CIAMTIS

U.S. DOT Region 3 University Transportation Center

Smart Mobile Platform for Model Updating and Life Cycle Assessment of Bridges

October 1, 2022

Prepared by:
S. Pakzad, M. Takac, Lehigh University

r3utc.psu.edu

PennState College of Engineering

LARSON TRANSPORTATION INSTITUTE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CIAM-UTC-REG20		

4. Title and Subtitle	5. Report Date
Smart Mobile Platform for Model Updating and Life Cycle Assessment of Bridges	October 1, 2022
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Shamim N. Pakzad, https://0000-00033127-4453, and Martin Takac, https://0000-0001-7455-2025	
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Lehigh University, 27 Memorial Drive West Bethlehem, PA 18015 USA	11. Contract or Grant No. 69A3551847103
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Research and Innovative Technology Administration 3rd Fl, East Bldg E33-461 1200 New Jersey Ave, SE Washington, DC 20590	Final Report 01/20/2020 – 07/31/2022
	14. Sponsoring Agency Code

15. Supplementary Notes

Work funded through The Pennsylvania State University through the University Transportation Center Grant Agreement, Grant No. 69A3551847103.

16. Abstract

Mobile sensing is an alternative paradigm that offers numerous advantages compared to the conventional stationary sensor networks. Mobile sensors have low setup costs, collect spatial information efficiently, and require no dedicated sensors to any particular structure. Most importantly, they can capture *comprehensive spatial information* using few sensors. The advantages of mobile sensing combined with the ubiquity of smartphones with the internet of things (IoT) connectivity have motivated researchers to think of *cars+smart phones* as large-scale sensor networks that can contribute to the health assessment of structures. Working with mobile sensors has several challenges. The signals collected within a vehicle's cabin are contaminated by the vehicle suspension dynamics; therefore, the extraction of bridge vibration from signals collected within a vehicle is not an easy task. Additionally, mobile sensors simultaneously measure vibration data in time while scanning over a large set of points in space, which creates a different data structure compared with fixed sensors. Since collected data are mixed in time and space, they contain spatial discontinuities. When these challenges are addressed, mobile sensing is a promising data resource enabling crowdsourcing and an opportunity to extract information about infrastructure conditions at an unprecedented rate and resolution. In this regard, deep learning-based frameworks have been developed in this project to (a) resolve the dynamic behavior of a vehicle by estimating input forces to which it is subjected from responses acquired from within a vehicle and (b) learn underlying partial differential equations governing the underlying dynamics of a system from recorded data.

17. Key Words	18. Distribution Statement
Mobile sensing, deep learning, input force estimation, data-driven discovery of governing equations	No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	16	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Table of Contents

List of Figures

CHAPTER 1

Introduction

Mobile sensing is an alternative paradigm that offers numerous advantages compared to the conventional stationary sensor networks. Mobile sensors have low setup costs, collect spatial information efficiently, and require no dedicated sensors to any particular structure. Most importantly, they can capture *comprehensive spatial information* using few sensors. The advantages of mobile sensing combined with the ubiquity of smartphones with the internet of things (IoT) connectivity have motivated researchers to think of *cars+smart phones* as large-scale sensor networks that can contribute to the health assessment of structures.

Working with mobile sensors has several challenges. The signals collected within a vehicle's cabin are contaminated by the vehicle suspension dynamics; therefore, the extraction of bridge vibration from signals collected within a vehicle is not an easy task. Additionally, mobile sensors simultaneously measure vibration data in time while scanning over a large set of points in space, which creates a different data structure compared with fixed sensors. Since collected data are mixed in time and space, they contain spatial discontinuities.

When these challenges are addressed, mobile sensing is a promising data resource enabling crowdsourcing and an opportunity to extract information about infrastructure conditions at an unprecedented rate and resolution. In this regard, this project proposes deep learning frameworks specific to mobile sensing to perform input force identification and learn underlying governing equations of a dynamic system from data.

