**Proceedings of The ASME 2021 Conference on Smart Materials, Adaptive Structures and Intelligent Systems**
**SMASIS2021**
**September 15, 2021, Virtual, Online**

# SMASIS2021-68338

# TIME SERIES FORECASTING FOR STRUCTURES SUBJECTED TO NONSTATIONARY INPUTS

**Puja Chowdhury**
Department of Mechanical Engineering
University of South Carolina
Columbia, South Carolina 29208
Email: pujac@email.sc.edu

**Philip Conrad**
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina 29208
Email: conradp@email.sc.edu

**Jason D. Bakos**
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina 29208
Email: jbakos@cse.sc.edu

**Austin Downey**
Department of Mechanical Engineering
Department of Civil and Environmental Engineering
University of South Carolina
Columbia, South Carolina 29208
Email: austindowney@sc.edu

## ABSTRACT

*In this paper, a method for real-time forecasting of the dynamics of structures experiencing nonstationary inputs is described. This is presented as time series predictions across different timescales. The target applications include hypersonic vehicles, space launch systems, real-time prognostics, and monitoring of high-rate and energetic systems. This work presents numerical analysis and experimental results for the real-time implementation of a Fast Fourier Transform (FFT)-based approach for time series forecasting. For this preliminary study, a testbench structure that consists of a cantilever beam subjected to nonstationary inputs is used to generate experimental data. First, the data is de-trended, then the time series data is transferred into the frequency domain, and measures for frequency, amplitude, and phase are obtained. Thereafter, select frequency components are collected, transformed back to the time domain, recombined, and then the trend in the data is restored. Finally, the recombined signals are propagated into the future to the selected prediction horizon. This preliminary time series forecasting work is done offline using pre-recorded experimental data, and the FFT-based approach is implemented in a rolling window configuration. Here learning windows of* 0.1*,* 0.5*, and* 1 *s are considered with different computation times simulated. Results demonstrate that the proposed FFT-based approach can maintain a constant prediction horizon at* 1 *s with sufficient accuracy for the considered system. The performance of the system is quantified using a variety of metrics. Computational speed and prediction accuracy as a function of training time and learning window lengths are examined in this work. The algorithm configuration with the shortest learning window (*0.1 *s) is shown to converge faster following the nonstationary when compared to algorithm configuration with longer learning windows.*

## INTRODUCTION

Structures experiencing high-rate dynamics are subjected to 1) large uncertainties in external loads; 2) high levels of nonstationarities and heavy disturbances, and 3) the generation of unmodeled dynamics from changes in system configuration [1]. The development of a real-time monitoring and prediction methodology that observes the current state of a structure and

predicts its future state will enable active structures that can respond to high-rate dynamics in real-time [2]. If an effective framework can be developed, control commands can be initiated to prevent further harm or complete system failure [3]. One key challenge in the development of a real-time monitoring and prediction methodology is its ability to operate through nonstationarities. A nonstationary event is one in which the statistical representation of the signal changes. Stationarities can be classified as weak stationarity, covariance stationarity, or second-order stationarity [4]. If a shift in time does not induce a difference in the distribution form, a time series has stationarity, and therefore, the distribution properties (e.g., mean, variance, and covariance) are constant over time.

There are a variety of cases in which a time series does not stay stationary. If these distribution properties are mishandled, the time series would display nonstationary attributes, which is a significant test for a few fields. The nonstationary time arrangement incorporates time trends, arbitrary strolls (additionally called unit roots), and seasonality. Time trends in a signal can also be thought of as low frequency components with periods longer than the considered data set. Thus, a few methodologies are created to break down the nonstationary attributes. These methodologies can be characterized into two primary sorts: time strategies (e.g., Auto-Correlation Analysis strategy, Regression technique, Seasonal Auto-Regressive Incorporated Moving Average, Break for Additive Trend and Season) and Spectro-Temporal strategies [5]. Spectro-Temporal techniques consider the portrayal of frequency varieties [6].

Time series forecasting of high-rate dynamics is difficult. Specifically, a time series forecasting technique must be robust enough to operate with noisy sensor data. Time series forecasting is performed by studying patterns in a variable (or the relationships between variables), building a model, and using this knowledge to build a model. The model is then used to extrapolate the variable into the future. This demonstrating approach is especially valuable when little information is accessible on the information-producing operation or when there is no agreeable illustrative model that relates the expectation variable to other illustrative factors. Much effort has been committed to the improvement and development of time series forecasting models [7]. The investigation of the time series can be separated into two tasks. The initial task is to acquire the structure and basic knowledge (i.e. dynamics) of the observed information. The subsequent task is to fit a model that will be used to make predictions. Observing past information can be utilized for the examination of the dynamics of a structure under nonstationary inputs as well as prediction of its future dynamics. A standard methodology in dissecting time series is to decompose the monitored variable into the three segments, trend, nonstationary, and residual [8]. For the most part, time series examination can be isolated into univariate and multivariate examinations. Univariate time series includes a period arrangement containing a solitary perception recorded consecutively over time. Multivariate time arrangement is utilized when several time series factors are included, and their connections are to be considered [9]. Common techniques for time series prediction incorporate the sliding window, smoothing, and autoregressive expectation techniques, which are broadly applied in a forecast of high rate dynamics system states, financial turn of events, environmental change, and energy interest. The sliding window technique is similar to the single dramatic smoothing strategy while the smoothing and autoregressive techniques are similar to the two-fold dramatic smoothing technique and the triple outstanding smoothing strategy, respectively [10].

