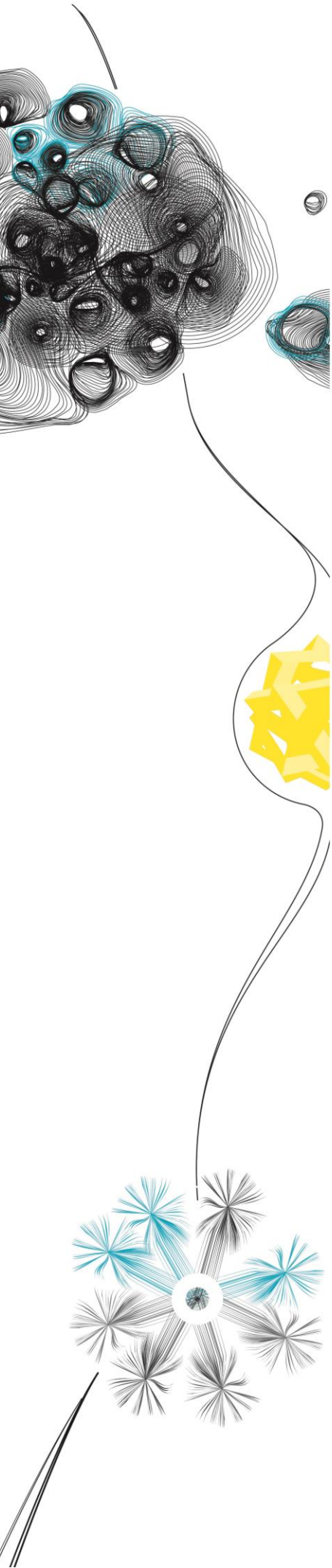

MASTER THESIS

Distributed Processing for Operational Modal Analysis of Bridge Infrastructures Using Wireless Sensor Networks

Alkindi Rizky Dzulqarnain

s1595105 / EMSYS

UNIVERSITY OF TWENTE.

University of Twente

Master Thesis

Distributed Processing for Operational Modal Analysis of Bridge Infrastructures Using Wireless Sensor Networks

Alkindi Rizky Dzulqarnain
s1595105

supervised by :
Prof. Dr. P.J.M. Havinga
Dr. Ir. Nirvana Meratnia
Dr. Ir. Richard Loendersloot

August 11, 2017

Acknowledgements

I would like to express my gratitude to Paul Havinga and Nirvana Meratnia for the patience, guidance, and feedbacks throughout the duration of my thesis work. I would also like to thank Richard Leondersloot for all of the helps whenever I have problems in understanding or interpreting things related to applied mechanics and modal analysis.

Furthermore, my thanks to LPDP Scholarship from Ministry of Finance of Indonesia for giving me the opportunity to study in University of Twente. Also, thanks to my family and friends for their continuous support throughout my study in the Netherlands.

Abstract

The modal parameters of a structure is useful for various reasons, such as finite element model checking or structural damage detection. For large structure such as bridge, where controlled vibration excitation is not possible and only operational vibration data can be used, so-called operational modal analysis (OMA) are used to estimate the modal parameters from structural vibration. In the recent years, there is a trend of using wireless sensor network for continuous OMA of structural health monitoring due to its ease in installation and maintenance. However, understanding the large amount of data produced from structural vibration and energy constrain of wireless sensor node, simply sending raw vibration data to central node means less lifetime for sensor node. Therefore, it is preferable to perform as much as possible OMA processes locally on sensor node such that sensor node only transmit meaningful or transformed data. Based on this, an automatic distributed OMA based on Frequency Domain Decomposition (FDD) is proposed. Unlike centralized FDD, the FDD is only applied to a set of small areas surrounding natural frequencies instead of full frequency range. By performing FFT operation locally on sensor node, each sensor node only send reduced FFT result data, correlated to the selected area, to the central node. Experiment on three real bridge shows that the method could reliably detect correct modal parameters and reduce the transmitted data size into 1.27 percent of its time-domain vibration signal data size. The practical feasibility review also shows that its implementation on wireless sensor node consumes minimum amount of RAM, processing power, and energy consumption. The result shows that lifetime of sensor node is estimated to be able to exceed one year using only 2000 mAh battery.