Input force identification is of great interest among researchers across various disciplines, such as mechanical, structural, and aerospace engineering. For structural modeling and analysis as well as system identification, input estimation is a key component for both computational and experimental implementations (Park et al. 2009). In many realistic situations, measuring applied loads to structures with high accuracy is impractical (for instance wind or traffic loads on buildings and bridges, respectively). In the case of structures subjected to an ensemble of environmental and stochastic load sources, the collective effect of the applied load is usually modeled as a white Gaussian random process, which is a simplification and may adversely influence further analyses. The majority of existing methods for input estimation rely on a prior description of the dynamic system, which limits the application. In addition, unlike solving for the response by a system of differential equations, the inverse problem can lead to issues such as non-unique solutions and high condition numbers. Sanchez and Benaroya (2014) present a comprehensive review of input estimation techniques. Input estimation of vehicle systems is of particular interest for indirect bridge structural health monitoring on the grounds that the vehicle response that can be conveniently collected within the cabin is highly contaminated by vehicle suspension systems. However, the tire-level input is substantially more informative (Eshkevari et al. 2020).

Due to the nonlinearity and complexity of realistic dynamic systems, it is required to design an approach that accomplishes the input estimation with no baseline model or restrictive assumptions. In this project, a recurrent neural network (RNN) framework is developed that is able to learn the nonlinear input to output transformation of dynamic systems and then exploit this information to deconvolve the output. Figure 1 presents a schematic overview of the inference using the proposed framework. In this figure the neural network is represented as an RNN block with an inverted L-shape input; at each time step the RNN block processes the input and output values inside the L-shape binder to predict the one-step backward estimation of the input. This process is repeated until the maximum possible length of the input signal is estimated. In this framework, the input signals are associated to the tire contact point (CP) of a vehicle and cabin signals are systems' outputs. Note that the figure depicts a single input single output (SISO) case in which the number of response channels equals to one. However, in multi-degree-of-freedom (MDOF) systems, the network dimensions adapt accordingly with no substantial change in the proposed structure or the pipeline.

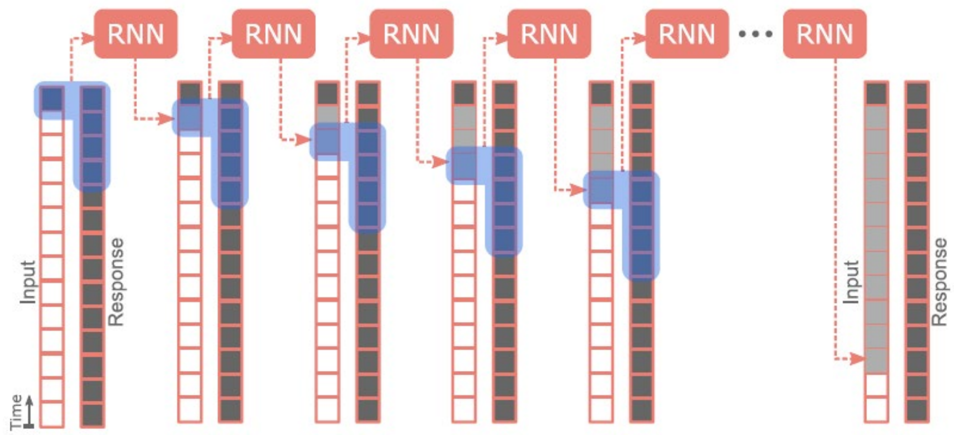

Figure 1. Schematic diagram of the input estimating network.

Partial differential equations (PDEs) are widely adopted in a plethora of science and engineering fields to explain a variety of phenomena such as heat, diffusion, electrodynamics, fluid dynamics, elasticity, and quantum mechanics, to mention a few. This is primarily due to their ability to model and capture the behavior of complex systems as well as their versatility. However, solving PDEs is far from a trivial task. Often incredible amounts of computing power and time are required to get reasonable results, and the methods used can be complicated and highly sensitive to the choice of parameters. The rapid development in data sensing (collection) and data storage capabilities provides scientists and engineers with another avenue for understanding and making predictions about these phenomena. The massive amounts of data collected from highly complex and multi-dimensional systems have the potential to provide a better understanding of the underlying system dynamics.

In this project, inspired by finite-difference approximations and residual neural networks (He et al. 2016), we propose a novel neural network framework, finite difference neural networks (FD-Net), to learn the governing partial differential equations from trajectory data, and iteratively estimate future dynamical behavior. Mimicking finite-difference approximations, FD-Net employs "finite-difference" block(s) (FD-Block) with artificial time steps to learn first-, second- and/or higher-order partial derivatives, and thus learn the underlying PDEs from neighboring spatial points over the time horizon.