The main aim of this work is to investigate the real-time implementation of time series forecasting over a nonstationary event. In this work, a change in loading is introduced into a cantilever beam structure to generate a nonstationarity event. This is intended to represent a structure subjected to a high-rate dynamic event (e.g. impact) that changes the state (i.e. damage) of the structure. This work presents a numerical analysis for the real-time implementation of a Fast Fourier Transform (FFT)-based approach for time series forecasting. For this preliminary study, a testbench structure that consists of a cantilever beam subjected to nonstationary inputs is used to generate experimental data. For online time series forecasting, the FFT-based approach is implemented in a rolling window configuration. The main contribution of this paper is a investigation into how the FFT-based approach responds before, during, and directly following a nonstationary event, while considering different learning window lengths and assumed computation times. The performance of the system is quantified using a variety of metrics that investigate the quality of the prediction. The FFT-based approach with the shortest learning time achieves the best performance. Here 0.1 s, 0.5 s, and 1 s learning window length have been considered with different computation times simulated. The effect of learning window length in different states is described with MAE (mean absolute time) and transient state. The mean absolute error (MAE) can be used to classify errors that are uniformly distributed [11]. The computation time is an approximation of the actual computation time needed for the FFT, signal extraction, and IFFT. These values are reasonable approximations for actual hardware. The effect of computational time is also analyzed in different states and described with mean error and transient time. The algorithm configuration with the shortest learning window that exceeds the lower bound set by the Nyquist sampling theorem (0.1 s) is shown to converge faster following the nonstationary when compared to algorithm configurations with longer learning windows. The shortest computational time (0.01 s) is also impactful for smaller mean errors in different states and for obtaining the shortest transient time.

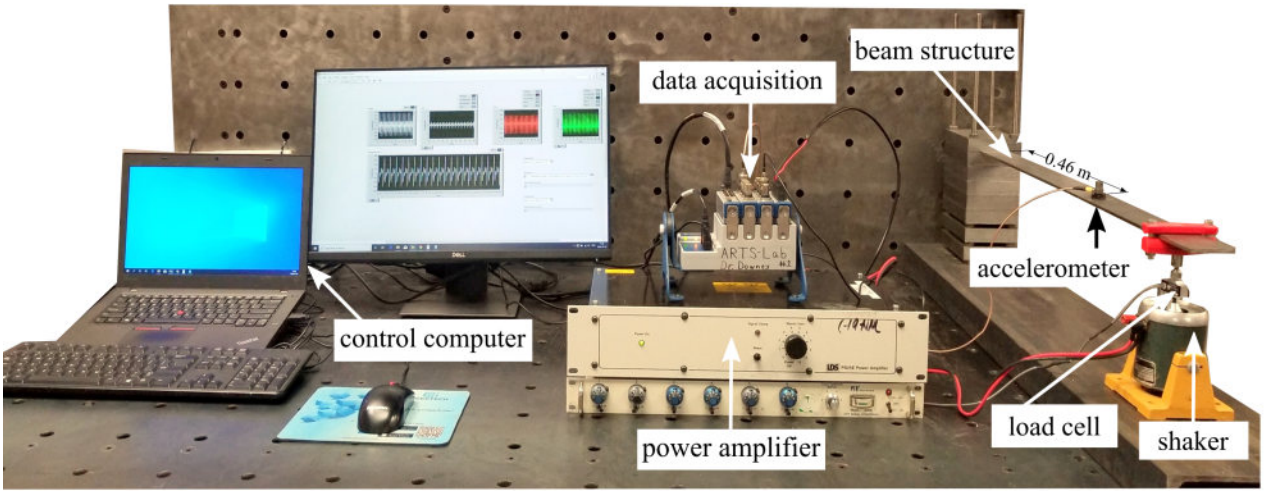


**FIGURE 1**. Experimental setup of a cantilever beam with key components and data acquisition setup.

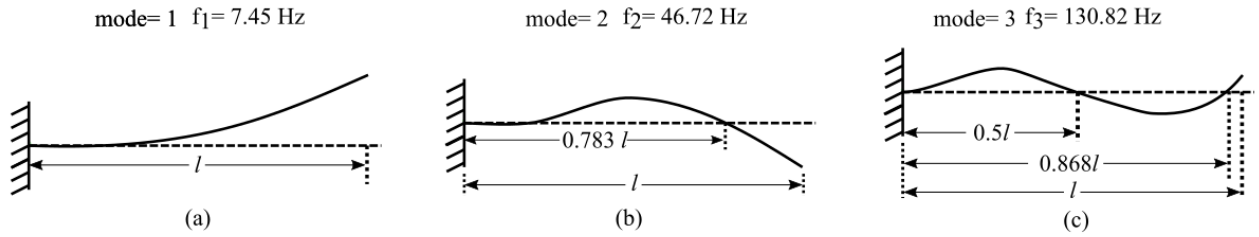


**FIGURE 2**. Mode shapes and frequencies for the cantilever beam setup showing: (a) mode shape 1; (b) mode shape 2, and; (c) mode shape 3.

## EXPERIMENTAL SETUP

The experimental setup is shown in Figure 1. For the purpose of the experiment, a steel cantilever beam structure of 759 x 50.66 x 5.14 mm is used and a single Integral Electronics Piezoelectric (IEPE) accelerometer (model J352C33 manufactured by PCB Piezotronics) is mounted close to the edge of the beam structure.The location of the accelerometer is 0.46 m from the fixed point of the cantilever beam as shown in Figure 1.This accelerometer has a frequency range of 0.5 Hz to 9k Hz with a sensitivity of 100 mV/g. The sensor data is digitized using a 24-bit NI-9234 IEPE signal conditioner manufactured by National Instruments.

