Deterministic and low-latency time-series forecasting of nonstationary signals

Chowdhury, Puja, Barzegar, Vahid, Satme, Joud, Downey, Austin R., Laflamme, Simon, et al.

Puja Chowdhury, Vahid Barzegar, Joud Satme, Austin R. J. Downey, Simon Laflamme, Jason D. Bakos, Chao Hu, "Deterministic and low-latency time-series forecasting of nonstationary signals," Proc. SPIE 12043, Active and Passive Smart Structures and Integrated Systems XVI, 120431D (20 April 2022); doi: 10.1117/12.2629025

SPIE.

Event: SPIE Smart Structures + Nondestructive Evaluation, 2022, Long Beach, California, United States

Deterministic and low-latency time-series forecasting of nonstationary signals

Puja Chowdhury*[a], Vahid Barzegar*[b], Joud Satme[a], Austin R.J. Downey[a, d], Simon Laflamme[b, e], Jason D. Bakos[c], and Chao Hu[b]

[a]Department of Mechanical Engineering, University of South Carolina Columbia, Columbia, SC, USA
[b]Department of Structural Engineering, Iowa State University,Ames, IA, USA
[c]Department of Computer Science and Engineering, University of South Carolina Columbia, Columbia, SC, USA
[d]Department of Civil and Environmental Engineering, University of South Carolina Columbia, Columbia, SC, USA
[e]Department of Electrical Engineering, Iowa State University,Ames, IA, USA

ABSTRACT

Hard real-time time-series forecasting of temporal signals has applications in the field of structural health monitoring and control. Particularly for structures experiencing high-rate dynamics, examples of such structures include hypersonic vehicles and space infrastructure. This work reports on the development of a coupled software-hardware algorithm for deterministic and low-latency online time-series forecasting of structural vibrations that is capable of learning over nonstationary events and adjusting its forecasted signal following an event. The proposed algorithm uses an ensemble of multi-layer perceptrons trained offline on experimental and simulated data relevant to the structure. A dynamic attention layer is then used to selectively scale the outputs of the individual models to obtain a unified forecasted signal over the considered prediction horizon. The scalar values of the dynamic attention layer are continuously updated by quantifying the error between the signal's measured value and its previously predicted value. Deterministic timing of the proposed algorithm is achieved through its deployment on a field programmable gate array. The performance of the proposed algorithm is validated on experimental data taken on a test structure. Results demonstrate that a total system latency of 25.76 μs can be achieved on a Kintex-7 70T FPGA with sufficient accuracy for the considered system.

Keywords: real-time, low-latency, fpga, time-series, multi-layer perceptrons, machine learning

1. INTRODUCTION

Microsecond (μs) structural awareness, damage detection, prognostics, and control of structures that experience high-rate dynamics (i.e. shock) would benefit from real-time time-series forecasting of structural responses. Knowledge of future structural response would increase structure survivability in harsh dynamic environments by responding appropriately and adapting mission goals/outcomes to changing conditions. Examples of structures that experience high-rate dynamics include hypersonic vehicles, space infrastructure, and active blast mitigation structures.[1] Hong et al.[2] summarized the high-rate problem as one having:

1. large uncertainties in the external loads;
2. high levels of nonstationarities and heavy disturbances; and
3. generated unmodeled dynamics from changes in system configuration.

Further author information: (Send correspondence to Puja Chowdhury: E-mail:pujac@email.sc.edu

Active and Passive Smart Structures and Integrated Systems XVI, edited by Jae-Hung Han Shima Shahab, Jinkyu Yang, Proc. of SPIE Vol. 12043, 120431D
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2629025

In general, structures that experience high-rate dynamics have acceleration amplitudes higher than 100 g_n for a duration of under 100 ms.

The timing requirements driven by μs structural health monitoring of structures were articulated by Dodson et al. 1 and are presented in Table 1. Based on the dynamics of the considered class of system, a prediction horizon of 1 ms is used for this work. This work experimentally demonstrates that an ensemble of MLPs can be used for time series forecasting at an iteration step of 40 μs, resulting in a new forecasted data point 1 ms into the future every 40 μs.