BACKGROUND

Among the most notable and recent works, variations of Kalman filtering and Gaussian process latent force modes (GPLFM) have been proposed for estimating inputs of MDOF systems. In the Kalman-based approach, the general strategy is to concatenate the input vector to the state vector in order to build an augmented state and perform state tracking algorithms (Maes et al. 2016). Nayek et al. (2019) proposed a GPLFM-based method for predicting the state-input vectors of a known linear mechanical system. The process was numerically validated on a 10-DOF shear frame and a 76-story building subjected to various excitation scenarios (e.g., impact, harmonic, earthquake, and random) with comparisons to the previous joint input state estimation algorithms. The GPLFM showed high accuracy in most experiments; however, the primary challenge for the application of such methods is their reliance on a baseline model of the structure. This can be partially overcome with preliminary or simultaneous system identification. Nevertheless, the baseline model is not always available, and additional system identification may not lead to a sufficiently accurate surrogate model. Hence, model-free algorithms, such as deep learning models, play as a qualified substitute to model-based methods as this project pursues. In the context of vehicle input force estimation, the first approach to address this inverse dynamic problem was suggested based on deconvolving the response with a simplified linear transfer function (Eshkevari et al. 2020). Yang et al. (2018) proposed a closed-form transfer function-based solution for calculating the contact point (CP) input of an undamped single-degree-of-freedom (SDOF) vehicle suspension model, given its cabin response and the model properties. Yang et al. (2020) tested the direct solution by constructing a single-axle trailer matching the SDOF suspension model. Simplifying assumptions aside, the study demonstrated enhanced bridge modal identification using real data when CP estimates are used instead of the recorded cabin responses. In addition, the improvements lowered the detrimental impact of the sensing vehicle's speed. To expand on these deterministic approaches, Nayek and Narasimhan (2020) proposed a GPLFM-based stochastic method for CP estimation of vehicles for bridge health monitoring. The GPLFM jointly estimates the state and input of a known damping MDOF vehicle system. The proposed model was evaluated using numerical trials and was found beneficial for retrieving information regarding the higher bridge modes that were low-pass filtered by the vehicle transfer function, further supporting the use of CP measurements for indirect bridge health monitoring. However, as stated earlier, these approaches are limited by the requirements of a model. Hence, in this project, the developments in deep learning have been harnessed for the construction of a purely data-driven approach to input force estimation.

OBJECTIVES

The overarching theme of the project was to employ recent advances in deep learning in conjunction with rigorous physics-based foundations to exploit sensed data from SHM applications. The key objectives of the project are as follows:

1. To develop a deep learning-based framework for input force estimation. This allows for estimating tire-level response in vehicles from cabin vibrations, thus facilitating the removal of the impact of vehicle dynamics on recorded responses that will be used for bridge identification and monitoring.
2. To develop a deep neural network-based approach that can identify governing partial differential equation of a dynamic system purely from response data.
3. To validate the performance of the proposed frameworks through numerical and experimental studies. The experiment includes data collected from a real vehicle.

DATA AND DATA STRUCTURES

The project evaluates the performance of the proposed DNN framework through numerical and laboratory experiments. For objective 1, experiments involving sensors deployed on a real car was performed. The details of the setup are provided in the subsequent sections of this report. The installed sensors collect accelerations from inside the cabin and from a location close to the tires. For objective 2, numerical experiments were performed. The raw data are available upon request from the PI as ASCII text files.

CHAPTER 2

Methodology

PROPOSED RNN FRAMEWORK

RNN for input estimation

Overview and network architecture

The state transition equation of a nonlinear time invariant dynamic system can be represented as:

$$x_{k+1} = f(x_k, u_k) + \upsilon_k$$

where x_k is the full state vector of the system at time step k, f(.) is a characteristic function of the system, u_k is the applied load at time step k, and υ_k is the process noise. This state equation usually pairs with an observation equation in which the measurement vectors and full state vectors are related. In the direct dynamic problems, given u_k and initial state x_0, the full state space can be estimated in a data-driven fashion with no need for available estimates of f(.). Recurrent neural architectures have been widely proposed for signal regression tasks due to their capability of learning temporal dependencies and dynamic equations. The main objective is to tackle the inverse problem: given x_k for k∈(1,2,….,T) and a prior estimate for u_T, it is desired to estimate u_k for k∈(1,2,….,T-1). This problem is equivalent to the response deconvolution of a dynamic system (linear or nonlinear) without using the prior knowledge about the system.