To ensure that the accelerometer was not placed at a node of the beam, the mode shapes and natural frequencies for the first three modes of the cantilever were computed via Euler's formula [12] and are shown in Figure 2. The node of the system for the second mode is at 0.594 m while the nodes of the system for the third mode are at 0.380 m and 0.659 m. Therefore, the location of the accelerometer at 0.46 m does not lie directly at any node.

The beam is excited by an electromagnetic shaker (model V203R manufactured by LDS), with a useful frequency range of 5-13000Hz and a peak sine force of 17.8N, and is driven by a power amplifier (model PA25E-CE manufactured by LDS). A 45 N load cell (model MLP-10 manufactured by Transducer Techniques) is mounted in-between the shaker and beam structure. A 24-bit bridge input signal conditioner (NI-9237 manufactured by National Instruments) is used to acquire the load-cell data. The experiment is run through a control computer with a Virtual Instrument written in LabVIEW.

Figure 3 reports the structure's measured acceleration response ($x_v$) for a composite sinusoidal input from the shaker. In this work, the composite signal is made up of 100, 120, and 150 Hz sinusoidal signals. Two sine wave signals are concatenated together at $t$=0 where a 50% nonstationary is present. A 50% nonstationary event is introduced at 0 s, as measured by a 50% increase in the standard deviation of the signal. To achieve this, an input signal of 0.25 V is used before $t$=0 while a signal of 0.375 V is used after $t$=0. The first half of the composite signal is built from 100, 120, and 150 Hz frequencies while the second half signal consists of 100 and 120 Hz frequencies. The entire

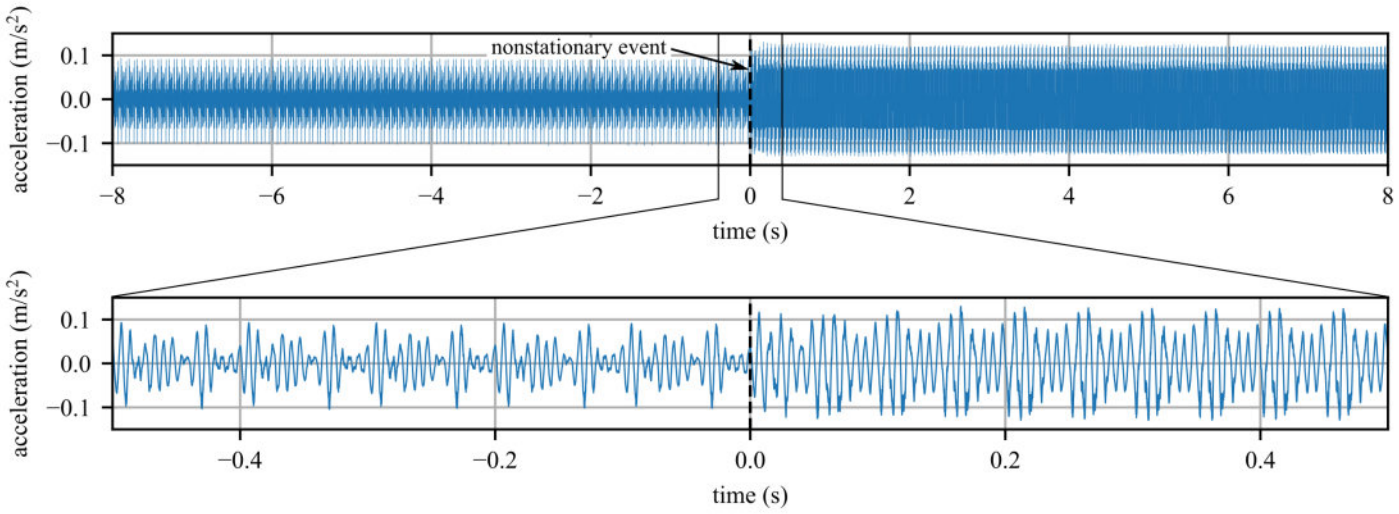


**FIGURE 3**. The full 16-second test is shown in the upper plot while the inset shows the 1 second around the nonstationarity.

16-second test is shown in Figure 3 while the expanded view shows the 1 s around the nonstationary. This data is available in a public repository [13]. For this introductory work, only a harmonic load with a single non-stationary event is considered. Future work will consider non-harmonic loads.

## METHODOLOGY

In this study, periodic structural vibration is analyzed. The measured acceleration signal is $x_v = (x_1, x_2, x_3, \ldots, x_V)$ where $V$ is the total sample points in the observed signal. A rolling window, $x_a$ of size, $N$ moves forward through time as time progresses. By applying the FFT-based time series forecasting method, a signal is generated that is $M$ points long where $M > N$. The difference, $(M - N)$ presents the length of the prediction horizon. By determining $N$ and $M$, this method can be applied to achieve a predicted signal of desirable length. The rolling window is $x_a = (x_{a1}, x_{a2}, x_{a3}, \ldots, x_{aN})$. The first step is to remove any trend line from the acceleration $x_a$. To do this, a polynomial function used where

$$x_{\text{trend}} = p(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_q x^q \tag{1}$$

and $q$ is the degree of the polynomial and $c$ is a set of coefficients. In this work, $q = 1$. After removing the trend, the new acceleration signal without trend is $x = x_a - x_{\text{trend}}$ which has the same sample size as $N$. As considered, the acceleration signal without the trend, $x = (x_1, x_2, x_3, \ldots, x_N)$, is a time series of $N$-samples that the frequency content is extracted from. Therefore, the discrete Fourier transform (DFT) of that series can be expressed as

$$X_k = \sum_{n=0}^{N-1} x_n e^{(-i2\pi(kn/N))} \text{ for } k = 0, \ldots, N \tag{2}$$

where,

$$\omega = 2\pi/N = 2\pi f \tag{3}$$

$$(X_{\text{amp}})_k = |X_k| \tag{4}$$

$$(X_{\text{phase}})_k = X_k/|X_k| \tag{5}$$

Similarly, the inverse DFT can be written as

$$x_n = 1/N \sum_{k=0}^{N-1} X_k e^{(i2\pi kn/N)} \text{ for } n = 0, \ldots, N \tag{6}$$