Table 1. Types and examples of timescales for high-rate monitoring 1.

Time scales of...	Time Scales	Examples
duration of the event	30 µs – 100 ms	structural loading - blast, high-speed impact, automotive crash 2
sensor response	3 µs – 10 µs	accelerometer, strain gage, ect. 3
different physical behavior regimes	250 µs – 1 sec	energy propagation, structural resonance
algorithm execution and decision-making	100 µs – 1 ms	damage detection, uncertainty quantification, state awareness

To enable deterministic and low-latency time-series forecasting 4 of nonstationary signals, an ensemble-based approach was developed that consists of a series of Multi-Layer Perceptrons (MLP) implemented on a Field Programmable Gate Array (FPGAs) 5. The MLPs are trained offline where the proportion of trustworthiness that is associated to the output of any particular MLP is updated online through an attention layer. Through parallelizing the neural networks, the length of the graph is reduced, thereby enabling low-latency inference. In this preliminary work, a series of MLPs are trained offline on dynamics learned from the system. When the system experiences a nonstationarity and transfers to another state the attention layer adapts to the changing dynamics, thereby allowing for a continuous prediction horizon.

This paper describes the creation of a software-hardware system for online structural vibration time-series forecasting that can recognize nonstationary events and alter their anticipated signal in response to them. An ensemble of multi-layer perceptrons is used in the proposed technique, which is trained offline on actual and simulated data relevant to the structure. The outputs of the multiple models are then selectively scaled by a dynamic attention layer to generate a unified anticipated signal over the relevant prediction horizon. The results show that for the system under consideration, a total system latency of less than 1 ms may be attained with appropriate precision. The time consumption 6 for various components of code and device utilization is the major focus for hardware implementation in this preliminary work.

2. METHODOLOGY

Fig. 1 depicts the experimental setup used in this work. A steel cantilever beam measuring 759×50.66×5.14 mm^3 is utilized for the experiment, and a single Integrated Electronics Piezo-Electric (IEPE) accelerometer (model J352C33 produced by PCB Piezotronics) is positioned near the beam's edge. As indicated in Fig. 1, the accelerometer is 0.46 m from the fixed point of the cantilever beam. A 24-bit NI-9234 IEPE signal conditioner from National Instruments is used to digitize the sensor data. An electromagnetic shaker (model V203R from LDS) is used to drive a forced excitation into the beam. The excitation in this work is made up of 100, 120, and 150 Hz sinusoidal waveforms. At t=9.75 s, where a 50% nonstationary is present, two sine wave signals are concatenated together. At 9.75 s, a 50% nonstationary event is introduced, as measured by a 50% rise in the signal's standard deviation. Before t=9.75 s, an input signal of 0.25 V is utilized, and after t=9.75 s, an input signal of 0.375 V is used. The first half of the composite signal is made up of frequencies of 100, 120, and 150 Hz, while the second half is made up of 100 and 120 Hz. This data is available in a public repository 7.

The algorithm consists of an ensemble of MLPs running in parallel, each sampling the incoming observations at a different rate. The use of an ensemble empowers multi-rate sampling to capture multi-temporal features of the time series. Also, the parallel arrangement of the network enables the fast computation times required

