Synthesizing Dynamic Time-series Data for Structures Under Shock Using Generative Adversarial Networks

Zhymir Thompson[1], Austin Downey[1], Jason Bakos[2], Jie Wei[3]

[1]Department of Mechanical Engineering
[2]Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208
[3]Department of Computer Science
The City College of New York, 160 Convent Ave, New York, NY 10031

ABSTRACT
Validation of state observers for high-rate structural health monitoring requires the testing of state observers on a large library of pre-recorded signals, both uni- and multi-variate. However, experimental testing of high-value structures can be cost and time-prohibitive. While finite element modeling can generate additional datasets, it lacks the fidelity to reproduce the non-stationarities present in the signal, particularly at the higher end of the digitized signal's frequency band. In this preliminary work, generative adversarial networks are investigated for the synthesis of uni- and multi-variate acceleration signals for an electronics package under shock. Generative adversarial networks are a class of deep learning approach that learns to generate new data that is statistically similar to the original data but not identical and thus augmenting the data diversity and balance. This paper presents a methodology for synthesizing statistically indistinguishable time-series data for a structure under shock. Results show that generative adversarial networks are capable of producing material reminiscent of that obtained through experimental testing. The generated data is compared statistically to experimental data, and the accuracy, diversity, and limitations of the method are discussed.

Keywords: Time-series, Machine Learning, Adversarial Network, Impact, High-rate Dynamics

INTRODUCTION

A high-rate dynamic event is defined by its time scale of less than 100 milliseconds and encountered high amplitude exceeding 100 g_n [1]. Hypersonic structures, automobiles during accidents, and active blast mitigation are examples of structures that undergo high-rate dynamic events. The goal of high-rate structural health monitoring (HRSHM) is to quickly assess and mitigate changes to a structure caused by high-rate dynamics [2]. One approach to achieve this goal is the development of state observers for tracking the state (i.e. damage) of the structure [3, 4]. The validation of observers for all possible states of high-rate systems is challenging; one reason for this is the inconsistent responses of the structure to these events as structural damage progresses through the life of the structure. As structures fail through a variety of modes, repetition of experiments result in tests with significant variations between tests. A sufficiently large data set that covers the dynamics of the system would allow for the validation of state observers for high-rate systems.

High-rate events can cause plastic deformation to a structure, thereby requiring new structures for each successive experiment. The cost of the experiments themselves is another consideration as the cost of a test program scales with the price for the construction of the structure. Simulating high-rate events is a complex task that requires making various modeling assumptions that may miss key 1st order effects while ignoring various follow-on effects for the sake of stability. Additionally, the