Keywords : operational modal analysis, frequency domain decomposition, distributed process, wireless sensor network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Due to repetitive usage and aging, every kind of structure suffers from damage and deterioration, including bridges. In order to maintain the safety, bridges are periodically inspected. It is common that this inspection is conducted manually by structural maintenance experts. However, due to the size and complexity of a bridge, this process is time consuming, labor intensive, and costly. As a result, the inspection cannot be done frequently. This also means that there is a possibility that damage could not be detected in time and safety of civilians will be at risk.

Currently, there are increasing efforts toward implementing automatic structural inspection or monitoring systems using various sensors and methods, commonly called Structural Health Monitoring (SHM). Based on several SHM methods review [1–3], SHM could be categorized as local and global. As the name suggest, local SHM damage detection is limited to a small area around the installed sensors. However, it is able to detect small or early damage and can be used to precisely localize the damage. Example of local SHM methods are guided-wave based, impedance based, and acoustic emission based damage detection. Global SHM, on the other hand, could detect structural damage in a large area (even whole structure) using minimum and way less number of sensors than local SHM methods. The disadvantage is, the kind of damage can be detected tend to be medium to severe and it cannot localize the damage as precise as local SHM methods.

Commonly, local SHM is installed to complement global SHM. Local SHM is used to monitor several fracture critical components in structure. Global SHM is installed for continuously measure the healthiness of the structure

as a whole. Understanding the size and complexity of a bridge, global SHM is an attractive option as the first step to continuously monitoring a bridge. Various kind of studies, such as [4–6], have shown that global SHM can be implemented in bridge structure successfully.

Every global SHM methods is based on vibration-based monitoring. The premise is that various structural damages will induce change of vibration characteristic of the structure. This characteristic is reflected in structural modal parameters, which are natural frequency, mode-shape, and damping ratio. By monitoring the modal parameter of a bridge, further analysis may be done to determine whether damages occur and to possibly locate them.

However, modal parameter cannot be measured directly, it must be estimated by processing vibrational data of a structure. For decades, modal parameter estimation has been done to small structures successfully by applying controlled excitation that can be measured. Using this measured excitation and vibrational response of the structure, modal analysis can be extracted. This method is called experimental modal analysis (EMA). However, this method is problematic for large structure such as bridge, which is often difficult to be excited in controlled manner. Moreover, applying controlled excitation means the operation of structure needs to be halted, which is normally undesirable for bridge structures. Thus, for large structure, the so-called operational modal analysis (OMA) is used. Unlike EMA, OMA could extract modal parameter by using only measured vibrational response induced by ambient excitation.

In the recent years, there is a trend of using wireless sensor network for structural vibration monitoring for operational modal analysis. The clear advantage of wireless solutions is easier and cheaper installation compared to cabled system, especially for large structures such as bridges. In addition, the cost and time needed for maintenance are lower. However, the structural vibration monitoring often produces quite large amount of vibration data needed for modal parameter estimation. Commonly, these data are transmitted to a central node in a network that will process them and extract the modal parameters. This will pose a problem for wireless sensor nodes, which are usually battery-powered. Wireless transmission costs more energy than local processing. Transmitting large data means high energy consumption which results in less lifetime of the node. Thus, it is preferable to process the vibration data first on the node or in the network, and transmit only meaningful or transformed data for further operational modal analysis processing.

1.2 Objectives

Based on aforementioned problems, this thesis focuses on designing a distributed process implementation of operational modal analysis (OMA). One of proven centralized OMA algorithm will be chosen and modified such that as much as possible of its process can be distributed into wireless sensor node. The whole processes should be automatically done without human intervention. However, instead of obtaining all of the modal parameter, this thesis only focus on obtaining mode shape and natural frequency. In damage detection on structural health monitoring, damping ratio has not been proven to be damage sensitive, and obtaining accurate damping ratio is quite difficult. Only mode shape and natural frequency (and their derivatives) that have been shown to be sensitive to structural damages.

