

HYBRID MACHINE LEARNING AND COMPARISON ERROR MINIMIZATION FOR
FREQUENCY DOMAIN-BASED RAPID STATE ESTIMATION IN STRUCTURES
SUBJECTED TO HIGH-RATE BOUNDARY CHANGE

by

James Scheppegrell

Bachelor of Science
University of Florida, 2015

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Nuclear Engineering

College of Engineering and Computing

University of South Carolina

2025

Accepted by:

Austin Downey, Director of Thesis

Junso Lee, Reader

Ann Vail, Dean of the Graduate School

© Copyright by James Scheppegrell, 2025
All Rights Reserved.

ABSTRACT

Dynamic forces and evolving structural boundary conditions pose challenges for various structural systems such as aircraft, orbital infrastructure, and energy harvesting devices. The design, evaluation, and functionality, of such systems can be aided through the collection and analysis of data. However, real-time decision-making for systems experiencing high-rate changes can pose unique challenges, if assessments are to be made accurately and rapidly enough to be relevant.

In cases where the systems are well-defined and thoroughly understood, monitoring the frequency response can be instrumental in determining the state of structures subjected to high-rate structural boundary condition changes. This study focuses on investigating frequency detection methods to enable real-time state estimation for such structures. The research explores progress and findings related to extracting the real-time frequency response of structures.

This study compares a novel technique; Delayed Comparison Error Minimization, a more traditional FFT based method, a trained neural network-based method, and a method combining aspects from the Delayed Comparison Error Minimization and neural network techniques in an attempt to potentially leverage the strengths of each. The performance of each method will be demonstrated, and the results examined and discussed in terms of latency, precision, and possible applicability.

Training of systems that require it will be performed using synthetic data, and the performance of each method demonstrated on a synthetic data set. Each method's performance will also be evaluated on data collected from the DROPBEAR testbed in order to validate performance on a physical system with a changing boundary condition. The DROPBEAR testbed consists of an oscillating beam with one fixed end and a roller support that moves in a controlled manner along the beam's length, altering the frequency response of the beam proportionally to the roller location.

TABLE OF CONTENTS

Abstract.....	iii
List of Figures.....	vi
List of Abbreviations.....	vii
Chapter 1 Introduction.....	1
Chapter 2 Background.....	4
Chapter 3 Materials and Methods.....	6
Chapter 4 Results.....	18
Chapter 5 Discussion.....	28
Chapter 6 Funding.....	30
References.....	31

LIST OF TABLES

Table 4.1: Lag metrics, single sweep, edge effects excluded	22
Table 4.2: Lag metrics, full sweep, time range of interest.....	23

LIST OF FIGURES

Figure 3.1 DROPBEAR testbed	6
Figure 3.2 DCEM Sum vs Delay Step Plots	10
Figure 3.3 DCEM Sum vs Delay Step Plots	11
Figure 3.4 Delayed Comparison and Hybrid intermediate steps	16
Figure 4.1 Frequency-varying signal with target frequency indicated	19
Figure 4.2 Output of frequency estimation methods run on synthetic data set.....	19
Figure 4.3 Estimation methods output, sweep closeup.....	20
Figure 4.4 Lag of methods on single sweep, edge effects included	21
Figure 4.5 Lag of methods on single sweep, edge effects trimmed.....	22
Figure 4.6 Lag of methods, time range of interest on full length synthetic signal	23
Figure 4.7 Roller location and beam acceleration on DROPBEAR	24
Figure 4.8 Output of frequency estimation methods on DROPBEAR	25
Figure 4.9 Output of frequency estimation methods on DROPBEAR, closeup.....	26

LIST OF ABBREVIATIONS

DCEM.....	Delayed Comparison Error Minimization
FFT.....	Fast Fourier Transform
MLP	Multi Layer Perceptron

CHAPTER 1

INTRODUCTION

Many aspects of a structure influence its dynamic response when subjected to an input. As such monitoring that response can allow one to determine if a change in the structure has occurred. In some cases, these systems will be subjected to high-rate dynamic events (defined as changes that occur in under 100 ms [1]). If a condition or state expected to pre-cede a malfunction of the system occurs, detection and reaction prior to failure of the system can improve the outcome. This chain of thought has led to a demand for observers capable of assessing structure or system states, determine the condition from that assessment, and decide what action to take within the possibly very short span of time between the occurrence of a detectable issue and complete failure. If effective general purpose tools with such capability were developed, they would be likely to find use in a wide range of fields and applications where new materials and interface methods are being implement-ed [2], including blast mitigation [3] [4], hypersonic craft [5], and various aspects of machinery and automation.

This paper covers some of the steps in implementing and designing frequency-based observers prioritizing precision and speed, and demonstrates how they function and their performance in a couple of example scenarios. One observer will be an implementation of the Fast Fourier Transform, or FFT. Commonly used for extracting

frequency information from a signal [6], the FFT is a powerful tool with some drawbacks. One, though much reduced [7] [8] compared to calculating the discrete Fourier transform, or DFT, using the definitional equation [9], is computational intensity. Another, more relevant and which is demonstrated here, is the need for long samples in order to achieve high frequency precision. When the Discrete Fourier Transform is calculated using the Cooley-Tukey FFT algorithm [10], for example, the number of frequency bins generated as output is directly de-pendent on the length of the sample used as input. This means that achieving any given frequency precision requirement is inherently a requirement for sample collection time [11] [12], and puts the need for precision in determining frequency at odds with the need for making a decision rapidly based on conditions or states that are changing rapidly compared to the signal frequency.

If the characteristics of a system, and the signals it will generate, are adequately understood, alternative methods of tracking the system response may offer performance benefits compared to an FFT. This work proposes a few methods of tracking frequency, including an explanation of the theory behind each, some details on the implementation, and a comparison of performance between all methods. The results show the performance advantages of the alternate methods compared to an FFT-based approach, including a comparison of output when given data recorded from a physical system in order to demonstrate real-world applicability.

This work has four unique contributions: 1) the proposal of the Delayed Comparison Error Minimization method, 2) a comparison of the Delayed Comparison Error Minimization to two other methods using numerical and experimental data from the DROPBEAR testbed, 3) the proposal of a hybrid version that combines Delayed

Comparison Error Minimization with machine learning techniques to enhance precision and reduce lag, and 4) the introduction of the Lag Error Measurement approach to quantify the response delay of various frequency estimation methods in high-rate dynamic systems.

CHAPTER 2

BACKGROUND

An event is defined as involving high-rate dynamics when it occurs on a time-scale of less than 100 ms, and can include large uncertainties in external loads, high levels of nonstationarities and heavy disturbances, and generation of unmodeled dynamics from changes in system configuration [1]. Prior work by Hong et al. discusses and demonstrates some of the challenges this presents [13], with a series of tests involving circuit boards and accelerometers packaged for shock survivability subjected to high impact conditions using an accelerated drop tower. This prior testing demonstrated the characteristics of a high-rate dynamic event, with the challenges of the system operating on a very short time-scale, and both inconsistent external loading and cumulative damage to the in-ternal structure of the test unit contributing to variation in the dynamics between each event.

While such testing clearly demonstrates the challenges associated with high-rate dynamics, the same test article damage and resulting inconsistencies between test runs that make it a great example can also become hindrances to its applicability in the development and experimental validation of observers designed for rapid state estimation on structures experiencing high-rate dynamic events. Recognizing these

difficulties, Joyce et al. sought to address them by introducing the Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR) [14] testbed, which is described in more detail later in this paper. The controllable parameter changes were designed to produce repeatable effects in the system dynamics, while simulating changes (i.e. damage) occurring in a structural system. As the DROPBEAR's changes are not destructive, test repeatability exceeds that of the accelerated drop tower experiments discussed earlier. Downey et al. used the DROPBEAR to develop a millisecond model-updating technique, comparing an FEA model to the physical system and minimizing error in the frequency domain through model updates. Results were achieved using this method of pin location updates every 4.04 ms with an accuracy of 2.9% [15].