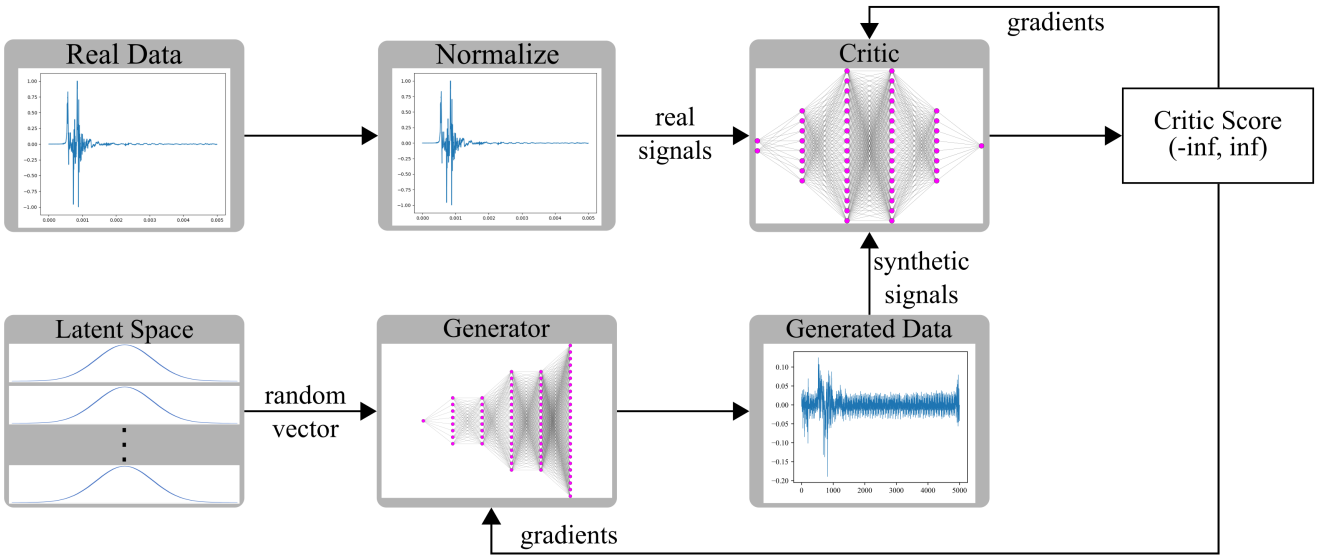

Figure 1: An electronics package developed by Dodson et al. [9] where accelerometers were mounted each unpopulated PCB board housed in axial fixture, and held in place with a single lock ring torqued at 25 ft-lbs; the right-hand-side of the figure shows examples of the measured accelerations for the four PCBs.

introduction of non-Gaussian sensor noise to the system requires the making of various assumptions and it is computationally expensive to rerun tests for each considered noise case. These technical and financial costs limit the potential size of the data set. Utilizing machine learning for the development of high-rate structural health monitoring algorithms requires large amounts of data. Generating a large dataset of examples would require developing an extensive experimental and/or numerical testing campaign. There is a decent likelihood that there exist combinations of variables that are not present in the data set but exist in real-world scenarios for a multivariate problem.

The data size and type limitation can be circumvented through generative models. The type of generative model chosen in this case was the Generative Adversarial Network (GAN). GAN is a deep learning algorithm for the synthesis of statistically indistinguishable data. Collecting real-world data is both expensive and time-consuming, but GANs can mitigate this by producing statistically indistinguishable data based on a library of prerecorded events. In data collections, some events or classes are rare or hard to get, resulting in highly unbalanced data set, which will cause great trouble for future effective training and analysis. GANs can be used to generate data items for these events or classes thus mitigating the annoying unbalanced data problem. Furthermore, GANs can produce combinations of data not found in the original dataset. This means experimental data can be more focused on creating a representative sample and minor variations can be artificially generated in the future. The purpose of the generative model is to augment the existing data to yield a diverse and balanced set of convincing results for use in the training or testing of state observers. GANs have been used to generate audio, radar, optical, and EEG signals [5–8]. As of the writing of this paper, to the best knowledge of the authors, no work involving GANs to produce high-rate dynamic vibration signals have been reported. The contributions of this paper include: 1) a GAN trained on high-rate dynamic vibration data, and 2) development of a multi-modal conditional GAN for multi-class interpolation.

BACKGROUND

Generative Adversarial Networks are a type of deep learning technique capable of producing realistic data similar to, but not the same as, the given set of input data. They accomplish this through a minimax adversarial game in which a discriminator and generator compete [10]. The discriminator attempts to tell apart the real examples and those ”fake” ones produced by the generator. The generator attempts to produce new data out of random noises dictated by the latent space that appears authentic enough to fool the discriminator. This relationship is described with the equation:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \quad (1)$$

Throughout training, the discriminator tries to maximize the function and the generator attempts to minimize it until they reach a balance. Wasserstein GAN (WGAN) is an alternative approach to the typical GAN setup [11]. WGAN transforms the