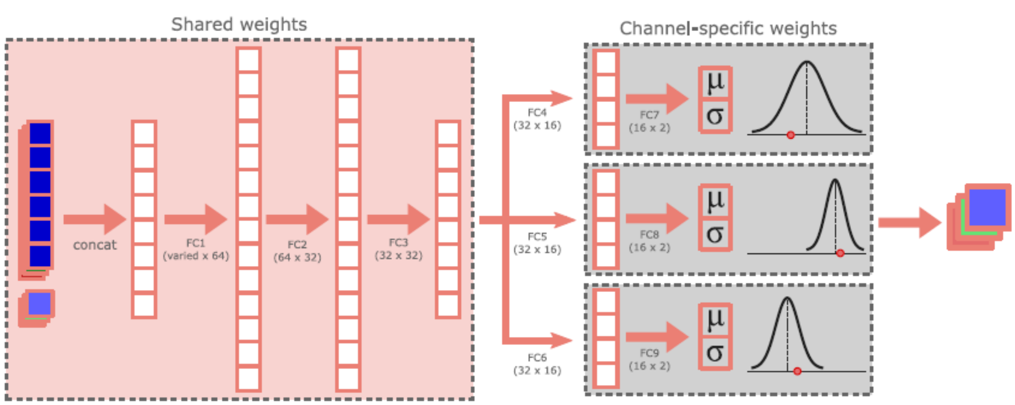

Figure 2. Neural architecture of the RNN blocks.

The structure of the RNN block is given in Figure 2. This architecture incorporates fully connected layers for transitioning between two consecutive time steps. The data flow through two stages in the network: shared layers and channel-specific layers. Channels are defined as input signals collected from different axes (i.e., x, y, z) or at different degrees of freedom. By using the shared layers, the network is constrained to learn information that is equally useful for input estimations in different channels. In addition, the number of learning variables is reduced significantly. In contrast, our preliminary experiments showed that by merely utilizing shared layers, the performance is noticeably lower, suggesting the use of a few channel-specific layers at the terminal end of the network flow. In general, learning-based regression models are designed to directly estimate regressed outputs with no confidence quantification (e.g., estimate scalars or a set of values). In contrast, the estimated outputs of a classifier are class probabilities, which are also useful for uncertainty analyses. For example, in the case of risk-averse problems, by setting higher bars for the classification probabilities, it is possible to enhance the accuracy of the classification predictions at the expense of lowering the recall. Such estimation confidence analyses are not possible when the neural network outputs deterministic values (i.e., regression models). To address that, we introduce a probabilistic learning-based regression model that estimates a normal distribution for the regressed values, instead of estimating actual values. During the training process, the optimization objective is designed to push the mean values of the normal distributions to the actual regression values and shrink the variance for reaching higher confidence. For inference, the means of distributions are considered as actual regressed predictions, while standard deviations indicate the prediction confidence (e.g., low standard deviation means a narrow normal distribution, which translates to high confidence). From Figure 2, the last layer of each channel-specific network predicts μ and σ, constructing the normal distribution for that particular channel. To certify that $\sigma > 0$, the value of the associated output node is passed through an exponential function. The proposed network is termed as a probabilistic regression model, since its final product is not a deterministic regression value but a Gaussian probability distribution (see Figure 2). For better demonstration of the probabilistic nature of the estimations, consider two regression estimations, both yielding the same mean value, but one has lower variance. Given this information and the construction of both Gaussian probability distributions, the model results in a higher probability density value for the prediction associated with the lower variance (i.e., narrower distribution is taller). Therefore, the network not only estimates the regression value but also quantifies its probability. For inference, the output inevitably collapses to the mean values of the distributions to enable the recurrent feedback.