The intent for the DROPBEAR was for it to serve as a generalized representation of various real world systems, with a system response that changes non linearly with the variable input parameter. Downey et al. demonstrated that using an FFT to analyze the accelerometer readings of the DROPBEAR's response to dynamic input presents a challenge in trying to achieve adequate frequency precision without excessive lag [15]. The recorded output of the DROPBEAR was used to guide the generation of synthetic data sets for use in initial testing and development, with certain aspects of the physical data isolated in the synthetic data sets to better understand which characteristics will challenge frequency tracking tools and simplify testing.

CHAPTER 3

MATERIALS AND METHODS

3.1. DROPBEAR Test Bed

The DROPBEAR, shown in Figure 3.1, is a cantilevered beam, with a fixity holding one end of the beam via clamping force and a second support consisting of a pinned condition with continuously adjustable location. The pinned condition is achieved via a pair of rollers in contact with the top and bottom faces of the beam, and a linear actuator moves the location of the pinned condition in a pre-programmed sequence. Excitation of the beam is provided by the motion of the rollers along the beam. The data collected on this system and used in this work is available in a public repository [16].

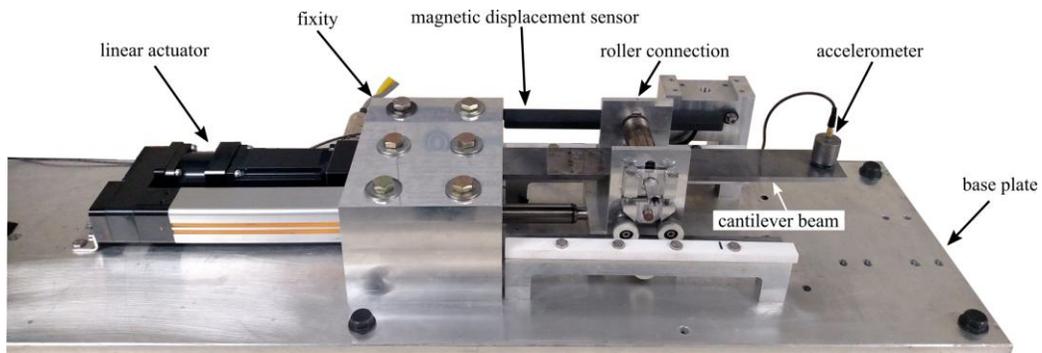


Figure 3.1: DROPBEAR testbed with key components labeled (used with permission [16]).

3.2. Frequency estimation methods

Several methods will be presented; namely FFT using a rolling window, delayed comparison error minimization inspired by Doran's autocorrelation pitch tracking [17], MLP regression using a pre-trained network and data presented directly to the network after only a normalization processing step, and a hybrid method combining portions of the delayed comparison method with an MLP network. Some aspects of the implementation and operation of each will be explained and demonstrated. Demonstration of each method, and comparison between them in terms of the output results and theoretical minimum delay time (the length of time between the earliest data sample used and output of estimate), will be performed. Single frequency identification and the tracking performance, in terms of accuracy and delay, will be demonstrated on a dataset consisting of a waveform containing sections of constant frequency, frequency sweeps and steps. The frequency variation will not exceed 60%, and as such each method will be required to meet a high degree of precision in order to differentiate a useful number of steps within the relevant variation range.

3.2.1 Rolling FFT

A well known, and commonly used, method for converting signals into the time domain is the FFT, or Fast Fourier Transform. The algorithm's output is a series of bins, each containing an amplitude value and the number of which is equal to half as many samples as were provided to it. The bins cover a range from DC for the lowest bin to half the sampling rate of the input for the highest bin, with the remaining bins evenly spaced between. The frequency spacing between bins, which represents the frequency precision

of the method, is equal to the input data's sampling rate divided by how many samples were fed into the FFT.

The value in a given bin is determined by the cumulative amplitude and duration of signal components within that bin's frequency range. Inspecting the magnitude of the frequency bins does not readily allow one to determine the time of an event occurring within the sample. Likewise, the output of the FFT may appear similar for events within the sample that are low in duration and high in amplitude, vs a long duration low amplitude event in another sample. These characteristics [15] are described to emphasize the limitations of applying an FFT to track rapidly varying signal frequencies, transients, or other features or characteristics where frequency components are not present evenly throughout the sample [18].

As the application in this paper is determining changes in frequency response with minimal delay, the FFT is run repeatedly on a short window, each overlapping with the previous but moved in the direction of newer data. The short windows and time difference between them can be thought of as analogous to the shutter speed and frame rate in taking pictures or video; shorter times help to capture rapid changes, longer times will give blurry results unless everything is stationary. Running the frequency estimation using shorter sampling windows and decreasing the time difference between windows can each be helpful in determining when an event occurs, but shorter windows create the drawback of reduced frequency precision and shorter times between sample windows produce diminishing returns as the overlap between windows increases. As any sampling window length must be a compromise between temporal precision and lag vs frequency precision, various window lengths were investigated in earlier work [19]. To note,

increasing the sampling rate of the time series data fed into the FFT does increase the number of bins it will produce, but also extends the frequency response on the high end; the result is that the spacing between bins, and therefore the frequency precision, remains unchanged.

In the method employed in this paper, the fundamental frequency is determined by arranging the values produced by the FFT, excluding those corresponding to frequency bins outside of the range of interest, and checking the frequency bins to see which contains the highest value. In plotting the frequency estimator outputs, the time value for each estimate is taken to be the time of the newest time-series sample used in the FFT; this gives time alignment that would correspond to performing the estimate in real time if there were no computational or other delays introduced.

3.2.2 Delayed Comparison Error Minimization

Delayed Comparison Error minimization, proposed and explained previously in [19], aims to allow high frequency precision even when working with relatively short sections of samples taken from time series data of periodic waveforms. In certain circumstances, it can estimate the primary frequency of a signal with greater precision and less lag than is possible when using the FFT approach implemented here. The method works by comparing many pairs of samples taken from a signal, with each pair having a known time difference between them, and the complete set of pairs spanning a time difference proportional to the range of signal periods of interest. If a strong periodic component is present in the signal, the time difference of the pair where the samples are most similar should indicate that signal period [19].

As implemented in this paper, the two samples that form a pair are each treated as a list of values, and are compared by subtracting each value within one sample from the corresponding value within the other sample of the pair. The result is a difference list equal in length to either sample; the individual values within this list are then squared,