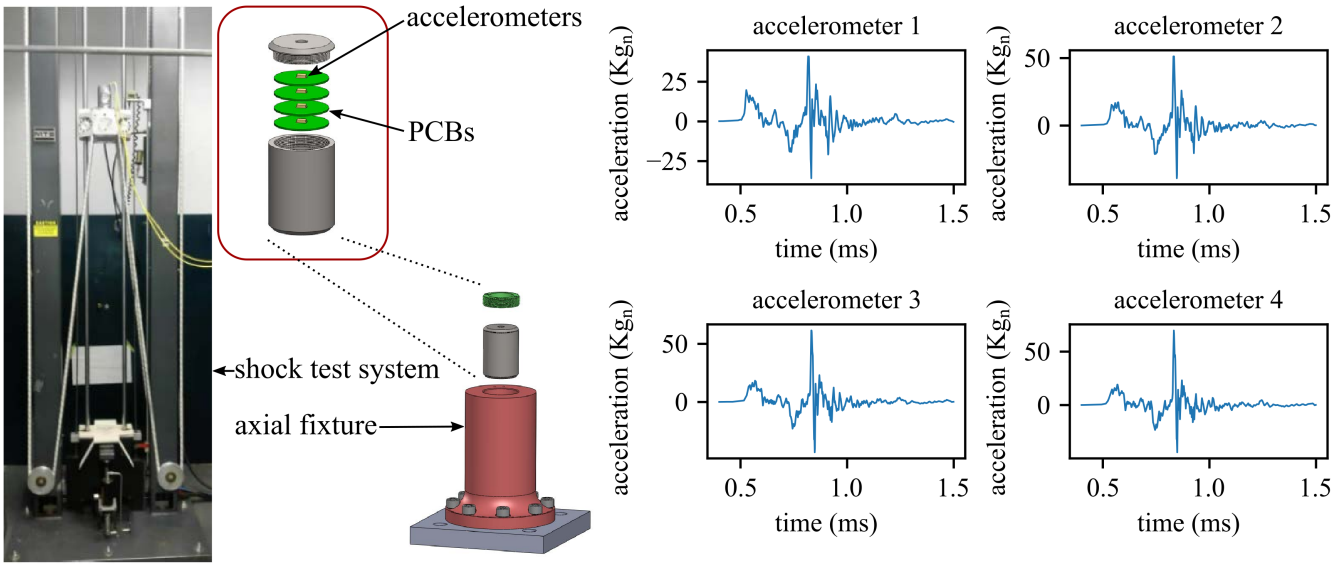

Figure 2: A flowchart of training steps utilized for the proposed GAN methodology.

objective function for the discriminator and generator. Rather than maximizing and minimizing the log-likelihood that a given generated example is not genuine, the discriminator, now called a critic, attempts to minimize an approximation of the Earth Mover Distance while the generator attempts to maximize it [11].

$$f : X \to \mathbb{R} = \max_{\|f\|_{\mathrm{L}} \leq 1} \mathbb{E}_{x \sim \mathbb{P}_{\mathrm{r}}}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\theta}}[f(x)] \tag{2}$$

The change to the objective function converts the gradient slope for backpropagation from a sigmoidal to a linear shape. This change helps prevent convergence failure by vanishing gradients and simultaneously reduces the likelihood of the generative model undergoing mode collapse.

The conditional WGAN (CWGAN) is merely a combination of the conditional GAN (CGAN) and the WGAN where a CGAN is a GAN that takes an additional input(i.e. a class label) for use in the discriminator and generator.

METHODOLOGY

The model architecture used for training was a conditional WGAN composed of CNNs. The discriminator had 8 convolutional layers followed by a single dense layer for reduction. Activation functions were all leaky ReLU and were only applied to the convolutional layers. The conditional data was applied just before the final layer. The generator had an initial dense layer with 7 convolutional layers after. The dense layer had no activation function, and all except for the last two convolutional layers had ReLU activation. The last two had a cosine activation and a tanh activation respectively [5]. Both used the adam optimizer, but the discriminator had a lower learning rate of 0.0001 while the generator had a learning rate of 0.0002.