Now, consider a new series of $M$ length where $M > N$. Using amplitude and phase information, the time series can be constructed and written as

$$x_m = \sum_{k=0}^{M-1} ((X_{\text{amp}})_k \cos(2\pi(\dot{k}m/M)) + (X_{\text{phase}})_k) \text{ for } m = 1, \ldots, M \tag{7}$$

The $x_m$ time series with the trend information added back can be expressed as

$$x_{\text{a\_new}} = x_m + x_{\text{trend}} \tag{8}$$

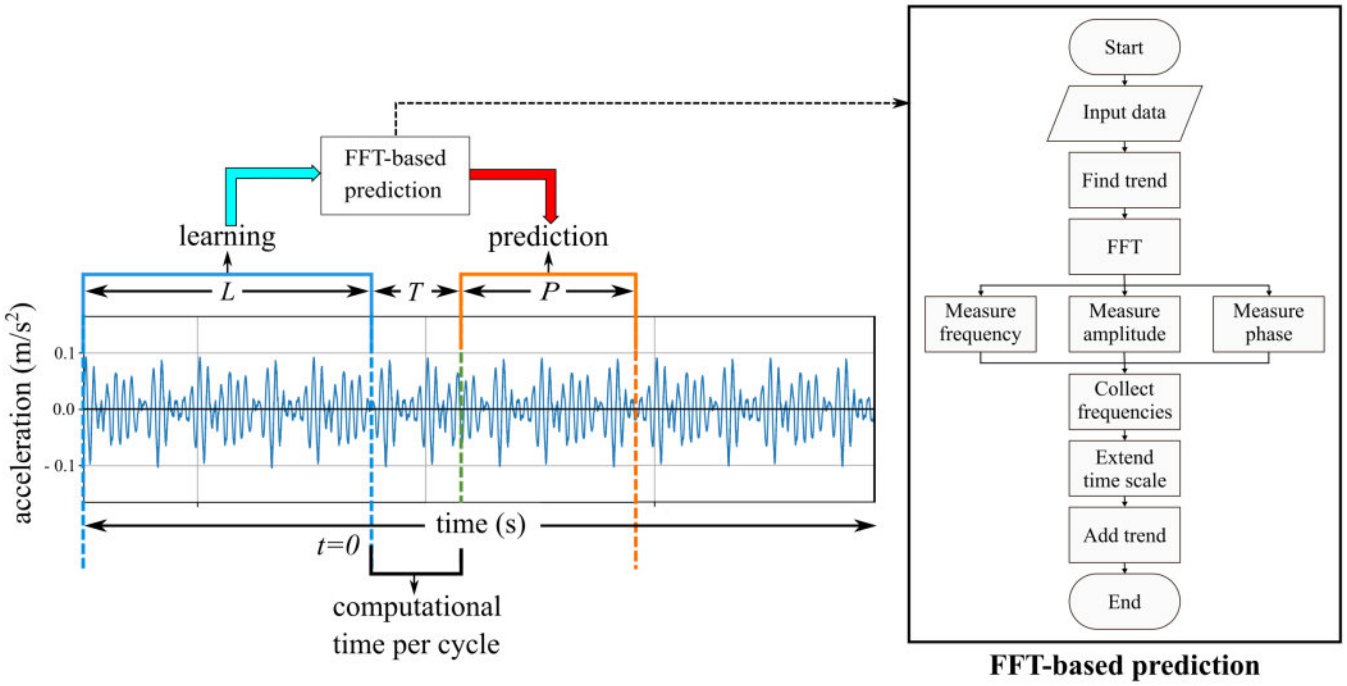


**FIGURE 4**. Schematic Algorithm diagram of FFT-based time series prediction algorithms.

The predicted series would then be $x_{\text{pred}} = (x_{\text{a\_new}(N+1)}, x_{\text{a\_new}(N+2)}, x_{\text{a\_new}(N+3)}, \cdots, x_{\text{a\_new}(M)})$. Frequency domain $X_k$, measures for frequency ($f$), amplitude $(X_{\text{amp}})_k$, and phase $(X_{\text{phase}})_k$. Here $k$ is the index of components in the frequency domain while $n$ and $m$ are the indexes in the time domain for the original series and the new extended series respectively. Thereafter, selective frequency components are collected. For this work, only 28 frequencies are used for propagating the frequencies forward. The selected frequencies are those that have the most energy in the original signal. Here, the list of collected frequencies are given in Table 1

**TABLE 1**. Collected frequencies

| **frequencies (Hz)** | | | | | | |
|---|---|---|---|---|---|---|
| 20 | -20 | 60 | -60 | 70 | -70 | 80 |
| -80 | 100 | -100 | 120 | -120 | 140 | -140 |
| 150 | -150 | 160 | -160 | 170 | -170 | 180 |
| -180 | 200 | -200 | 220 | -220 | 240 | -240 |

For time series forecasting, the FFT-based approach is implemented in a rolling window configuration. In this work, the sliding window length and the prediction window length is 1 s. Here, three lengths of learning windows are used, they are: i) 0.1 s window length, ii) 0.5 s window length, and iii) 1 s window length. The performance of this work has been analyzed with different learning window length ($L$) and various computational times ($T$). Here, four computational times are assumed, intended to model the latency of the FFT, signal extraction, and IFFT on various hardware architectures. These are: i) 0.01 s, ii) 0.1 s, iii) 0.5 s, and iv) 1 s.