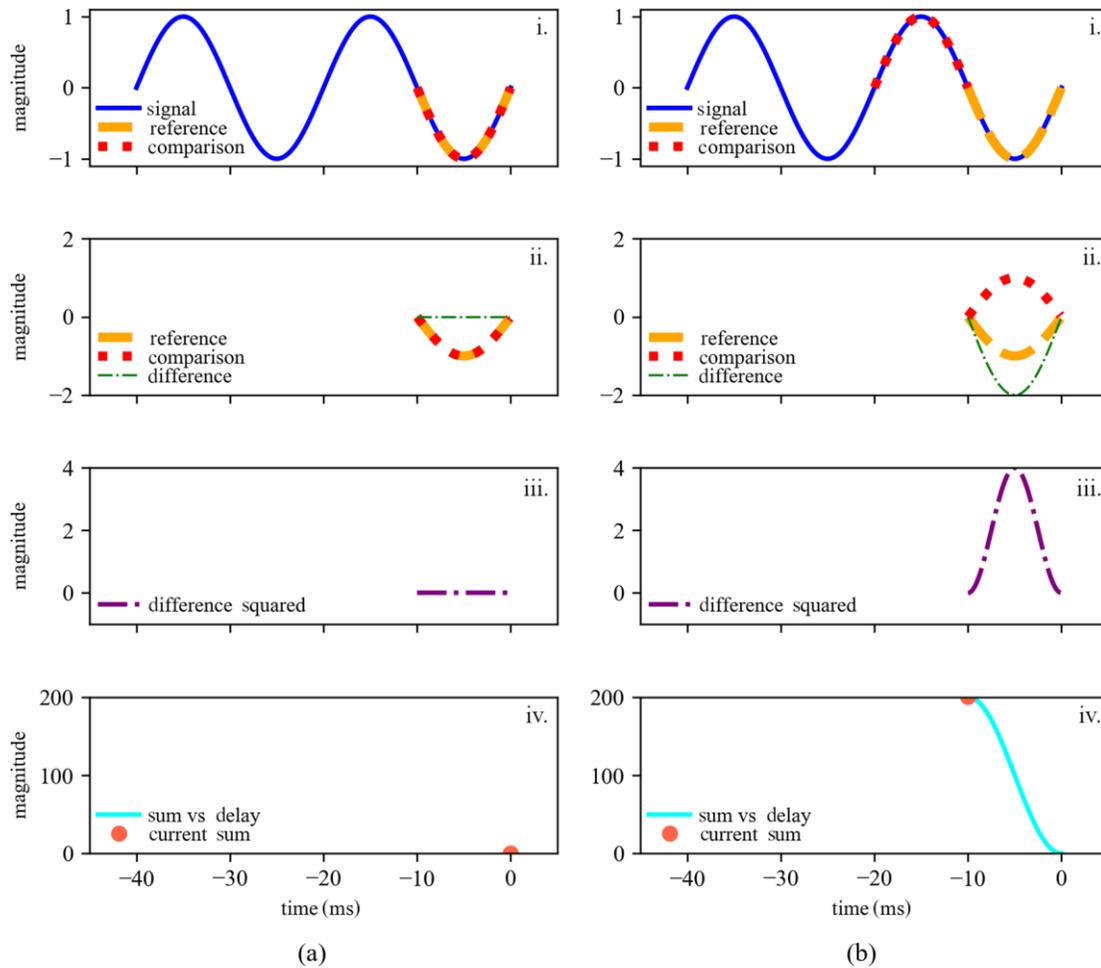


Figure 3.2: Visual representation of delayed comparison error minimization steps, showing the samples taken as input in i., intermediate calculation values in ii. and iii., and the output values in iv. (a) shows these values at delay of 0 ms, and; (b) shows these at a delay of 10 ms along with the stored output values from all previous runs up to the current delay.

and then added together to give a single sum. The bandwidth ranges from a frequency corresponding to a wave period equal to the longest delay between sample pairs on the

low end, up through half of the original data's sampling rate on the high end. The inverse of the data sampling rate determines the frequency precision of the method.

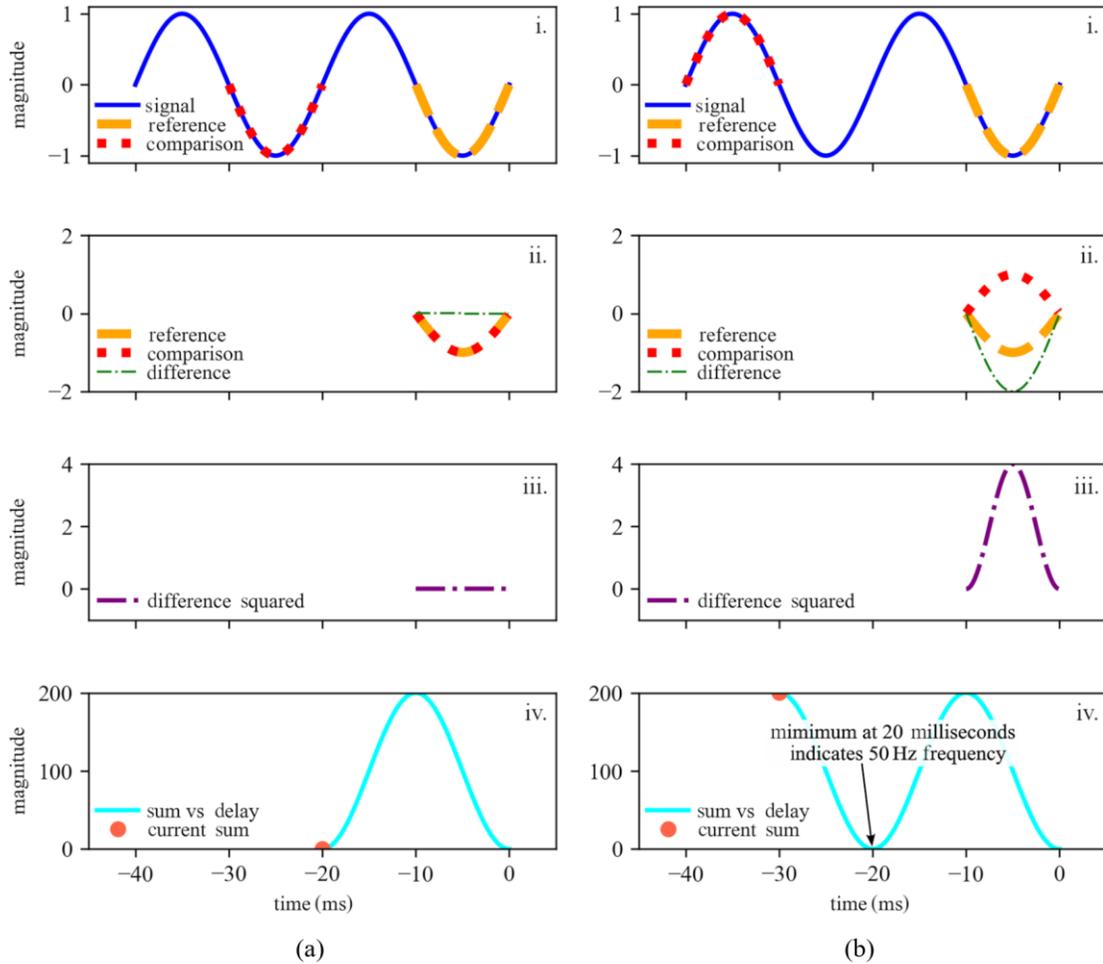


Figure 3.3: Visual representation of delayed comparison error minimization steps, showing the samples taken as input in i., intermediate calculation values in ii. and iii., and the output values in iv. (a) shows these at a delay of 20 ms along with the stored output values from all previous runs up to the current delay, and; (b) shows these at a delay of 30 ms along with the stored output values from all previous runs up to the current delay and identifies minimum in “sum vs delay” curve indicating input frequency.

Figures 3.2 and 3.3 visually represent what is occurring in each step of the Delayed Comparison method. Each step and what is occurring in the figures is further explained below.

1. Figure 3.2 (a) i : A sample set is selected, starting at the data point for Time = 0 (which is on the right in these figures) up through the data point matching the sum of the length of the longest wavelength to be detected plus the length of the “reference” or “comparison” sample sets. These are 300 and 100 sample points, or 30ms and 10ms at a 10,000 sample/second sampling rate, for a sample set “signal” totaling 400 points in length..
2. Figure 3. (a) i . The 100 most recent, and thus right-most, samples are copied from “signal” into “reference”. While this will vary with each subsequent cycle, on the first cycle which figure 3.2 (a) depicts, the same 100 most recent values are copied to “comparison” as well.
3. Figure 3.2 (a) ii. The difference between each point in “reference”, and each respective corresponding point in “comparison”, are found. This process outputs a list of difference values 100 points long.
4. Figure 3.2 (a) iii. Each value in the list is squared.
5. The list is summed to a single value, “current sum” in Figure 3.2 (a) iv.
6. That value is stored into the list “sum vs delay” in Figure 3.2 (a) iv, which will contain the difference squared sum values at their respective delays. The first entry in the list represents difference squared sum at the delay value of 0.
7. Steps (2) through (6) are repeated; on each subsequent cycle the samples placed into “comparison” come from one sample earlier, or to the left, on “signal”. The sum values are each appended to the “sum vs delay” list, with each value representing 1 sample length additional delay. Figure 3.3 (b) iv shows the values going into, and result after the completion of, the final cycle.

“comparison” and “reference” in that cycle are 300 points offset on “signal”, and the “sum vs delay” list is 300 values long upon completion.

Upon completion of the 300th cycle, the local minimum within the region of interest corresponding to the expected wavelength is found from “sum vs delay”. The position of the minimum value within the list indicates the primary wavelength of the signal under analysis; that position value and the time of the last sample used in the comparison cycle are copied to the “signal wavelength” list. That process is repeated from different starting times, each offset by the desired temporal resolution, until the “signal wavelength” list is populated with data covering the length of the signal being analyzed for its primary frequency over time.

