 (2020) 106822.

[25] Z.-Z. He, L.-X. Zhang, H.-G. Gao, H.-S. Wang, P. Pan, Estimation of the displacement time history of high-rise building structures using limited measurement data and structural information, Mech. Syst. Signal Process. 202 (2023) 110716.

[26] Z. Zhu, J. Lu, S. Zhu, Multi-rate Kalman filtering for structural dynamic response reconstruction by fusing multi-type sensor data with different sampling frequencies, Eng. Struct. 293 (2023) 116573.

[27] G.-R. Liu, S.S. Quek, The Finite Element Method: A Practical Course, Butterworth-Heinemann, 2014.

[28] B. Moaveni, J.P. Conte, F.M. Hemez, Uncertainty and sensitivity analysis of damage identification results obtained using finite element model updating, Comput.-Aided Civ. Infrastruct. Eng. 24 (2009) 320–334.

[29] H.-P. Wan, W.-X. Ren, Parameter selection in finite-element-model updating by global sensitivity analysis using Gaussian process metamodel, J. Struct. Eng. 141 (2015) 04014164.

[30] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2009) 1345–1359.

[31] C.-Y. Chang, C.-W. Huang, Non-contact measurement of inter-story drift in three-layer RC structure under seismic vibration using digital image correlation, Mech. Syst. Signal Process. 136 (2020) 106500.

[32] V.A. Matsagar, R. Jangid, Seismic response of base-isolated structures during impact with adjacent structures, Eng. Struct. 25 (2003) 1311–1323.

[33] V.A. Matsagar, R. Jangid, Influence of isolator characteristics on the response of base-isolated structures, Eng. Struct. 26 (2004) 1735–1749.

[34] S.-R. Yi, J. Song, First-passage probability estimation by Poisson branching process model, Struct. Saf. 90 (2021) 102027.

[35] D.P. Pasca, A. Aloisio, M.M. Rosso, S. Sotiropoulos, PyOMA and PyOMA_GUI: A Python module and software for operational modal analysis, SoftwareX 20 (2022) 101216.

[36] H.-F. Lam, J. Yang, S.-K. Au, Bayesian model updating of a coupled-slab system using field test data utilizing an enhanced Markov chain Monte Carlo simulation algorithm, Eng. Struct. 102 (2015) 144–155.

[37] P. Virtanen, R. Gommers, T.E. Oliphant, et al., SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272.

[38] S.A. Taher, J. Li, Time-delayed joint input and state estimation for systems with and without direct feedthrough in structural dynamics, Mech. Syst. Signal Process. 210 (2024) 111138.

[39] M. Imregun, W. Visser, D. Ewins, Finite element model updating using frequency response function data: I. Theory and initial investigation, Mech. Syst. Signal Process. 9 (1995) 187–202.

[40] M. Gul, F.N. Catbas, Statistical pattern recognition for structural health monitoring using time series modeling: Theory and experimental verifications, Mech. Syst. Signal Process. 23 (2009) 2192–2204.