The algorithm is diagrammed in Figure 4. The left side of the figure shows how the rolling window is used to enable time series forecasting while the right-side flow chart shows how frequency component extraction and time series prediction is performed. Table 2 shows the parameter values.

**TABLE 2**. Parameter values

| **learning length** | **computational time** | **prediction length** |
|---|---|---|
| $L$ **(s)** | $T$ **(s)** | $P$ **(s)** |
| 0.1, 0.5, 1 | 0.01 | 1 |
| 0.1, 0.5, 1 | 0.1 | 1 |
| 0.1, 0.5, 1 | 0.5 | 1 |
| 0.1, 0.5, 1 | 1 | 1 |

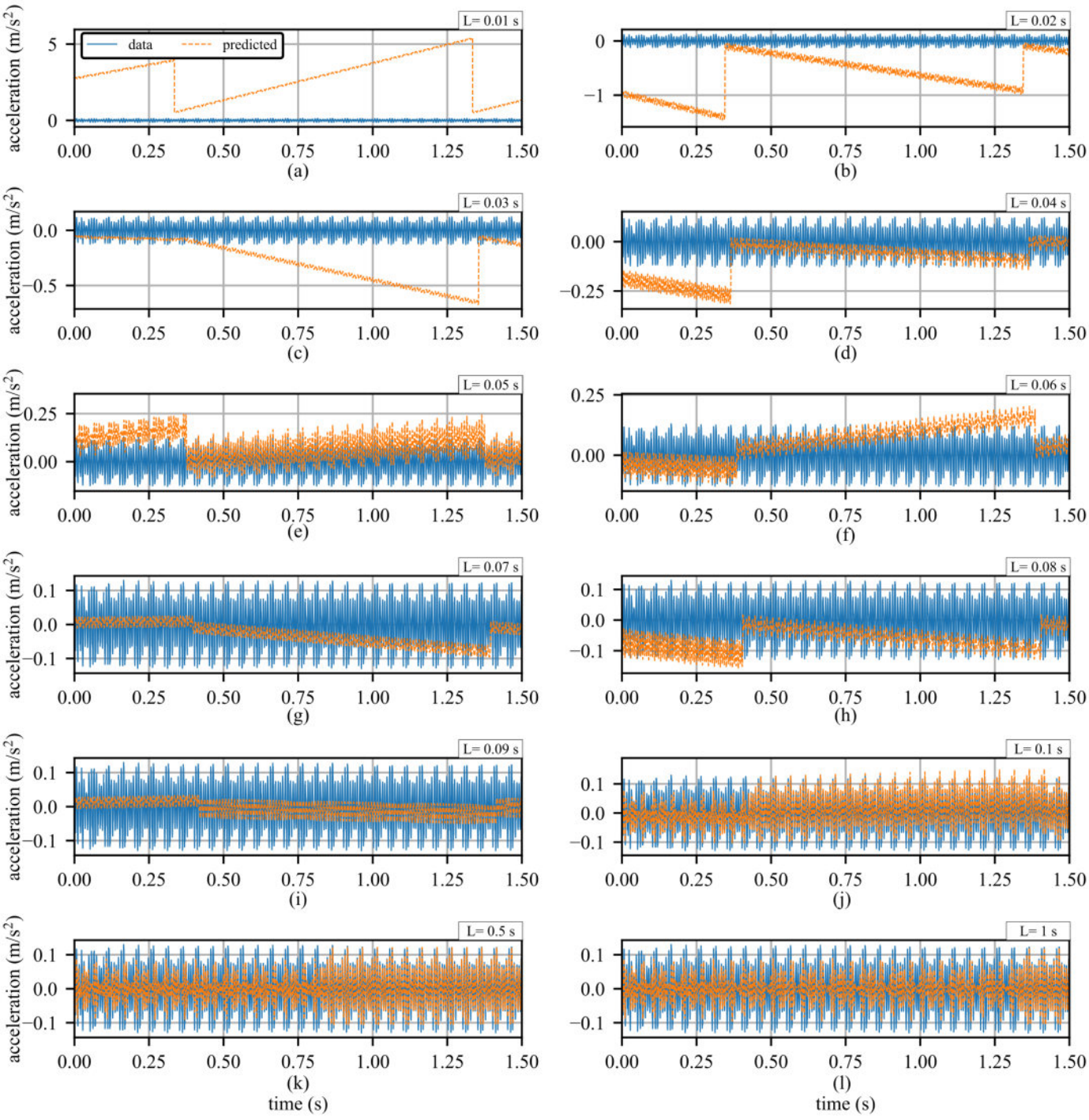


**FIGURE 5**. time series prediction using various learning window lengths showing: (a) 0.01 s window length; (b) 0.02 s window length; (c) 0.03 s window length; (d) 0.04 s window length; (e) 0.05 s window length; (f) 0.06 s window length; (g) 0.07 s window length; (h) 0.08 s window length; (i) 0.09 s window length; (j) 0.1 s window length; (k) 0.5 s window length; and (l) 1 s window length.