Training

In the training phase, multiple input and output signals from the system of interest are required so that the dynamical system can be learned by the RNN model. This is potentially made possible by temporarily sensing the system's input or using finite element surrogate models for simulation. In the inference stage, however, the only input value that should be available is the input at the terminal state (the systems' input at the last discrete value of the signal). This input value in many applications can be simply set to zero, considering an at-rest condition at the end of the sensing period (e.g., buildings after an earthquake will return to zero acceleration). Given this trivial input state, the system can unravel the previous inputs by processing the outputs that are fully available.

To train the proposed probabilistic regression model, conventional error-minimizing loss functions are not applicable, since these functions incorporate deterministic values rather than distributions. Instead, the loss function has to directly incorporate the negative log likelihood of the observations, given the model parameters. In this context, the RNN block is parameterized by θ and the goal is to maximize the probability of correctly estimating targets y_i given system inputs x_i under the trained parameters. With this definition, the loss function $\mathcal{L}(\theta)$ is defined as the following:

$$\mathcal{L}(\theta) = -log[p(y_i|x_i, \theta)] + \mathcal{L}_{proj}(\theta, n_{proj})$$

where $p(y_i|x_i, \theta)$ is the probability of drawing system input y_i given system output x_i and model parameters θ. The second term of the loss function is the projection loss, with a projection length of n_{proj} and based on the parameterized model $f(\theta)$. As the likelihood term becomes smaller, we ensure that the network's output distributions are more likely to predict values that are close to the actual outputs. The second term of the loss function also attempts to enhance the regression accuracy for longer projections in a conventional mean squared error (MSE) minimization manner. In this term, a strictly increasing geometric factor is element-wise multiplied to the outputs in the trajectory to put more weight on the accuracy of more distant estimations. Further algorithmic details can be found in Eshkevari et al. (2022).

EXPERIMENTAL SETUP FOR VALIDATING RNN FRAMEWORK

Test setup

To demonstrate the efficacy of the proposed network, an experiment was designed and conducted in order to estimate the input of a real-world vehicle using its cabin acceleration data. In this experiment, the data are collected in two locations: inside the vehicle cabin and in proximity to the CP. Note that the actual vehicle's CP is practically inaccessible for a sensor device. Therefore, the lower control arm was selected as a feasible location, and a manually assembled sensor bundle was attached to that location.

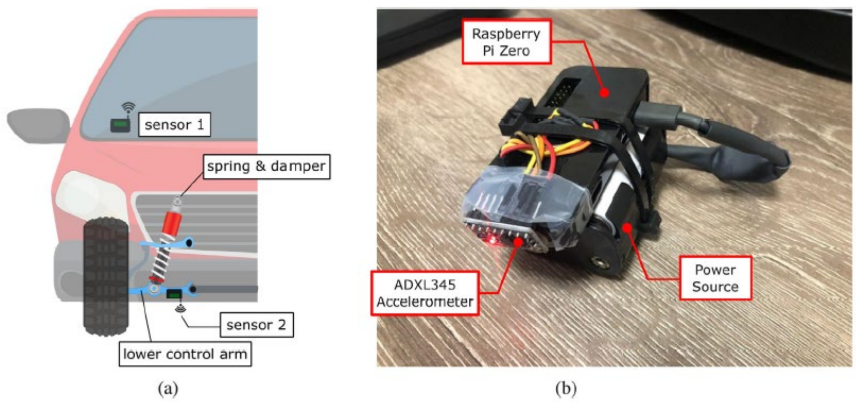

Figure 3. (a) Schematic view of the car and sensor layout; (b) sensor setup used in the experiment: the main board is a Raspberry Pi zero and the sensing device is an ADXL345 accelerometer.