Moreover, there are several things that need to be concerned in designing the distributed OMA. It should produce less data transmission than its centralized counterpart. Also, the accuracy should be as close as possible to its centralized counterpart. In addition, the distributed process should fit within the resource constrains of common wireless sensor node hardware, which mainly are memory usage, processing power, and energy consumption.

Chapter 2

Distributed Operational Modal Analysis

2.1 Operational Modal Analysis Algorithms

There are severals algorithm proposed for Operational Modal Analysis (OMA) by various researchers. However, there are two OMA algorithms that are frequently used for various structural modal analysis due to its accuracy. The first algorithm is Stochastic Subspace Identification (SSI) [7]. There are two variants of SSI algorithm, which are SSI-Data and SSI-Covariance. The SSI-Data works directly using time-domain structural vibration data, while the latter needs the time-domain vibration data to be converted into matrix that comprises of time-lag covariance between data from each sensors. While these algorithms are known as their accurate result, they are not quite suitable for wireless sensor network. The algorithms processes need close interdependence of time-domain signal between each sensors. This means that each wireless sensor node cannot perform any distributed processing, they simply transmit the data into a sink (or central) node where time-domain signal of each sensor will be processed further. The only attempt to distribute SSI algorithm is done by Cho et al. [8] by clustering scenario. The wireless sensor network is divided into several cluster, which contain one cluster heads and several cluster members. Instead of sending the time-domain signal to central node, each cluster member send the time-domain signal to its cluster head. Cluster head will perform SSI algorithm and produce partial modal parameter, which are a set of natural frequencies and partial mode shape. Each cluster head send these data to central node, where they will be combined together into global natural frequencies and mode shape. However, while the

transmission range from cluster member is potentially reduced, cluster member still need to send huge amount of time-domain signal to cluster head. The energy consumption of the sensor node from data transmission is still quite high. Furthermore, Sim et al. [9] shown that combining partial mode shape will introduce inaccuracy in global mode shape estimation.

Instead of based on the time-domain vibration signal, the second algorithm is based on frequency spectrum of the structural vibration. The algorithm is called Frequency Domain Decomposition by Brincker [10]. It is an extension from a simple OMA algorithm called Peak Picking (PP) [11], which basically works by selecting peaks, which signifies natural frequency location, in power spectral density (PSD) graphs of vibration data. FDD extend this further and solves the two main PP problems, inaccuracy in presence of closely spaced mode and subjective selection of peaks when multiple sensors are used. Similar to PP, FDD works by processing the PSD of vibration data. FDD extends the process by combining PSD data from various sensor using singular value decomposition (SVD). The SVD process resulted in a singular values plot that reveals which peaks are consistent throughout multiple sensor measurements, thus reducing the error on picking the peaks and estimating the natural frequency. The mode shape is extracted from corresponding singular vector of selected peak locations. This algorithm is attractive for wireless sensor network implementation of OMA. Instead of transmitting raw time-domain vibration signal, wireless sensor node can process the signal into frequency domain. Given the huge amount of required time-domain vibration signal in OMA and limited frequency range of interest, transforming the signal into frequency domain will potentially reduce the amount of data needs the be transmitted.

Regarding the accuracy, several studies has been done to compare the FDD with SSI results. Yi and Yun [12] performs three real structure OMA using FDD and SSI-COV (named SSI-CVA on the paper). Out of three experiment, only the first experiment on truss frame structure shows that one out of eight mode shape is estimated poorly compared to numerical model (with MAC[1]around 0.5), while SSI estimates all mode shape accurately. Peeters and Roeck [13] performs OMA on mast structure FE Model using various algorithm, including FDD (named CMIF in the paper) and SSI-Data. The result shows the natural frequency variation difference is less than 1 percent and MAC value differences of all of the mode shape are around 0.05 at maximum.

[1]MAC [14] is a similarity measurement value between two mode shapes. MAC value of 1 means two mode shape are basically same, while 0 means they are not correlated at all.