3.2.3. MLP Regression - Direct

An MLP, short for multi layer perceptron, is a neural network arrangement characterized by (multiple layers between the input and output layers), and a feed-forward data flow between layers without feedback to earlier layers from the subsequent ones [20]. While often used for classification, they are capable of performing regression and outputting continuous values as well, as is the case implemented here using TensorFlow.

During operation of this method, the sections of the waveform undergoing frequency identification are normalized in amplitude and then fed directly into the MLP network. As such, the network must be able to identify the frequency content of the sample in question when presented at any possible phase offset.

The MLP implementation demonstrated here is pre-trained and does not undergo learning while identifying frequency content. The training and testing data set consists of 22,140 distinct sinusoids, each 100 values long and of a distinct frequency and phase shift. 16,605 were used for training, and the remainder to verify performance. The network is constructed of 2 layers of 64 nodes each, plus the output layer. Training was performed for 1,000 epochs.

3.2.4 Hybrid MLP Regression - Delayed Comparison

This method combines aspects of the Delayed Comparison method with an MLP Regression network, in an attempt to leverage the strengths of each. Similar steps to Delayed Comparison are performed, through where the Sum vs Delay lists, which store the relative mismatch between a sample fixed in time and a comparison sample of progressively increasing time offset, are created. After that step, whereas DCEM identifies the time offset value within the range of interest that results in the minimum error and frequency is calculated from that value, the Hybrid method passes Sum vs Delay list to a pre-trained network. Earlier attempts used identical steps as in the DCEM method in order to calculate the Sum vs Delay lists; testing showed that small tweaks to these steps would improve the performance of the Hybrid method in validation tests with input of varying phase and when run on the example sweep data set shown later.

In the initial implementation of this method, the “sum vs delay” list was generated using the same steps used in the DCEM method, and was then provided to the network for identification of frequency content. “sum vs delay” was calculated using the same ‘reference’ and ‘comparison’ sample lengths as in the previously described Delayed

Comparison method, 100 samples each, and the difference between each point squared before summation.

However, testing revealed that the output of the neural network was being affected by the phase of signal data being analyzed using the hybrid method, and further inspection showed that phase differences affected the values of the “sum vs delay” lists. In an attempt to address the sensitivity to phase, a change in the calculation of “sum vs delay” was implemented; this consisted of changing a step from squaring the point-by-point error to instead finding the absolute value, and increasing the length of ‘Reference’ and ‘Comparison’ samples until manual inspection of data showed minimal effects from varying phase, which was determined to be at a length of 400 samples each.

Figure 3.4 (a) is an example of two “sum vs delay” lists plotted together; the two lists were produced from signals of the same frequency but different phase offset. The plotted lists overlap each other, demonstrating that phase does not have a significant effect for the implementation used in the Hybrid method.

Figure 3.4 (b) shows a similar plot, but produced using the steps employed in the DCEM method, with shorter sample sets used for ‘Reference’ and ‘Comparison’. In contrast to Figure 3.4 (a), the two lists plotted do not overlap fully along their length; the minimums of each list occur at the same location, but the maximums (which are not used to determine frequency in the DCEM method) occur in significantly different locations. This shows the difference in how much of an effect phase has on the calculation of “sum vs delay” when using the two slightly different implementations.

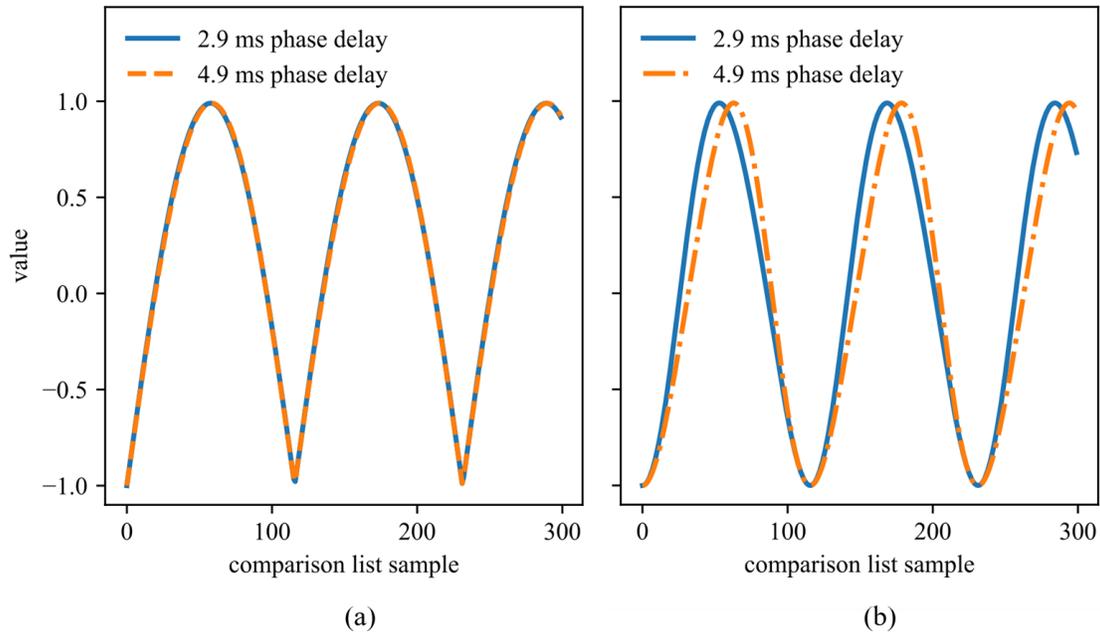


Figure 3.4: Delayed Comparison intermediate steps, calculated using different sample lengths, with inputs of 86.5 Hz, with 2 ms phase difference between inputs of blue and orange traces, showing: (a) calculations as performed for Hybrid method, with longer input samples and resulting reduced apparent phase sensitivity, and; (b) calculations performed as they originally would have been for DCEM method, using a shorter input sample length and resulting noticeable sensitivity to input phase

These changes simplified interpretation of frequency values by the neural network, decreasing the size of the training data set needed to achieve performance as shown in the Results section.

The MLP in this case is also pre-trained, though the data set is somewhat smaller than was used in the previous described Direct method, consisting of 200 samples each 300 values long. The training data set can be much smaller than in the MLP Regression – Direct method, as much of the phase information is hidden by the Delayed Comparison processing step as shown by inspecting the Sum vs Delay lists, so the MLP does not need to be trained to recognize that multiple waves of varying phase are the same frequency.

3.3. Lag quantification

While other factors such as frequency accuracy must be kept in mind when evaluating the performance of an estimator, one important performance metric in applications where reaction time can be critical is how quickly the estimator reports a change in response to a varying input signal. The “lag” of each estimator method was calculated on a portion of the synthetic signal described earlier, covering a length of time where the signal is increasing in frequency to just afterwards.

Delay analysis is performed by comparing each value in the target frequency list with the values in the frequency estimator output lists, and finding the difference between the target frequency times vs when the first estimator value of equal or greater frequency occurred. The pseudocode for the method used is shown in Algorithm 1.

Algorithm 1 Pseudocode for Lag Quantification

```
1: for length of target frequency array do
2:     if (index <= length of estimator frequency array) & (match = false) do
3:         if ( (current target frequency > previous target frequency and
4:             estimator frequency >= current target frequency) or (current target
5:             frequency < previous target frequency and estimator frequency <=
6:             current target frequency) ) and (time difference between estimator
7:             frequency and current target frequency <= max time difference)
8:             write estimator value, target value, and estimator
9:             time to output array
10:            set match = true
11:         else
12:             index++
13:     else
14:         set match == false
15:     proceed to next value in target frequency array
```

CHAPTER 4

RESULTS

The described methods were all run on an identical data set, and the output produced by each was inspected and assessed in order to describe their capabilities and compare their relative performance. The total difference in time between the newest and oldest sample used in calculations by each method, as well as the subjective visual lag of each method behind the original signal, will be noted. Precision, accuracy, and other characteristics will also be discussed.