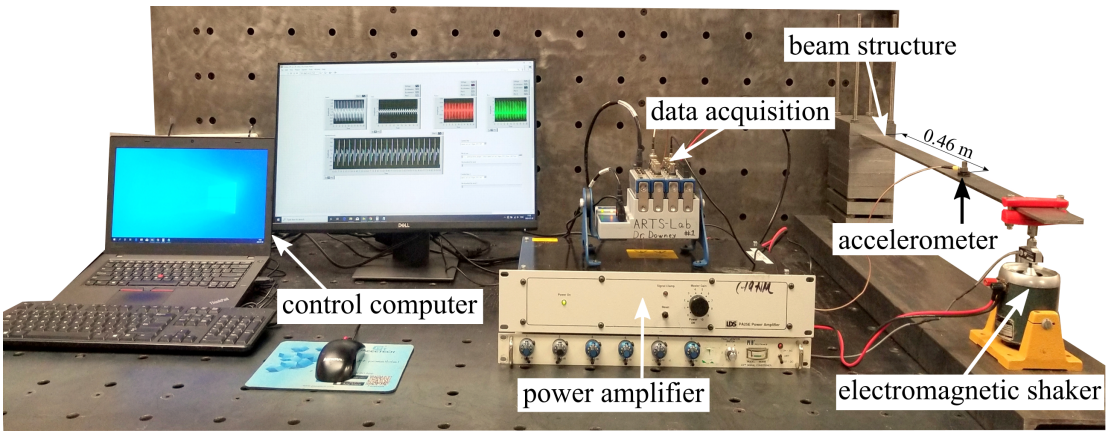

Figure 1. Setup for an experimental cantilever beam including main components and data acquisition.

by high-rate applications. An attention layer combines the output information of the individual MLPs in the ensemble to model the input time series. The architecture of the pre-trained networks connected with the attention layer is illustrated in Fig. 2. MLP i is a multi-layer perceptron network pre-trained to predict with a unique time delay and sequence input. The output of each of the MLPs is a pre-defined feature of the input time series, e.g. a specific frequency of the input. The attention layer scales the output features from the MLPs as the input to another feed-forward network for the target prediction. In this specific case, the attention layer is a single neuron with linear activation.

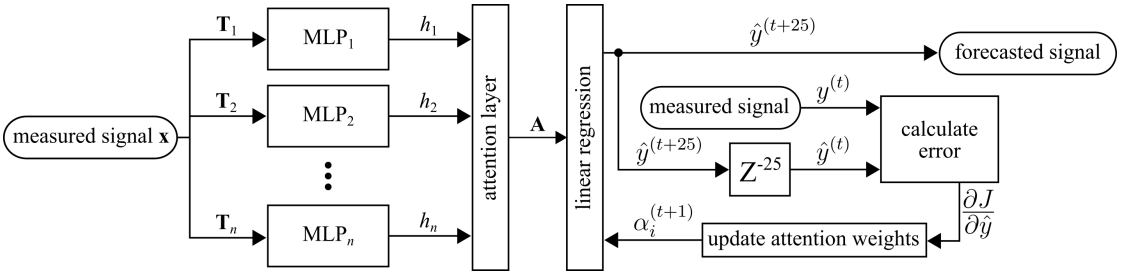

Figure 2. Schematic Algorithm diagram of an ensemble of MLPs using the 50 most recent data points and predicting 25 data points (1 ms) into the future.

The steps for the forward pass of the network for online prediction are as follows:

1. The input to the network is an online stream of observations sampled at f_s.
2. The input to MLP i is a vector $\mathbf{T}_i$ of length m_i constructed by taking every s_i ($s_i \in \mathbb{N}$) observation from the raw input. The length and sub-sampling rate of the inputs are determined according to the desired extracted features of the time-series during the pre-training phase of the individual MLPs. At time step, the input $\mathbf{T}_i$ is of the form:
$$\mathbf{T}_i = \{x_{t-(m_i-1)s_i}, \cdots, x_{t-s_i}, x_t\} \tag{1}$$
3. The input to the attention layer is the output of the individual MLPs. The attention layer assigns a real-valued weight to each of the outputs h_i of MLPs as:
$$\mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \tag{2}$$

where $\alpha_i \in \mathbb{R}$. The output vector of the attention layer is fed into a linear regressor neuron to make a prediction at a 25-step ahead (1 ms) as:

$$\hat{y}^{(t+25)} = \mathbf{W}_{1\times n}\mathbf{A}_{n\times 1} + b \tag{3}$$

where $\mathbf{W}$ and b are the weight matrix of the MLP outputs and the bias scalar, respectively.

4. The loss of the prediction is calculated as:

$$J = \frac{(y-\hat{y})^2}{2} \tag{4}$$

where y is the target value.