Cunha et al. [15] performs OMA on Vasco da Gama Bridge, a large cable-stayed bridge, to compare the result from FDD with SSI. The comparison result shows that the natural frequency difference is insignificant and nine out of twelve mode shapes estimated from FDD are similar to SSI result with MAC values higher than 0.9. Two out of twelve mode shapes has MAC around 0.8 and only one mode shape has low MAC value (around 0.45). Weng et.al [16] also compared SSI and FDD performance on real bridge, large cable-stayed bridge Gi-Lu in Taiwan. The result concluded that the modal parameter obtained using FDD is in agreement with SSI results. All of the result of previous studies shows that even though SSI tend to gives better accuracy in estimating modal parameter, the difference is minuscule. Understanding the potential of data transmission reduction on FDD by perform distributed process, it is worth to investigate further how to realize distributed FDD in wireless sensor network.

2.2 Related Works

There are several attempts already to implementing distributed FDD on Wireless sensor network. Zimmerman et al. apply a distributed FDD algorithm to pedestrian bridge [17] and theater balcony [18]. The wireless sensor network is divided into two node group chains. Each sensor node process the time-domain signal into frequency spectrum using Fast Fourier Transform (FFT). Then, each sensor node will send the frequency spectrum data into its neighboring node that is closer to the central node. Receiving node , based on received and its own frequency spectrum, will perform FDD algorithm to obtain partial modal parameter. This partial modal parameter will sent to the central node to be combined. As mentioned before, this clustering approach has shown by Sim et al. [9] to induce inaccuracy. The lesser amount of its cluster member, the higher the inaccuracy.

Another work on distributed FDD is done by Hackmann et al. [19]. Their work reduces the data transmission reduction further. Hackmann et al. notice that not all of the data in frequency spectrum is needed for FDD. Frequency spectrum at the exact natural frequency is the only data that is needed to extract the modal parameter using FDD. Based on these, each wireless sensor node will perform FFT and construct power spectral density (PSD). Then each of them will perform peak picking locally from the PSD, as a mean to estimate natural frequency. Frequency domain data from selected peaks are transmitted to the central node for further FDD process. Using this approach, the data transmission is reduced significantly. However, there are several drawbacks from this approach. What have been implemented by

Hackmann et al. is not different from Peak Picking algorithm. There are possibility that each sensor node detect different peaks, or that might be one of them is not natural frequency peak. Even if they detect the same peak, the nominal frequency value might be different, e.g. sensor A detect a natural frequency on 2.45 Hz while sensor B detect it at 2.5 Hz. This is possible due to calculation error or measurement variation (which will be shown in latter part of this report). These problems are not addressed in their paper. Despide the drawbacks, Hackmann et al. observation shown that FDD has high potential to be implemented in wireless sensor network. Thus, this thesis propose a distributed OMA solution that improves the previous attempts on distributed FDD.

2.3 Distributed FDD

The complete centralized Frequency Domain Decomposition process can be summarized as figure 2.1.

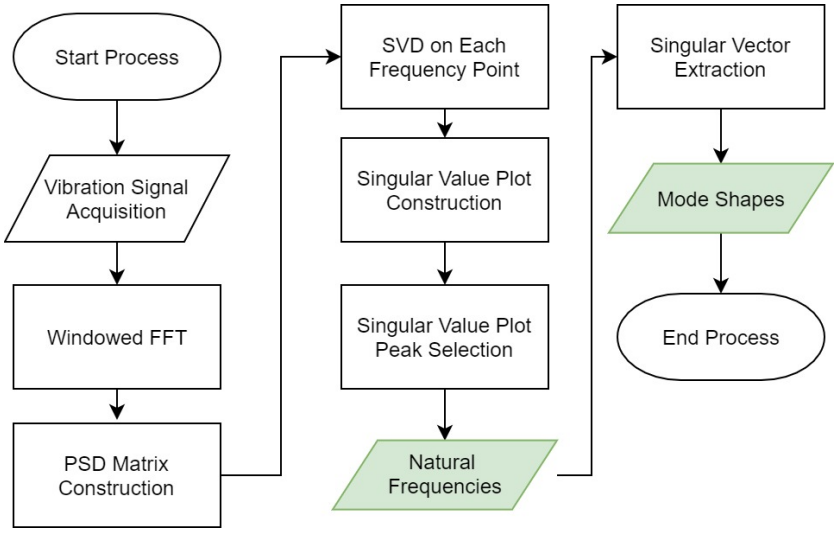

Figure 2.1: Overview of FDD Process

The vibration signals are converted into frequency domain data using FFT. Based on this FFT results, a PSD matrix that consists of auto and cross power spectral density (PSD) between each sensors is made. If there are M sensors and N-point FFT is used, then the matrix size in first two dimension is M x M, with third dimension is filled with 0.5N frequency points from 0 Hz to half of sampling frequency. Later half of FFT result is only mirror of the first half. Therefore, the last half part is unnecessary. Then, singular