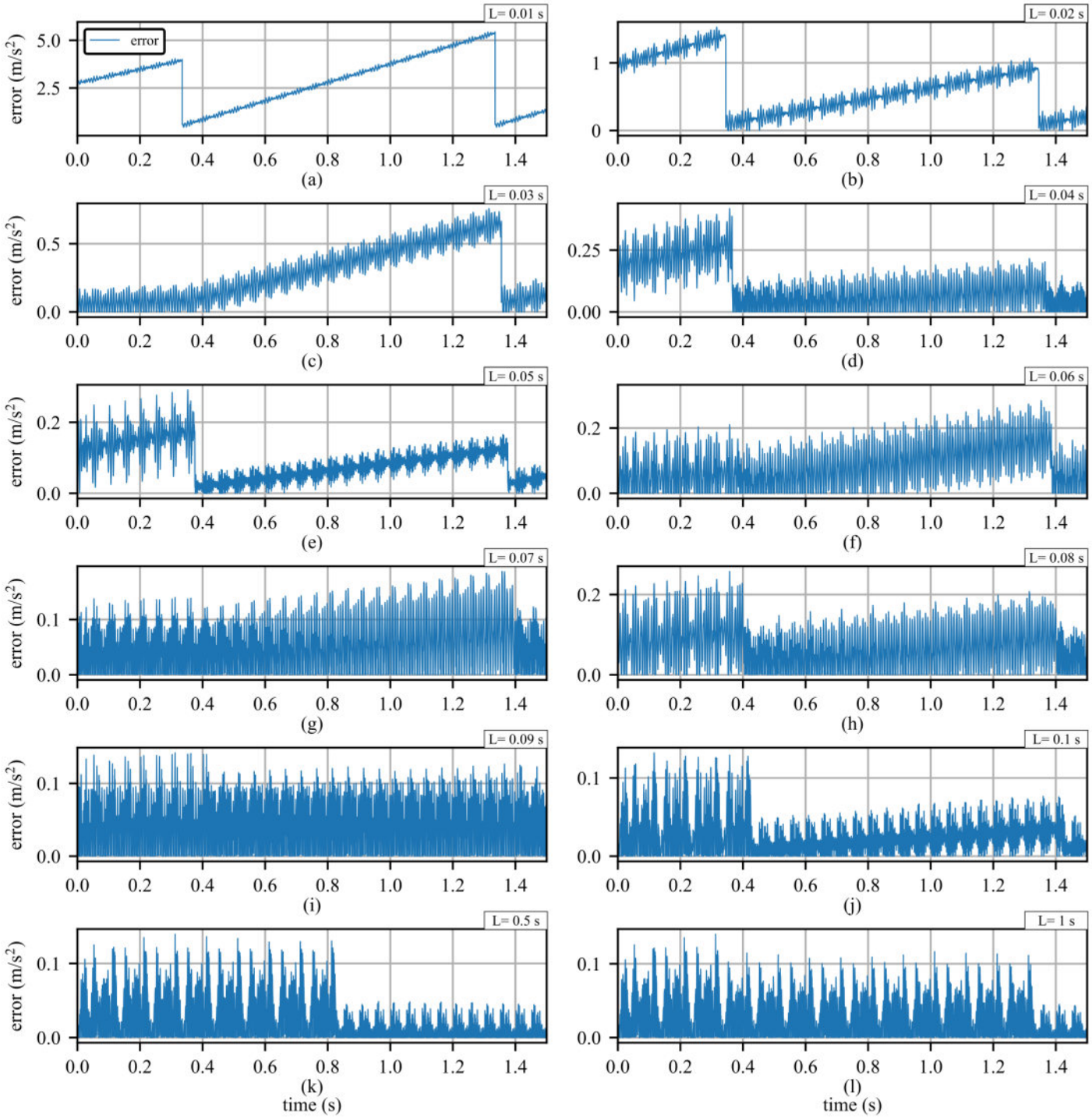


**FIGURE 6**. Calculated instantaneous error over for the experiment data with various learning window lengths showing: (a) 0.01 s window length; (b) 0.02 s window length; (c) 0.03 s window length; (d) 0.04 s window length; (e) 0.05 s window length; (f) 0.06 s window length; (g) 0.07 s window length; (h) 0.08 s window length; (i) 0.09 s window length; (j) 0.1 s window length; (k) 0.5 s window length; and (l) 1 s window length.

## RESULTS AND DISCUSSION

The effect of two parameters on the algorithm have been analyzed: the length of the learning window and the computational time required to perform the FFT-based algorithm.

Figure 5 reports the time series prediction for the FFT-based algorithm for the 12 window lengths considered. Figures 5(a)-(i) show that when the learning window is shorter than the period of the lowest frequency component of the signal, a periodicity in the predicted signal develops. However, when the learning window exceeds the base period in the signal, the periodicity in the predicted signal is removed; as shown in Figures 5(i)-(l). The lowest frequency among the frequencies that compose the signal is 20 Hz, as presented in Table 1. For a 20 Hz frequency, the corresponding period is 0.05 s. Therefore, to capture this frequency, the minimum learning window length needs to be twice the period of the signal, per the Nyquist Theorem. Applying the Nyquist Theorem, the minimum length of the learning window should be 0.1 s. This minimum sampling rate requirement is shown in Figures 5(j)-(l). For accurately capturing all the frequencies listed in Table 1, the minimum period should be higher than the Nyquist limit; for this data set the minimum learning rate is about 0.15 s. As the length of the learning window increases beyond 0.1 s, the quality of the reproduced signal improves. This is shown in Figures 5(k)-(l) where Figure 5 (l) shows the best prediction.