For model training, the event data was collected from multiple sources after experimentation was complete [9]. The experimental setup is shown in Figure 1. The accelerometers, attached to PCBs, were placed in an axial test fixture for the duration of the experiment. The test fixture was held steadfast to a plate in the shock test system. Each experiment consisted of the system inducing a shock loading to the PCBs that was measured as acceleration. Data was digitized at 1 MS/s for every five-millisecond experiment.

All signals were normalized to be on a scale of [-1, 1]. Due to the limited data size provided [9], the normalized signals were repeated so that there were about 50,000 samples in the training dataset. The generative model was given a 128 point vector for its latent dimension. The models were trained on batches of 16 samples where each batch passed through the critic 5 times, followed by the generator for one iteration. W1 regularization was applied to the critic to stabilize critic scores. Early stopping based on an FFT metric (though the critic score for generator could work) was applied, so training took less than 20 epochs to run.

Figure 2 shows the general flow of the training process. The real data used for training was first normalized to have the same shape but a range between [-1, 1], and the output of the generator was set to have the same range. The condensed range reduced convergence time without significantly hindering the generator's production power since the generated data could be easily scaled back up to the real range in practice. An array of points drawn from a Gaussian distribution were passed to the generator, and the critic loss was calculated on the output from the generator and a batch of training data. The critic was further regularized using w1 regularization wherein the training data regularized the critic [12]. This looped for five iterations before continuing on to train the generator. The generated batch created previously was passed to the newly trained critic since the generator's predictions were deterministic and had not been trained since the generated data was produced. The generator trained on the critic's score of the data. This is a standard training loop for a GAN where the critic can be trained for longer since it avoids the vanishing gradient problem.