4.1. Test Input Synthetic Dataset

A data set was generated that contains a continuous sinusoid, amplitude is constant. It starts at a constant 50hz, followed by a slow frequency sweep up to 100hz, holding at the higher frequency, then a slow sweep down back to the first frequency, followed by a similar pattern with a faster pair of sweeps and then frequency steps up and down. The sampling rate of the test input data set is 10,000 samples per second, giving a “length” of 0.1 milliseconds per data point.

The figure below, Figure 4.1, shows both the actual waveform (in blue) that the frequency estimators will be given, as well as the target frequency (in red) that was used to generate the waveform.

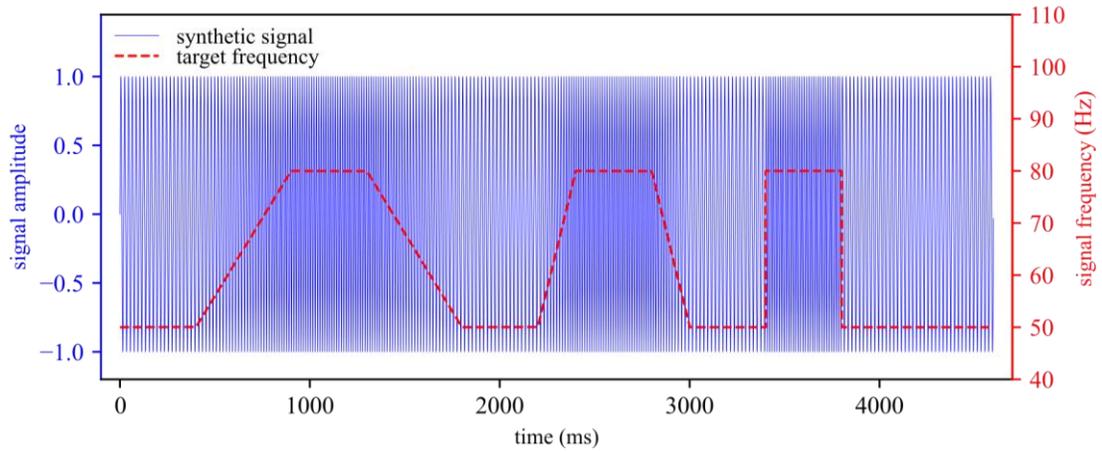


Figure 4.1: Time domain representation of frequency-varying signal, with the target frequency indicated.

4.2. Output of Each Method on Synthetic Dataset

When each frequency identification method is run on this synthetic signal, they give the outputs shown in Figure 4.2:

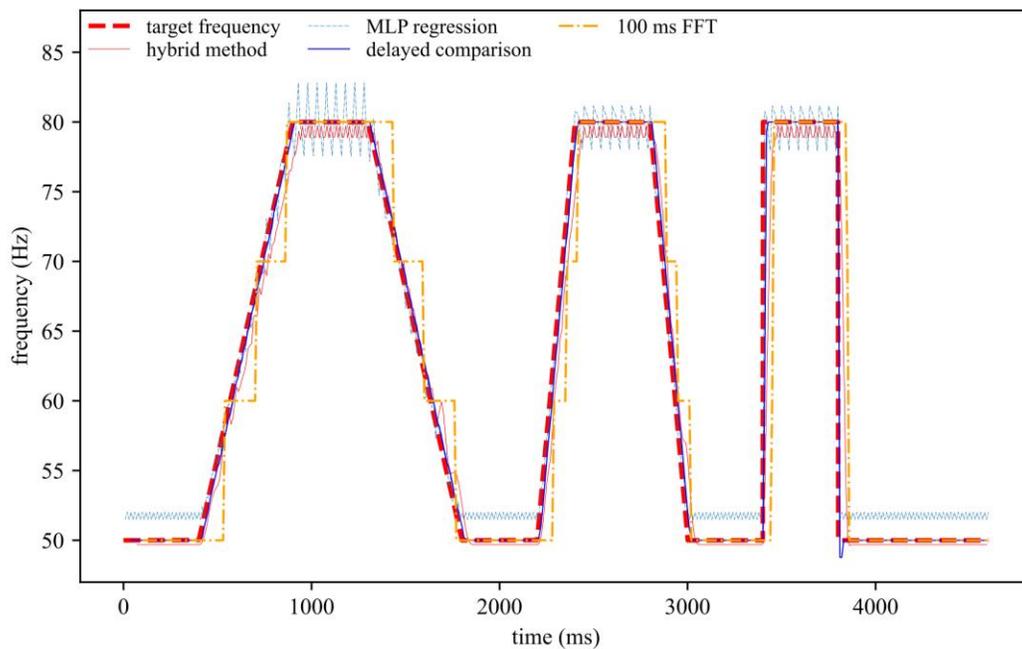


Figure 4.2: Output of frequency estimation methods run on synthetic data set, with target frequency of synthetic data set.

A closer inspection of the results given at the beginning of the first sweep, zoomed in on in Figure 4.3, helps to show the difference between the FFT, direct neural network, and hybrid method output.

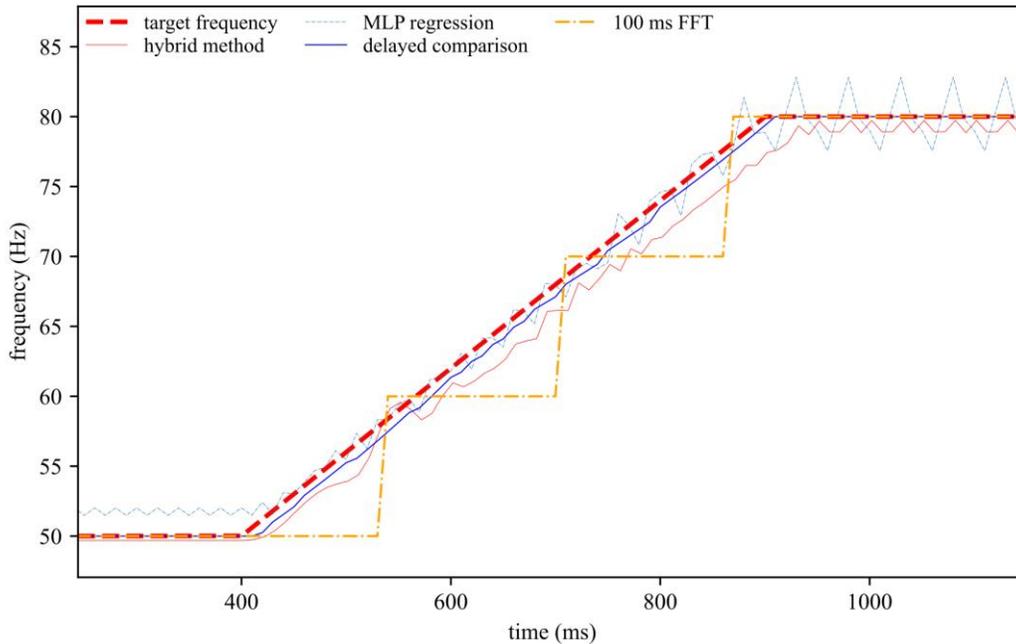


Figure 4.3: Output of frequency estimation methods run on synthetic data set, with target frequency of synthetic data set; close up on first frequency sweep.