The sensors were wirelessly communicating with a computer, which was held by the operator in the passenger's front seat. The cabin sensor was attached to the dashboard of the vehicle. The sensor layout is presented schematically in Figure 3(a). As shown in the figure, sensor 2 is mounted on the lower control arm, which was found to be a suitable location for the vehicle input data collection and is not affected by the suspension springs. The arm is a solid beam attached to the rim and is located right before the spring and the shock absorber on the load path from the tire to the vehicle cabin. The sensor bundle used for vehicle data collection is shown in Figure 3(b) (a similar configuration is used in both locations). The bundle consists of three components: (1) a Raspberry Pi zero board, (2) an ADXL345 accelerometer, and (3) a power source. The Raspberry Pi was selected for its data processing and storage functionality as well as its low cost, easy programming, and wireless connectivity. ADXL345 is a three-axis accelerometer, which is compatible with Raspberry Pi and collects data with a high rate. The acceleration range and sampling frequency can be tuned based on the application and required accuracy. To select these parameters, a lab-scale experiment was conducted on a single-degree-of-freedom system and the accuracy of the neural network predictions was compared for data collected from different sensor settings. Based on this preliminary study, the sampling frequency of 500 Hz and acceleration range of ±16.0 g were set for the final experimental trial. Note that the adjusted frequency is an upper bound for the sensor, and in practice the sensor collects data with nonuniform time intervals and lower rates. This is found affected by the throughput rate of the Raspberry Pi and its wireless communication.

For the road test, a KIA Forte 2020 was equipped with the sensor sets. According to the vehicle's official specifications, the vehicle suspension is equipped with nonlinear suspension systems in front and rear positions. In particular, the suspension system consists of MacPherson strut and twin tube shock absorbers that both exhibit nonlinear behaviors. The instrumented vehicle was driven over roads with different roughness conditions, including recently paved, poor condition, and gravel roads near Lehigh University campus. In total, 23 scans of 50,000 samples were collected. The vehicle speed was mostly kept within 10–12.5 mph; however, in rare situations of traffic congestion in the testing area, the speed varied. The collected data were then preprocessed for training, which included the following steps: (1) signal resampling in order to even the time intervals between samples, (2) signal filtering using a band-limited filter, and (3) downsampling to 100 Hz. Filtering and downsampling steps reduce high-frequency noise as well as measurement drifts in the collected signals. After preprocessing, signals were normalized linearly using the previously explained approach. This approach for normalization is found to yield better performance compared to other conventional methods (e.g., based on maximum absolute value). The training process of the real-world vehicle experiment is the same as the previous case studies. From 23 scans, 10, 1, and 12 samples were randomly picked for training, evaluation, and testing, respectively. Note that the majority of data were kept for testing for better performance assessment.

DISCOVERY OF GOVERNING EQUATIONS

FD-Net

Overview and network architecture

In this section, we describe the fundamental components of FD-Net. The building blocks of FD-Net are FD-Blocks, whose design is inspired by finite-difference approximations of partial derivatives. Figure 4 shows an instance of FD-Block. An FD-Block is a deep residual learning block that aims to learn the evolution of a dynamical system for one artificial time step on $[t, t + \Delta t]$. It is composed of groups of convolutional layers, a fully connected (FC) layer, and a multi-step skip connection. More details on the architecture and training and testing data can be found in Shi et al. (2020). The algorithm was numerically tested for the heat equation.

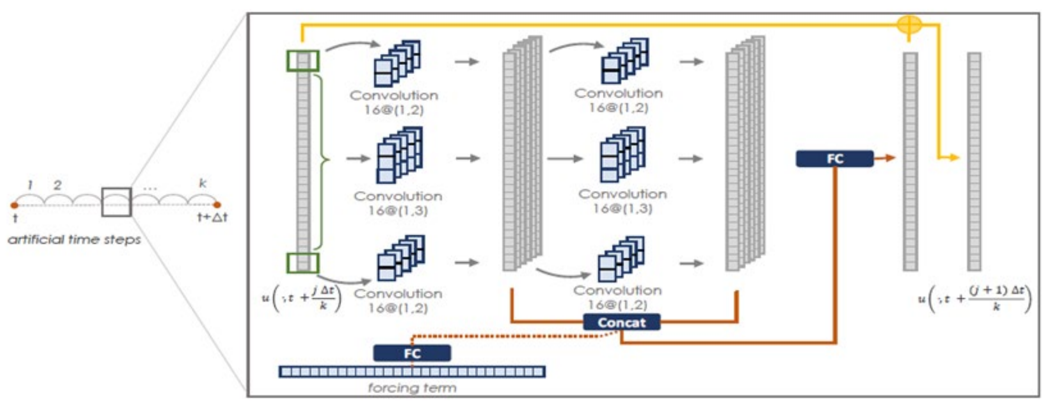

Figure 4. An illustration of a FD-Block.