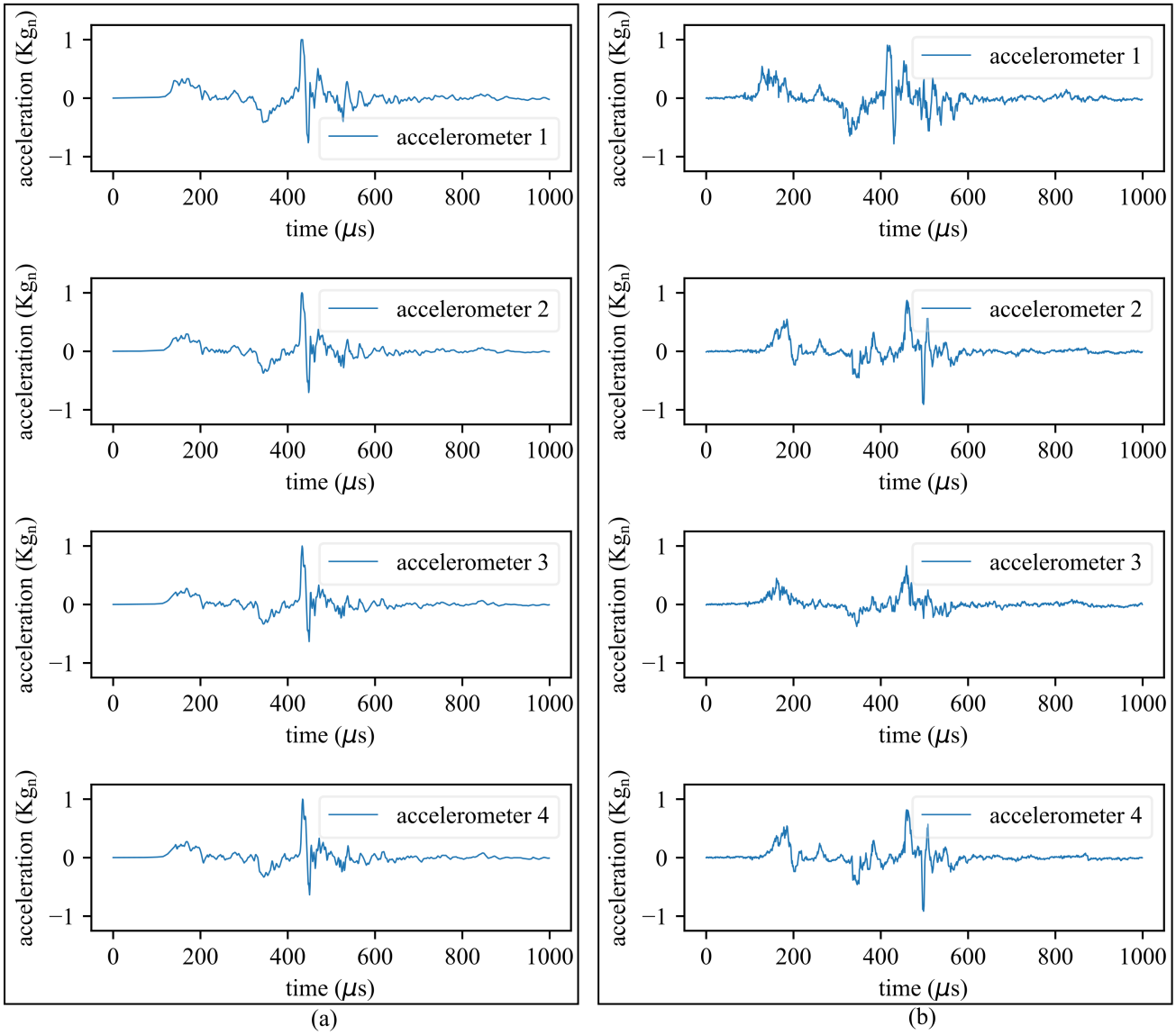

Figure 3: Unimodal generated signals. Each of the four accelerometers has an arbitrary example from the dataset in column (a) and a generated example in column (b).