4.3. Lag Analysis of Methods on Synthetic Dataset

Delay analysis was performed on a section of the synthetic signal and the associated estimates of each of the methods. The section used in the analysis was the rising frequency sweep shown in Figure , determining lag values for each estimation method for the target frequency curve between 400 ms and 920 ms (frequency estimator outputs from outside of that time span can be used for comparison during lag analysis). These “lag” values were plotted, relative to the target frequency times, for each estimator method, in Figure 4.4. Around the beginning and end of the target frequency sweep, some

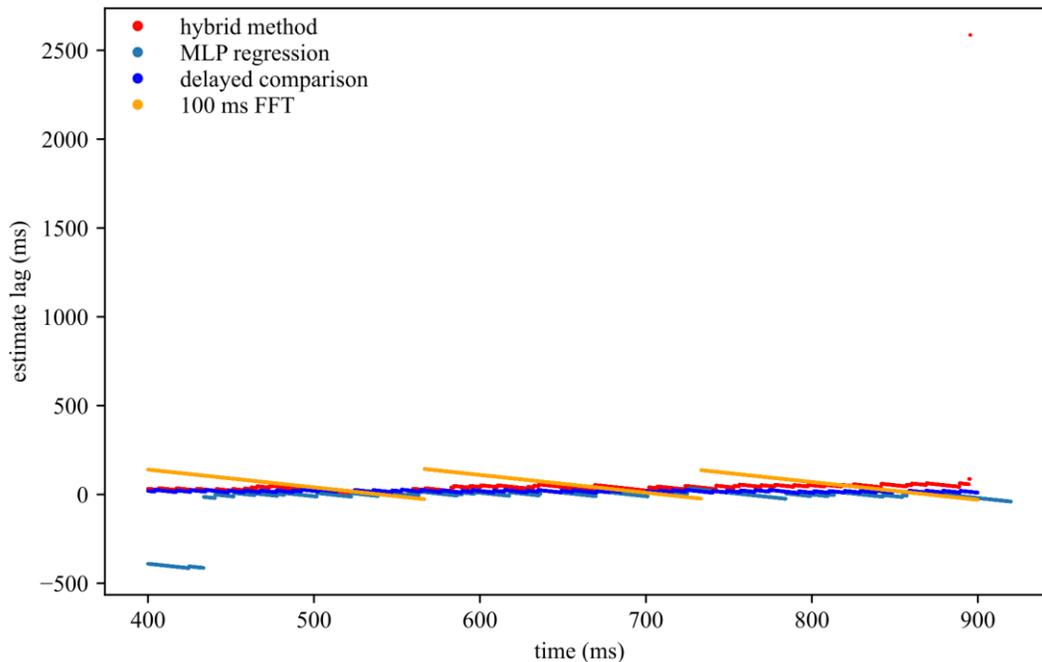


Figure 4.4: Lag of each estimation method on the first sweep of the synthetic data set, with edge effects included to help identify the time range which lag calculation on the frequency estimation methods are meaningful.

of the lag values have a high delay or lead, at times due to the frequency estimators not reaching the target frequency.

Performing the same analysis, but looking at the target frequency only from 450 ms to 850 ms, excludes the edge effects visible at the beginning and end of Figure 4.4. The lag analysis on a single sweep, without edge effects, is shown in Figure 4.5.

Based on the range of lag values that occurred with each method when run across a single sweep, time spans of interest for the potential lag of the frequency estimation methods were determined. These time span of interest ranges were then used when running each method across the full synthetic signal, with its multiple sweeps, steps, and sections of constant frequency. If a lag value result within the time span of interest was not found, no result was generated, eliminating some of the edge effects visible in 4.4

along with other non-relevant results. The results from this time range of interest method on the full sweep are shown in Figure 4.6.

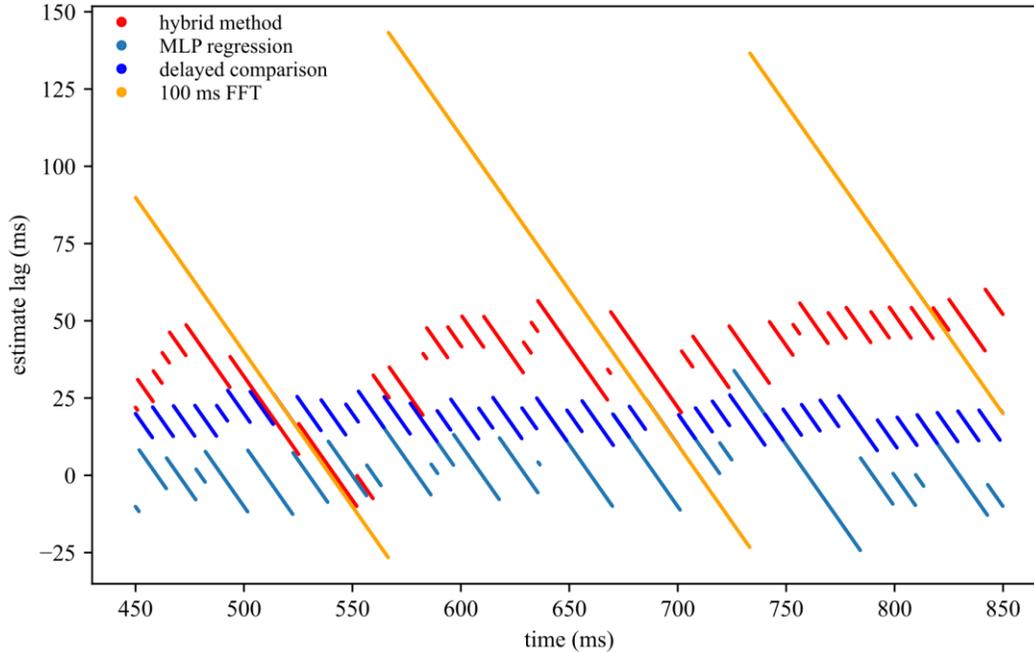


Figure 4.5: Lag of each estimation method on a single sweep, with edge effects trimmed.

Table 4.1: Lag metrics, calculated on a single sweep with edge effects excluded as shown in Figure 4.5.

METHOD	MEAN LAG	MEAN ABSOLUTE LAG	RMS LAG	LAG STANDARD DEVIATION
FFT	57.07	60.21	72.16	44.16
DCEM	17.95	17.95	18.34	3.78
MLP	2.57	7.49	9.83	9.49
HYBRID	36.69	37.08	39.51	14.66

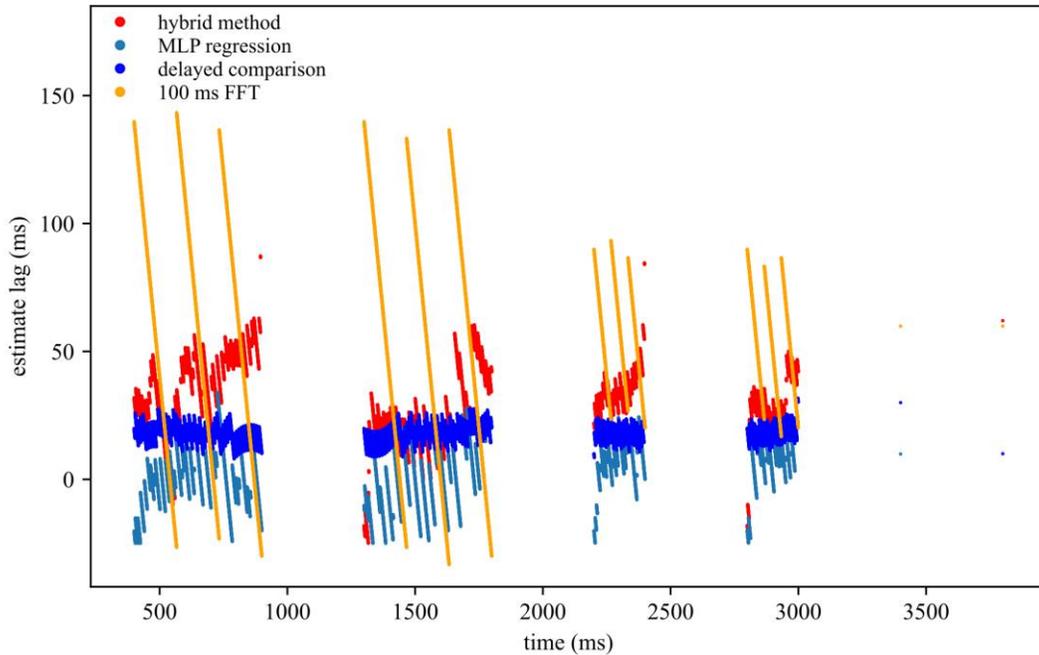


Figure 4.6: Time range of interest lag results for the estimation methods on full length synthetic signal.