The prediction error can be propagated backward in the network to train the attention layer and the linear regressor. The goal is to obtain the gradient of the prediction error with respect to trainable parameters α_i, $\mathbf{W}$, and b, i.e. $\frac{\partial J}{\partial \alpha_i}$, $\frac{\partial J}{\partial \mathbf{W}}$, and $\frac{\partial J}{\partial b}$. Using the chain rule :

$$\frac{\partial J}{\partial \alpha_i} = \frac{\partial J}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial \mathbf{A_i}}\frac{\partial \mathbf{A_i}}{\partial \alpha_i} \qquad \frac{\partial J}{\partial \mathbf{W}} = \frac{\partial J}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial \mathbf{W}} \qquad \frac{\partial J}{\partial b} = \frac{\partial J}{\partial \hat{y}} \tag{5}$$

The chain derivative terms are calculated as:

$$\frac{\partial J}{\partial \hat{y}} = \hat{y} - y \qquad \frac{\partial \hat{y}}{\partial A} = \mathbf{W} \qquad \frac{\partial \mathbf{A}}{\partial \alpha_i} = h_i \qquad \frac{\partial \hat{y}}{\partial \mathbf{W}} = \mathbf{A}^T \tag{6}$$

With the gradients obtained, the parameters are trained with gradient descent method as:

$$\alpha_i^{t+1} = \alpha_i^t - \text{learning\_rate} \times \frac{\partial J}{\partial \alpha_i} \tag{7}$$

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \text{learning\_rate} \times \frac{\partial J}{\partial W} \tag{8}$$

$$b^{t+1} = b^t - \text{learning\_rate} \times \frac{\partial J}{\partial b} \tag{9}$$

3. HARDWARE VALIDATION

In this work, hardware validation is done on a Kintex-7 70T FPGA housed in a NI cRIO-9035 that also incorporates a CPU running NI Linux Real-Time. Fig. 3 diagrams how data is collected and processed on the FPGA 8 , as well as how data is transmitted through the parallel MLP tracks.

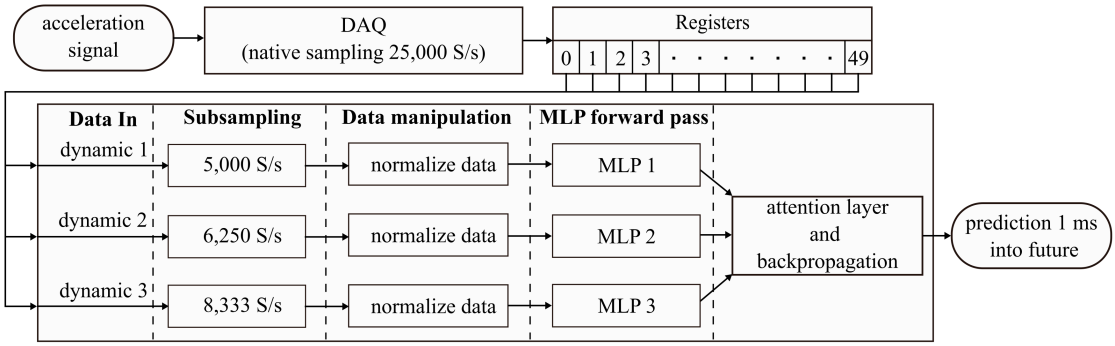

Figure 3. Flow chart of data collection and processing in parallel MLP tracks.

The sampling rate of the system is set to 25,000 S/s and is restricted to intervals of the internal clock of the 24-bit ADC used in this project, a NI-9239 manufactured by NI. Data is passed from the DAQ to a set of 50

registers, stored in the FPGA's look-up table memory. The registers make up a software-defined rolling buffer of the 50 most recent digitized signals. The rolling buffer is sub-sampled at 5,000 S/s, 6,250 S/s, and 8,333 S/s for the three different MLP tracks. The data is then normalized, by detecting maximum and minimum values from input data and ranging the data between -1 to 1. Next, the normalized data is fed through the MLP (i.e. forward pass) to obtain a prediction that is then passed to the attention layer before a final prediction of the signal 25 clock cycles (1 ms) into the future is produced.