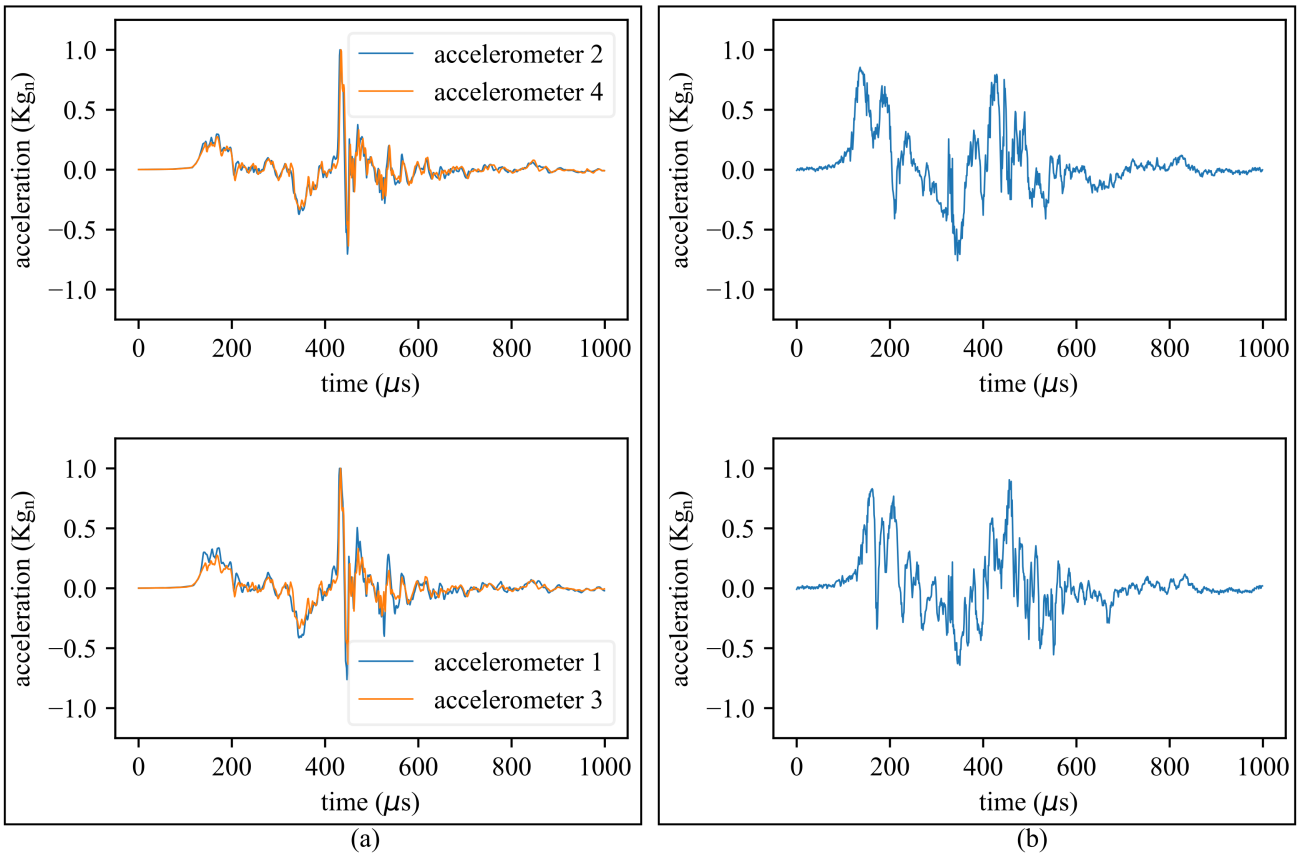

Figure 4: Multimodal generated signals. Column (a) contains examples from the classes used for generation overlaid. Column (b) contains the corresponding generated examples.

ANALYSIS

Evaluating the efficacy of GANs is notoriously difficult as GANs lack an objective function [13]. Furthermore, the existing models for evaluation are mostly used for synthetic image generation and rely on pre-trained models built on that assumption. To evaluate the results of our GAN, an approach using the signals' frequency domain was modified to find the best fit for some given set generated signals in comparison to some set of reference signals [14]. Specifically, the frequency domain for each reference signal and generated signal were computed using the Fast Fourier Transform (FFT). Each generated signal was compared to every reference signal, and the minimum mean squared error (MSE) was saved. Finally, the square root of the average of these differences was taken as the end score. The equation is summarized below:

$$f(x,y) = \frac{1}{m}\sum \min\left(\mathrm{MSE}[\mathrm{FFT}(x) - \mathrm{FFT}(y)]\right) \tag{3}$$

where x is a batch of real samples, y is a batch of generated samples, m is size of y. This score was found to correlate well with the convergence of the generator. While this score can observe accuracy for reproducing a signal, it fails to detect mode collapse, so all generated signals could be virtually identical. The scores in Table 1 were calculated with this scoring method.

Table 1: Scoring for generative model; the total is FFT score over all accelerometer data and each subsequent column is an FFT score for the respective accelerometer data.

	Total	Accelerometer 1	Accelerometer 2	Accelerometer 3	Accelerometer 4
FFT Score	1.496	3.161	3.449	4.017	1.987