Table 4.2: Lag metrics calculated from the time range of interest lag results over the full synthetic data set as shown in Figure 4.6.

METHOD	MEAN LAG	MEAN ABSOLUTE LAG	RMS LAG	LAG STANDARD DEVIATION
FFT	54.98	58.47	69.22	42.05
DCEM	17.96	17.96	18.33	3.67
MLP	2.07	9.47	11.43	11.24
HYBRID	30.87	31.62	34.43	15.24

When viewing the lag results and comparing between methods, it is important to understand that errors in frequency estimation will influence the calculated lag, and as such these results should be considered carefully. Metrics on the lag of each estimation method were performed, and the results are shown below. The metrics in Table 4.1 were

calculated only on lag estimates occurring between 450 ms to 850 ms as shown in 4.5, a single sweep which excludes the edge effects from the beginning and end. The edge effects are visible in 4.4, especially on the MLP regression, hybrid, and 100 ms FFT methods. The statistics in Table 4.2 were calculated from the time range of interest lag results on the complete synthetic data set as shown in Figure 4.6.

4.4. Test Input DROPBEAR Dataset

Acceleration at the end of the DROPBEAR's cantilever beam was measured and recorded over the course of a roller location motion sequence, and that data was used to demonstrate the performance of each of the frequency analysis methods on a physical system. A section of the measured roller location during the sequence, as well as the accelerometer output, are both shown in Figure 4.7 overlaid on top of each other. Within

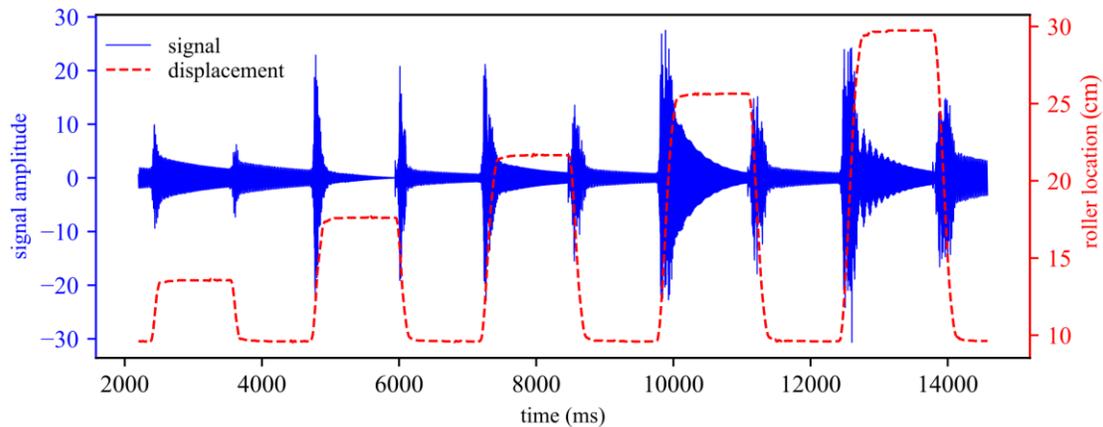


Figure 4.7: Measured roller location and oscillating beam acceleration during roller motion sequence on DROPBEAR.

the range of roller location used in this data set, calculation and previously performed measurements have shown that the frequency the beam oscillates at varies fairly consistently with roller location. Note the beat frequency present in the signal between

approximately 12500 ms and 13500 ms; the presence of which suggests multiple components of similar frequency and amplitude are present. The simultaneous presence of more than one frequency at similar amplitude may present a challenge in determining the signal's primary frequency.

4.5. Output of Each Method on DROPBEAR Dataset

When each frequency detection method is run on the acceleration data collected from the DROPBEAR system, the results are as shown in Figure 4.8. In Figure 4.9, inspecting the outputs closer around the time of a roller location change shows the behavior of each method during roller motion, as well as the relative performance and behavior of each at two different stationary locations.

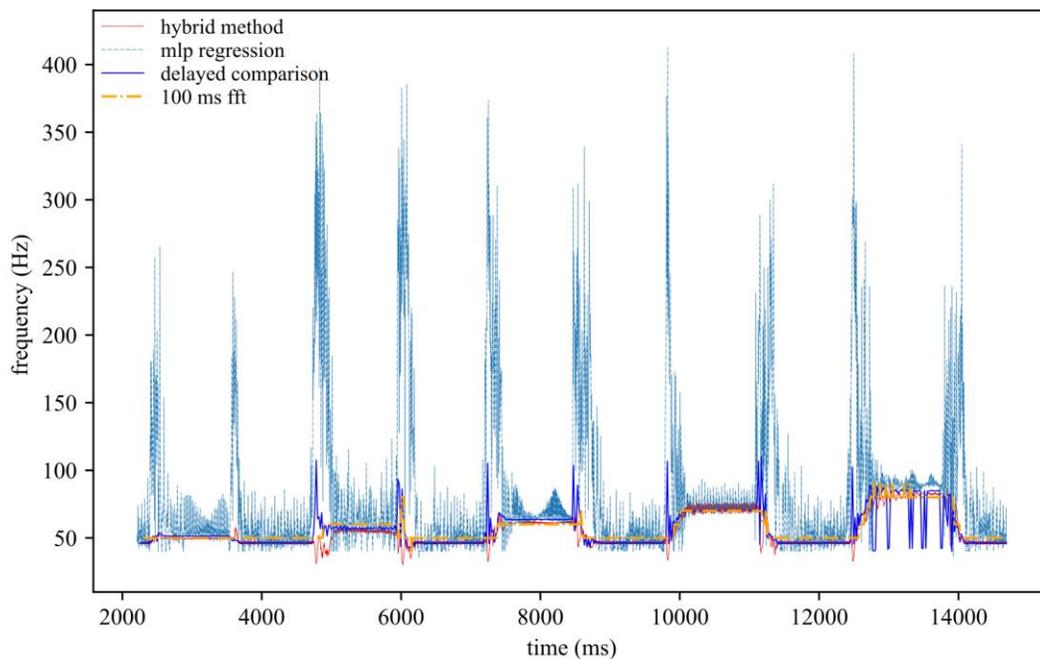


Figure 4.8: Output of the frequency detection methods, when provided with DROPBEAR beam acceleration data.

The Delayed Comparison method is very stable during each period where the roller is stationary, with minor oscillations visible. During motion of the roller however,

there's a large jump; it is unclear whether this is representative of actual frequency content in the data when viewed on such short time scales.

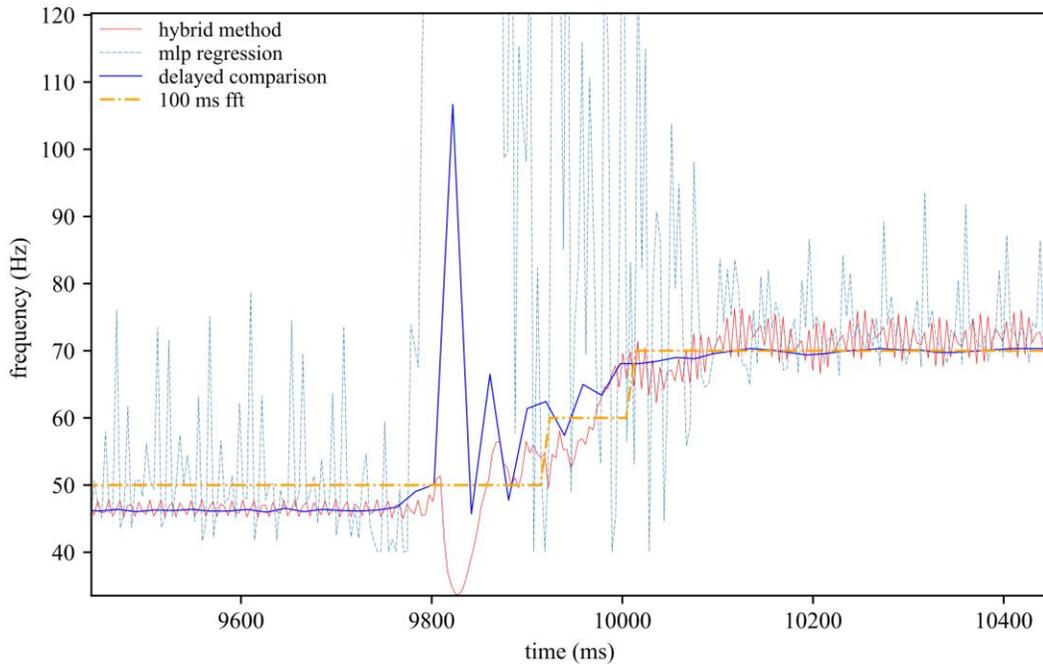


Figure 4.9: Output of the frequency detection methods, when provided with DROPBEAR beam acceleration data; close up during movement of the roller.