4. RESULTS

In validating the proposed algorithm, it is assumed that the input signal was available as prior knowledge based on which the hyper-parameters of the ensemble architecture and training sets of individual MLPs were selected. In this application, three MLPs are selected to represent the three harmonics making the input time series. Three synthetic datasets, each containing a single frequency harmonics with 1,562 Hz, 1,875 Hz, and 2,344 Hz frequency were created and sampled at a similar rate of 25,000 S/s to the target dataset to pre-train the MLPs. The inputs

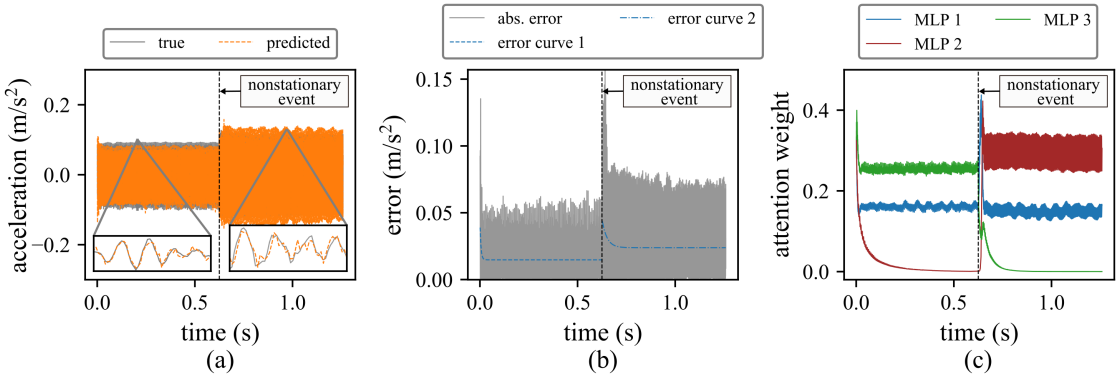

Figure 4. Algorithm results, showing: (a) the truth and prediction result of the ensemble; (b) the absolute error before and after the nonstationary event, and; (c) the evolution of the attention weights for different MLP's.

to the individual MLPs are sampled at four times the oscillating frequency of the corresponding harmonic, i.e. 6,248 Hz, 7,500 Hz, and 9,376 Hz, which translates to $s = \{5, 4, 3\}$ for MLPs 1, 2, and 3, respectively. All MLPs have an input length of 10 ($m = 10$) as well as a single hidden layer with rectified linear unit (ReLU) activation function and four neurons followed by a single neuron output layer with a linear activation function. The MLPs were pre-trained on batches of 10 for 10 epochs with a learning rate of 0.05 to predict 25 steps (1 ms) into the future.

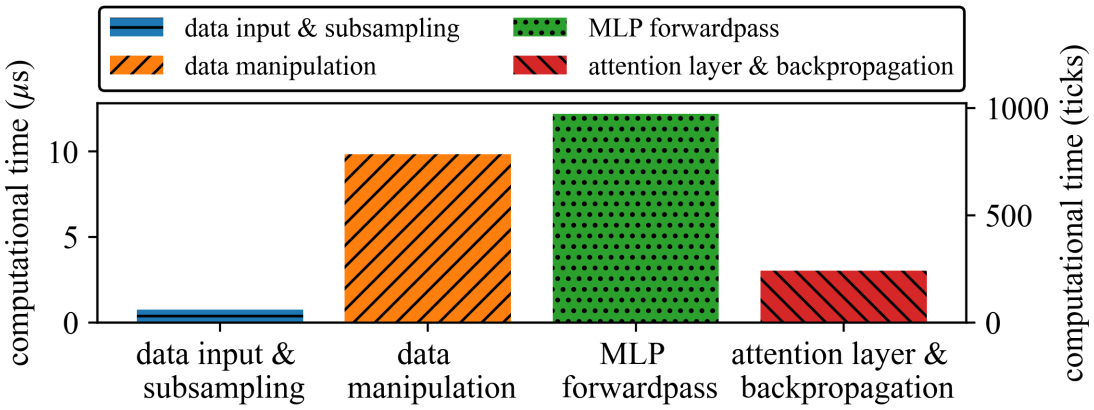

Figure 5. Time required for different aspects of the process.