CHAPTER 3

Findings

RESULTS FOR RNN FRAMEWORK

To evaluate the performance of the network for input estimation, the reconstructed input signals for one of the testing samples are presented in Figure 5. It generally confirms the efficacy of the input estimation in all three axes. The original input signal is highly nonstationary, which is caused by irregular road conditions (such as road bumps or potholes) that complicate the process of learning. Yet, the trained network successfully estimated the overall patterns

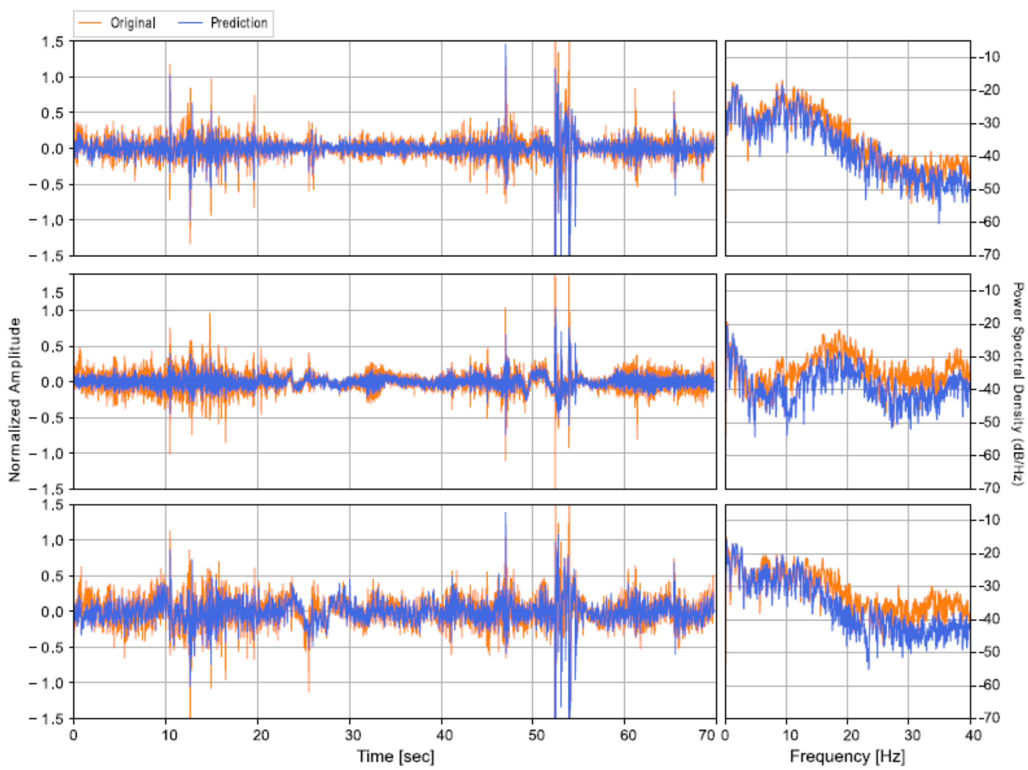

Figure 5. Vehicle input signal predictions in three axes.

RESULTS FOR FD-NET

As stated earlier, FD-Net was tested using the heat equation. The proposed framework's performance was compared to a forward Euler solver. Furthermore, the proposed algorithm was trained using two different optimizers. The popular ADAM optimizer with two different learning rates and 10,000 iterations was used. It is shown that training time can be significantly reduced and the accuracy of the solutions can be drastically improved by using a second-order method. Specifically, second-order Hessian-Free method, Trust-Region (TR) Newton Conjugate Gradient (CG) was employed (Steihaug 1983). The TR method was trained with only 100 iterations. Clearly, the TR-based approach yields the highest accuracy.