The FFT method gives its results at a fairly coarse resolution, with only 3 steps from the earlier to later roller positions across the time motion occurred. Output during each period where the roller is stationary is stable, and the change in output during roller motion, while beginning after each other method shows a change, does reach its end value sooner, and without any oscillation or unexpected values during motion.

The Hybrid method showed fairly similar performance compared to the Delayed Comparison method. Their outputs each start to shift, and likewise stabilize, at similar times. They each exhibited a large jump early in the motion of the roller, though interestingly the jumps are in opposite directions. One distinct difference in their performance is that the Hybrid method shows significantly more “noise” in its output,

with oscillations that are higher in amplitude and frequency than that of the Delayed Comparison method's output.

The output of the Direct Neural Network method, while tracking with roller location well enough to tell that that is what it's doing, exhibited much more high frequency oscillation noise than any of the other methods. This is especially pronounced during roller motion, but even during periods where the roller is stationary the amplitude of output variation is very significant.

CHAPTER 5

DISCUSSION

Visually, it's apparent that the methods vary in delay, frequency precision, and accuracy. Part of the variation in delay between the methods can largely be accounted for by the sample length of each method, and accordingly the age of the oldest data point in use. Sample length and apparent delay don't perfectly correlate because of the differences between how each method makes use of the data, but it is certainly a relevant factor.

It is notable that the performance on clean synthetic data varies from performance on physical acceleration data. Comparing the results on the synthetic data with the output of the methods running on acceleration data from the DROPBEAR shows how the performance of some of the methods are less consistent than others between the two scenarios. For example, the 100ms FFT performs very predictably on the synthetic data but lags behind the response rate of the direct neural network and is relatively imprecise. Comparing the same two methods on the DROPBEAR acceleration data, the FFT's output is unremarkable while the direct neural network shows inconsistent behavior and large jumps in its output especially while the roller is in motion. While additional training data, or modification to the training data set such as the addition of noise or harmonics, may improve the performance of the direct neural network method, the contrast in performance seems to show a potential drawback of the method that must be addressed. The hybrid method, which was trained using similarly generated synthetic data, didn't

suffer the same level of performance loss when run on the DROPBEAR acceleration data, suggesting the method may be inherently more tolerant of attributes of physical data which it was not specifically trained to account for.

One question that was raised when looking at the output of the estimation methods on the DROPBEAR data was whether tracking the frequency, as if it's a smoothly changing or quasi-static value, was a valid approach when the roller starts moving. Aside from the FFT, which was using the longest sample time, none of the methods seem to track early in the motion of the roller following a stationary period. Perhaps the assumption of continuous frequency change with roller position needs to be re-assessed, and initial roller movement introduces signal content that needs to be analyzed in a somewhat different way, compared to continuing roller movement which does appear to result in a frequency sweep as expected from an ideal oscillating beam with varying pin location. If additional noise is being introduced as the roller transitions from stationary to moving, perhaps that can be identified and filtered out; another possibility is that the primary mode hasn't accumulated enough energy to be easily detected early in roller motion events.

CHAPTER 6

FUNDING

This material is based upon work supported by the United States Air Force through the AFRL/RWK contract number FA8651-17-D-0002 and FA8651-17-F-1026. This work is also partly supported by the National Science Foundation Grant numbers. 1850012 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.

Distribution A. Approved for public release; distribution unlimited (AFRL-2025-1758)

REFERENCES

- J. Hong, S. Laflamme, J. Dodson and B. Joyce, "Introduction to state estimation of high-rate system dynamics," *Sensors*, no. 18, p. 217, January 2018.
- J. C. Dodson, R. D. Lowe, J. R. Foley, C. Mougeotte, D. Geissler and J. Cordes, "Dynamics of interfaces with static initial loading," *Dynamic Behavior of Materials*, vol. 1, pp. 37-50, August 2013.
- H. Wadley, K. Dharmasena, M. He, R. McMeeking, A. Evans, T. Bui-Thanh and R. Radovitzky, "An active concept for limiting injuries caused by air blasts," *International Journal of Impact Engineering*, vol. 37, pp. 317-323, March 2010.
- W. Chen and H. Hao, "Numerical study of blast-resistant sandwich panels with rotational friction dampers," *International Journal of Structural Stability and Dynamics*, vol. 13, July 2013.
- C. Stein, R. Roybal, P. Tlomak and W. Wilson, "A review of hypervelocity debris testing at the air force research laboratory," *Space Debris*, vol. 2, no. 4, pp. 331-356, 2000.
- M. T. Heideman, D. H. Johnson and C. S. Burrus, "Gauss and the History of the Fast Fourier Transform," *IEEE ASSP Magazine*, pp. 14-21, October 1984.
- J. W. Cooley, P. . A. W. Lewis and P. D. Welch, "Historical Notes on the Fast Fourier Transform," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1675-1677, 1967.
- J. W. Cooley, "The Re-Discovery of the Fast Fourier Transform Algorithm," *Mikrochimica Acta*, vol. III, pp. 33-45, 1987.
- A. N. Akansu and H. Agirman-Tosun, "Generalized Discrete Fourier Transform With Nonlinear Phase," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 1-10, 2010.
- J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comput.*, vol. 19, pp. 297-301, 1965.

J. M. Climbala, "Fourier Transforms, DFTs, and FFTs," 22 February 2010. [Online]. Available: https://www.me.psu.edu/cimbala/me345web_Fall_2014/Lectures/Fourier_Transforms_DFTs_FFTs.pdf.

M. Cerna and A. F. Harvey, "Application Note 041, The Fundamentals of FFT-Based Signal Analysis and Measurement," July 2000. [Online]. Available: https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf.

J. Hong, S. Laflamme, L. Cao, J. Dodson and B. Joyce, "Variable input observer for nonstationary high-rate dynamic systems," *Neural Computing and Applications*, December 2018.

B. Joyce, J. Dodson, S. Laflamme and J. Hong, "An experimental test bed for developing high-rate structural health monitoring methods," *Shock and Vibration*, vol. 2018, pp. 1-10, June 2018.

A. Downey, J. Hong, J. Dodson, M. Carroll and J. Scheppegrell, "Millisecond model updating for structures experiencing unmodeled high-rate dynamic events," *Mechanical Systems and Signal Processing*, vol. 138, p. 106551, April 2020.

A. Downey, J. Hong, J. Dodson, M. Carroll and J. Scheppegrell, "Dataset-2-DROPBEAR-Acceleration-vs-Roller-Displacement," apr 2020. [Online]. Available: <https://github.com/High-Rate-SHM-Working-Group/Dataset-2-DROPBEAR-Acceleration-vs-Roller-Displacement>.

D. Dorran, "pitch/period tracking using autocorrelation," September 2014. [Online]. Available: <https://dadorran.wordpress.com/2014/09/24/pitchperiod-tracking-using-autocorrelation/>.

R. M. Stern, "Notes on short-time Fourier transforms," 2019. [Online]. Available: https://course.ece.cmu.edu/~ece491/lectures/L27/STFA_Lecture_notes.pdf.

J. A. Scheppegrell, A. G. Moura, J. Dodson and A. Downey, "Optimization of rapid state estimation in structures subjected to high-rate boundary change," in *Proceedings of the ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS 2020)*.

P. J. Werbos, "Applications of Advances in Nonlinear Sensitivity Analysis," in *Lecture Notes in Control and Information Sciences*, New York City, 1981.