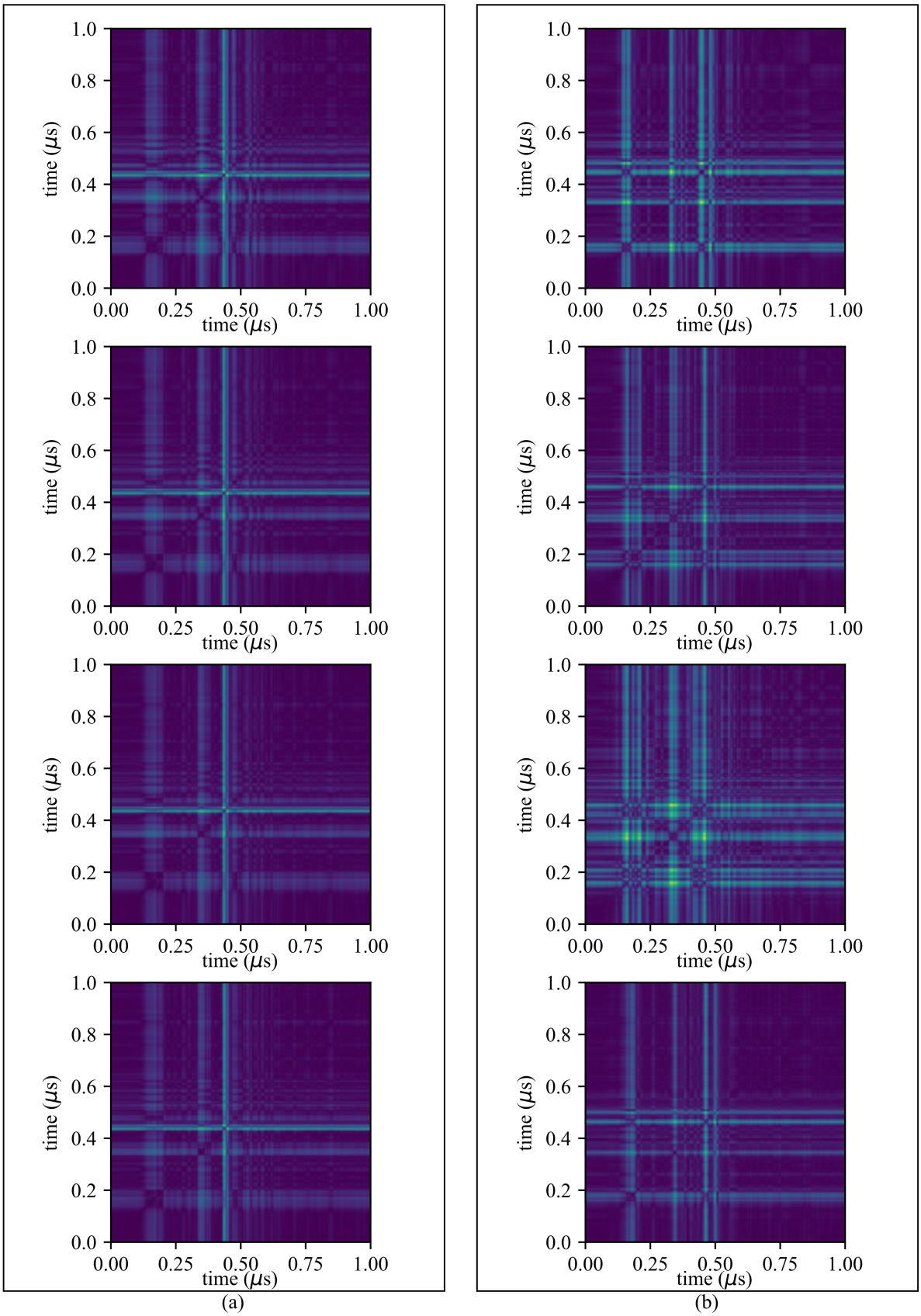

Figure 5: Recurrence plots for generated signals. Column (a) contains examples from the classes used for generation. Column (b) contains the corresponding generated examples.

Along with the FFT metric, samples were manually inspected to determine visual similarity. Manual inspection allowed for identifying potential mode collapse. Figure 3 shows the model's ability for generating examples from single accelerometers. Column (a) shows an arbitrary sample from each accelerometer of the training data, where row 1 corresponds to accelerometer 1 of the data. Column (b) consists of a single generated sample of the acceleration data for each of the four classes used.

The chosen samples provide a context for the generated data rather than a display of accuracy. The generated data now have a comparison for the general shape they were meant to learn. This model was able to produce examples from combinations of accelerometers as shown in Figure 4.