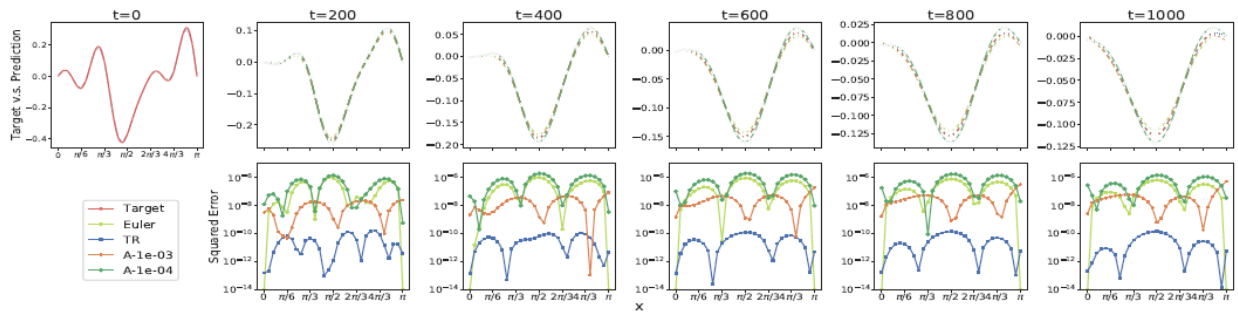

Figure 6. Sequence of predictions along with the squared errors for the forward Euler approach, Trust region-based optimizer approach (TR), and traditional Adam optimizer approach (A followed by the learning rate).

CHAPTER 4

Recommendations

FUTURE DIRECTIONS OF RESEARCH

In this project we demonstrate the efficacy of a DNN-based framework for estimating input forces for a vehicle, thus deconvolving the effects of vehicle dynamics in signals sensed from the cabin of a vehicle. The proposed framework demonstrated its efficacy for both numerical and field data. Furthermore, a DNN-based network was developed that can learn the underlying governing partial differential equation of dynamic systems. Both of these frameworks will help facilitate a mobile sensing paradigm for bridge monitoring.

In the future, the research team plans to further generalize the DNN framework and validate the data collection API by pursuing the following directions:

- Harness the power of further advancements in deep learning that will allow for modeling the various spatio-temporal dependencies of the problem.
- Augment deep learning frameworks with more involved physical principles associated with the problem at hand to enhance performance and facilitate interpretability from a physical standpoint.

References

Eshkevari, S.S., Matarazzo, T. J., and Pakzad, S.N. (2020). Bridge modal identification using acceleration measurements within moving vehicles. *Mechanical Systems and Signal Processing*, 141, 106733.

Eshkevari, S.S., Cronin, L., Eshkevari, S.S., and Pakzad, S.N. (2022). Input estimation of nonlinear systems using probabilistic neural networks. *Mechanical Systems and Signal Processing*, 166, 108368.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Maes, K., Smyth, A.W., De Roeck, G., and Lombaert, G. (2016). Joint input-state estimation in structural dynamics. *Mechanical Systems and Signal Processing*, 70, 445–466.

Nayek, R., Chakraborty, S., and Narasimhan, S. (2019). A Gaussian process latent force model for joint input-state estimation in linear structural systems, *Mechanical Systems and Signal Processing*, 128, 497–530.

Nayek, R., and Narasimhan, S. (2020). Extraction of contact-point response in indirect bridge health monitoring using an input estimation approach. *Journal of Civil Structural Health Monitoring,* 10, 815–831.

Park, J., Ha, S., and Chang, F-K. (2009). Monitoring impact events using a system-identification method. *AIAA Journal*, 47(9), 2011–2021.

Sanchez, J., and Benaroya, H. (2014). Review of force reconstruction techniques. *Journal of Sound and Vibration*, 333(14), 2999–3018.

Shi, Z., Gulgec, N.S., Berahas, A.S., Pakzad, S.N., and Takáč, M. (2020). Finite Difference Neural Networks: Fast Prediction of Partial Differential Equations. In *Proceedings of 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 130-135.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis*, 20(3), 626–637.

Yang, Y.B., Zhang, B., Qian, Y., and Wu, Y. (2018). Contact-point response for modal identification of bridges by a moving test vehicle. *International Journal of Structural Stability and Dynamics*, 18(05), 1850073.

Yang, Y.B., Xu, H., Zhang, B., Xiong, F., and Wang, Z.L. (2020). Measuring bridge frequencies by a test vehicle in non-moving and moving states. *Engineering Structures*, 203, 109859.