Signal convergence during the transient state is affected by the length of the learning window. As shown in Figure 5(l), the longest learning window length (1 s) takes approximately 1.3 seconds compared to the 0.4 s learning window length in Figure 5(j) which requires only 0.4 seconds to converge. For the purpose of this paper, the system is said to be transient or in a transient state when it has not converged to a steady-state, quantified by looking at the change in error of the system.

The instantaneous (i.e. point-by-point) error for the FFT-based algorithm for the 12 window lengths considered is shown in Figure 6. The periodicity in the predicted signal is removed when the learning window reaches twice the signal's base frequency, as seen in Figures 6(j)-(l). Figure 6(j) shows the error for a learning window length is 0.1 s and has a slight periodic pattern, this is due to it being at the Nyquist limit. In Figures 6(j)-(l), the period of the signal that is in the transient state is denoted by a significantly higher level of error.

Figures 5-6 consider the three learning window lengths of 0.1 s (j), 0.5 s (k), and 1 s (l) for further analysis. Figure 7 and Table 3 report the effects of the selected learning window length on both the MAE and the transient time. Figure 7(a) shows that if the learning window length is increased, the mean error for the considered state decrease. Moreover, Figure 7(b) shows that if the learning length increases, the transient time also increases. Therefore, the mean error and learning window length relationship are inversely proportional while the transient time and learning window length relationship are proportional.

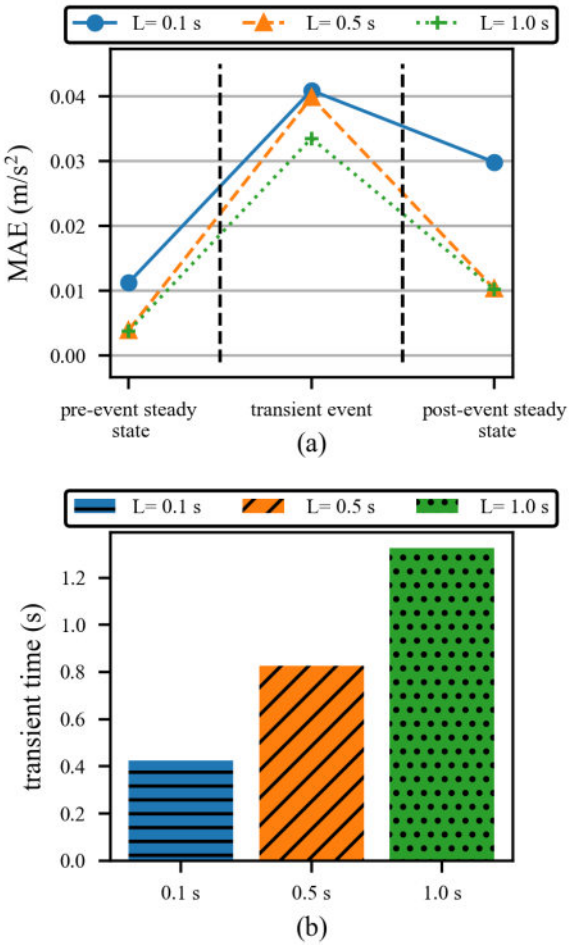


**FIGURE 7**. Effect of various learning window lengths ($L$) showing: (a) MAE in different states, and; (b) transient time.

**TABLE 3**. Performance metrics for various learning window lengths.

| | | learning window length | | |
|---|---|---|---|---|
| | | **0.1 s** | **0.5 s** | **1 s** |
| **State MAE (m/s$^2$)** | Pre-event steady state | 0.0112 | 0.0039 | 0.0038 |
| | Transient event | 0.0409 | 0.0398 | 0.0335 |
| | Post-event steady state | 0.0298 | 0.0103 | 0.0102 |
| Transient time (s) | | 0.42 | 0.82 | 1.32 |

Following the analysis of learning window length, the computational time of the algorithm is considered as another important parameter. Figure 8 and Table 4 display the impact of four simulated computational times (0.01 s, 0.1 s, 0.5 s, and 1 s) for a constant learning length of 0.1 s. The learning length of 0.1 s was chosen as it provides a nice trade-off between the considered

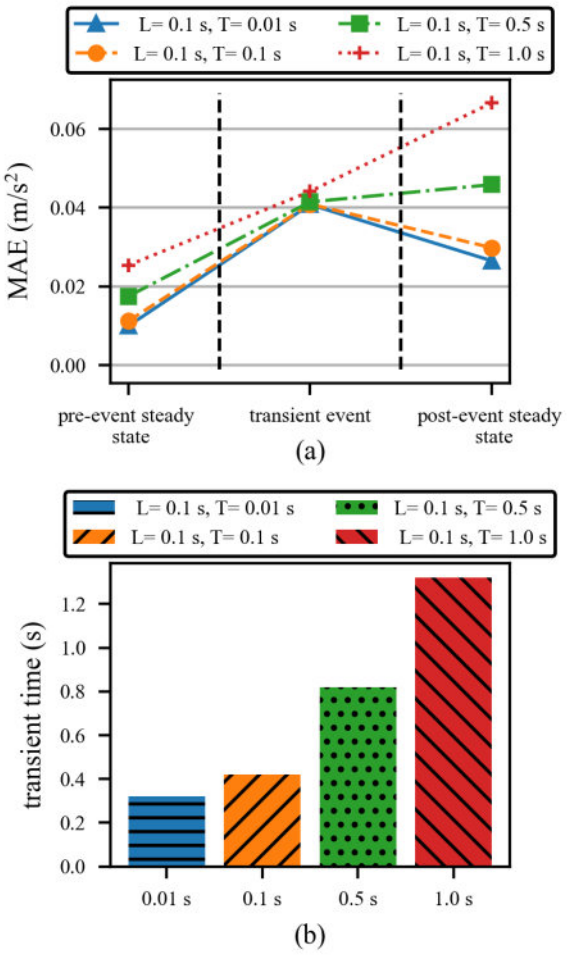


**FIGURE 8**. Effect of various computational time ($T$) in a specific learning window length ($L$) showing: (a) MAE in different states, and; (b) transient time.