Column (a) of Figure 4 includes an arbitrary example from each accelerometer that the generative model was meant to combine into a single output. The corresponding output is shown in column (b). It is clear that the generative examples contain elements of the training data (oscillations centered around zero, low amplitude left tail, longer medium-low right tail, high amplitude center), so it is clear the model has found some relationship. Figure 4 excels at visualizing relationships, but in place of a multimodal mapping function, it details the relationship between each pairwise set of points in an example.

The plot layout for Figure 5 is the same as in Figure 3, but Figure 5 contains the recurrence plots for the data rather than the data. These recurrence plots are generated by plotting the difference, as a heat-map where points closer together are dimmer, of the acceleration for each given point in time against the acceleration of all other points in time. For example, each plot has a low-valued diagonal from the bottom-left to the top-right since in a pair of accelerations for any point in time with itself the distance would be zero. The important point to note is that similar plots have points with similar relationships, implying their frequency components are similar in the case of these waves.

CONCLUSION

This paper proposes a solution for the lack of consistency for high-rate events via GANs. Using a sample of experiments, the generator can find a mapping across the latent space. This allows for better consistency since a generator given the same input will always return the same output. Given the FFT scores and manual inspection, our model is capable of producing realistic synthetic signals as well as combinations of different signals. The variations between signals of the same class are minor, and variations between classes are not fully representative of the data they originated from. Mode collapse is a recurring problem for some model architecture instances with small batch sizes. The model architecture generated by this work did not show evidence of mode collapse. The results we obtained so far are encouraging. Different choices of latent space dimension, network architecture, dropout rates, optimization method, and learning rate combinations between the discriminator and generator can lead to vastly different results, more investigations along these lines will be explored in the near future.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083 and no. FA9550-21-1-0082. This work is also partly supported by the National Science Foundation Grant numbers 1850012, 1956071, and 1937535. The support of these agencies 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 or the United States Air Force.

REFERENCES

[1] Jonathan Hong, Simon Laflamme, Jacob Dodson, and Bryan Joyce. "Introduction to state estimation of high-rate system dynamics". *Sensors*, 18(2):217, January 2018.

[2] Jacob Dodson, Austin Downey, Simon Laflamme, Michael Todd, Adriane G. Moura, Yang Wang, Zhu Mao, Peter Avitabile, and Erik Blasch. "High-rate structural health monitoring and prognostics: An overview". In *IMAC 39*, Febuary 2021.

[3] Jonathan Hong, Simon Laflamme, Liang Cao, Jacob Dodson, and Bryan Joyce. "Variable input observer for nonstationary high-rate dynamic systems". *Neural Computing and Applications*, 32(9):5015–5026, December 2018.

[4] Austin Downey, Jonathan Hong, Jacob Dodson, Michael Carroll, and James Scheppegrell. "Millisecond model updating for structures experiencing unmodeled high-rate dynamic events". *Mechanical Systems and Signal Processing*, 138:106551, April 2020.

[5] Sangwook Park, David K. Han, and Hanseok Ko. "Sinusoidal wave generating network based on adversarial learning and its application: synthesizing frog sounds for data augmentation". January 2019.

[6] Chris Donahue, Julian McAuley, and Miller Puckette. "Adversarial audio synthesis". February 2018.

[7] Baris Erol, Sevgi Z. Gurbuz, and Moeness G. Amin. "GAN-based synthetic radar micro-doppler augmentations for improved human activity recognition". pages 1–5, Boston, MA, USA, 2019. IEEE.

[8] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. "EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals". June 2018.

[9] Jacob Dodson, Jonathan Hong, and Alain Beliveau. "Dataset 1 high rate drop tower data set", December 2019.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". June 2014.

[11] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". January 2017.

[12] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. "Which training methods for GANs do actually converge?". *International Conference on Machine Learning 2018*, January 2018.

[13] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training gans". *Advances in neural information processing systems*, 29:2234–2242, 2016.

[14] Nicholas Lee. "How to measure the similarity between two signal?", March 2015.