Figs. 4(a) and (b) show the prediction result of the ensemble along with the error, respectively. The error is maximum at the beginning of the prediction and after the nonstationary event, but it quickly settles into a steady-state error. To measure the convergence of the error at the two locations, an exponential curve of the form $y = a - be^{-ct}$ is fitted to the error before and after the nonstationary event where a is the error floor, b the error amplitude, and c error convergence rate. The fitted curves are also shown in Fig. 4(b) in dashed and dashed-dotted lines. The convergence is defined as the time where the error curve reaches $\pm 5\%$ of the error floor. Table 2 lists the root mean squared error (RMSE), error floor, signal to noise ratio (SNR), and convergence time of the network before and after the nonstationary event.

Table 2. Performance metrics of the predicted results.

	RMSE (m/s^2)	a (m/s^2)	SNR	convergence (ms)
before nonstationary event	0.019	0.015	6.09	18.5
after nonstationary event	0.031	0.024	5.63	71.6

The deterministic characteristics of the algorithm are provided by the FPGA implementation, timing, and resource utilization are discussed here. The FPGA's base clock is compiled at 80 MHz and a single pass through the algorithm takes 2,005 clock ticks (25.76 μs). As a new sample is digitized every 40 μs, the system is dormant for 1,195 clock ticks (14.24 μs) between each iteration as it waits for a new data point to be added to the rolling memory buffer. Fig. 5 reports the timing performance for different aspects of the process. Resource utilization is presented in Table 3, which reports the resource utilization in terms of slices used, slice availability, and percentage (%).

Table 3. The FPGA elements are shown by the device utilization.

	slices used	slices available	percentage used (%)
total slice	9895	10250	96.5
slice registers	36661	82000	44.7
slice LUTs	27917	41000	68.1
block RAMs	19	135	14.1
DSP48s	48	240	20.0

5. CONCLUDING REMARKS

This study outlines the development of a software-hardware system for online forecasting of structural vibration time-series that can learn over nonstationary occurrences and adjust the expected signal accordingly. The proposed technique employs an ensemble of multi-layer perceptrons that are trained offline on simulated data relevant to the structure. The results reveal that a total system latency of 25.76 μs can be achieved with sufficient precision for the high-rate systems under discussion. The key focus for the current hardware implementation is the time consumption for various components of code and device utilization. The current implementation is largely limited by the amount of memory available in look-up tables at the cell level block.

ACKNOWLEDGMENTS

The National Science Foundation provided support for this work through Grants 1850012 and 1937535. The National Science Foundation's support is sincerely thanked. The authors' opinions, results, conclusions, and recommendations in this material are their own and do not necessarily reflect the views of the National Science Foundation. The authors would like to express their gratitude to Iowa State University collaborators for their contributions to this study.

REFERENCES

[1] Dodson, J., Downey, A., Laflamme, S., Todd, M., Moura, A. G., Wang, Y., Mao, Z., Avitabile, P., and Blasch, E., "High-rate structural health monitoring and prognostics: An overview," in [*IMAC 39*], (Febuary 2021).

[2] Hong, J., Laflamme, S., Dodson, J., and Joyce, B., "Introduction to state estimation of high-rate system dynamics," *Sensors* **18**, 217 (jan 2018).
[3] Ueda, K. and Umeda, A., "Dynamic response of strain gages up to 300 kHz," *Experimental Mechanics* **38**, 93–98 (jun 1998).
[4] Tealab, A., "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal* **3**, 334–340 (dec 2018).
[5] Omondi, A. R. and Rajapakse, J. C., [*FPGA implementations of neural networks*], vol. 365, Springer (2006).
[6] Guan, Y., Yuan, Z., Sun, G., and Cong, J., "Fpga-based accelerator for long short-term memory recurrent neural networks," in [*2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*], 629–634, IEEE (2017).
[7] Chowdhury, P., Downey, A., Bakos, J. D., and Conrad, P., "Dataset-4-univariate-signal-with-non-stationarity." https://github.com/High-Rate-SHM-Working-Group/Dataset-4-Univariate-signal-with-non-stationarity (Apr. 2021).
[8] Chang, A. X. M., Martini, B., and Culurciello, E., "Recurrent neural networks hardware implementation on fpga,"