metrics of MAE and transient time. Figure 8 and Table 4 show that as the computational time increases, the MAE and transient time increases. This is a proportional relationship. Figure 8(a) depicts the MAE in various states. The MAE across the three computational times varies by less than 3%, this relatively constant error results from the benefits of a shorter computational time only being leveraged when the system experiences a transient event it must respond to. Figure 8(b) and Table 4 shows that the 0.01 s computational time results in a response with less transient time than 1 s.

**TABLE 4**. Performance metrics for various computational times.

| | | computational time | | | |
|---|---|---|---|---|---|
| | | **0.01 s** | **0.1 s** | **0.5 s** | **1 s** |
| **State MAE (m/s$^2$)** | Pre-event steady state | 0.0099 | 0.0112 | 0.0175 | 0.0254 |
| | Transient event | 0.0408 | 0.0409 | 0.0414 | 0.0441 |
| | Post-event steady state | 0.0265 | 0.0298 | 0.0459 | 0.0666 |
| Transient time (s) | | 0.32 | 0.42 | 0.82 | 1.32 |

## CONCLUSION

This work describes a method for forecasting the state of dynamic structures experiencing nonstationary inputs, capable of time series predictions across different timescales. Hypersonic vehicles and space launch systems are the target applications for this system. This work presents a mathematical examination and exploratory outcomes for the continuous execution of a Fast Fourier Transform (FFT)-based methodology for time series forcasting. For offline time series forecasting, the FFT-based approach is implemented in a rolling window configuration. Two types of parameter effects have been analyzed for the algorithm. The length of the learning window and the computational time taken to run the FFT-based algorithm are the two parameters. The effect of learning window length is described concerning mean error and transient state. Learning window lengths are inversely proportional with mean error in different states and proportional with transient time. In the case of mean error, the longest leaning window length of 1 s provides the best performance in the steady-state condition. In the case of transient time or convergence duration, the shortest learning window length that is above the Nyquist limit (0.1 s) performs the best. The relationship between computational time and mean error in different states, as well as transient time, is proportional. The shortest computational time (0.01 s) shows the best performance in MAE and also in transient time. In future work, the FFT-based rolling window prediction method will be implemented in hardware for real-time online time series forecasting.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grants 1850012 and 1937535. The support of the National Science Foundation is gratefully acknowledged. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank collaborators at Iowa State University for their contributions to this work.

## REFERENCES


[1] Hong, J., Laflamme, S., and Dodson, J., 2018. "Study of input space for state estimation of high-rate dynamics". *Structural Control and Health Monitoring,* ***25***(6), p. e2159.

[2] Lowe, R., Dodson, J., and Foley, J., 2014. "Microsecond prognostics and health management". *IEEE Reliab. Soc. Newsl,* ***60***, pp. 1–5.

[3] Connor, J., and Laflamme, S., 2014. *Structural motion engineering*. Springer.

[4] Brockwell, P. J., Davis, R. A., and Fienberg, S. E., 1991. *Time series: theory and methods: theory and methods*. Springer Science & Business Media.

[5] Sang, Y.-F., Wang, D., Wu, J.-C., Zhu, Q.-P., and Wang, L., 2009. "The relation between periods' identification and noises in hydrologic series data". *Journal of Hydrology,* ***368***(1-4), pp. 165–177.

[6] Rhif, M., Ben Abbes, A., Farah, I. R., Martínez, B., and Sang, Y., 2019. "Wavelet transform application for/in non-stationary time-series analysis: a review". *Applied Sciences,* ***9***(7), p. 1345.

[7] Kay, S. M., and Marple, S. L., 1981. "Spectrum analysis—a modern perspective". *Proceedings of the IEEE,* ***69***(11), pp. 1380–1419.

[8] Pang, Y.-H., Wang, H.-B., Zhao, J.-J., and Shang, D.-Y., 2020. "Analysis and prediction of hydraulic support load based on time series data modeling". *Geofluids,* ***2020***.

[9] Brockwell, P. J., Brockwell, P. J., Davis, R. A., and Davis, R. A., 2016. *Introduction to time series and forecasting*. Springer.

[10] Owen, J., Eccles, B., Choo, B., and Woodings, M., 2001. "The application of auto–regressive time series modelling for the time–frequency analysis of civil engineering structures". *Engineering Structures,* ***23***(5), pp. 521–536.

[11] Chai, T., and Draxler, R. R., 2014. "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature". *Geoscientific model development,* ***7***(3), pp. 1247–1250.

[12] Mane, P. D., Yadav, A. A., Pol, A. M., and Kumbhar, V. A., 2018. "Comparative analysis of natural frequency for cantilever beam through analytical and software approach". *International Research Journal of Engineering and Technology (IRJET),* ***5***(2), pp. 656–671.

[13] High-Rate-SHM-Working-Group. Dataset-4 univariate signal with nonstationarity. https://github.com/High-Rate-SHM-Working-Group/Dataset-4-Univariate-signal-with-non-stationarity.