value decomposition (SVD) is applied to this PSD matrix for every frequency points, which means there will be 0.5N SVD operations performed. The result of this SVD are singular values and vectors for each frequency point. In order to obtain the natural frequency, the singular value is arranged into a plot. The plot example from a pit bridge measurement in University of Twente is shown as an example in figure 2.2. Since there are 6 sensors used in this example, there are 6 singular values and vectors, represented as 6 different color lines in plot. The one that contains the information of natural frequency is first singular value, which has the highest plot value and colored as blue in figure 2.2. Based on this plot, peaks from first singular value are selected. That peak locations in frequency point is the value of natural frequencies, which in this case are 51.29 Hz, 88.18 Hz, 148 Hz, and 166.1 Hz. Then, mode shape can be extracted from first singular vector of each corresponding peak location.

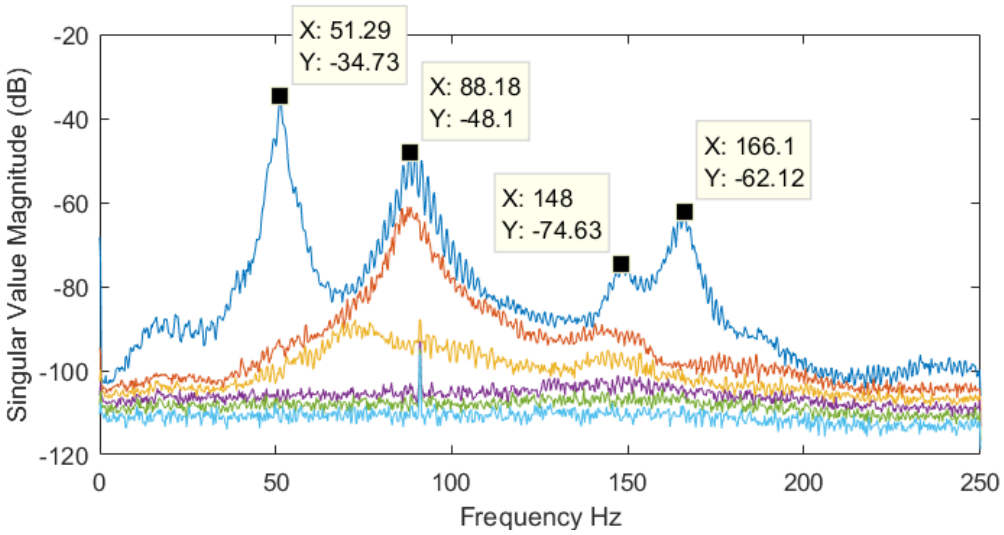

Figure 2.2: Example of Singular Value Plot

Based on Hackmann et al. [19] proposed solution and as can be seen by closely inspecting the whole process of FDD, the SVD process does not need to be applied to whole frequency points. If natural frequencies information can obtained beforehand, then only PSD matrices located in natural frequencies are needed to obtain the mode shapes. In previous case, only PSD matrix for frequency points of 51.29 Hz, 88.18 Hz, 148 Hz, and 166.1 Hz. The remaining data are unused in the end. Hackman et al. propose peak picking routine to before PSD matrix construction process, so that each node can determine the natural frequency locally by detecting peaks on its own

PSD plot. Then, instead of sending the whole FFT result, each sensor node only send FFT data corresponding to frequency location of obtained peaks. Using this method, it can be seen that the data transmission can be greatly reduced. However, each sensor could have different shape of PSD and natural frequency location. Figure 2.3 illustrate this.

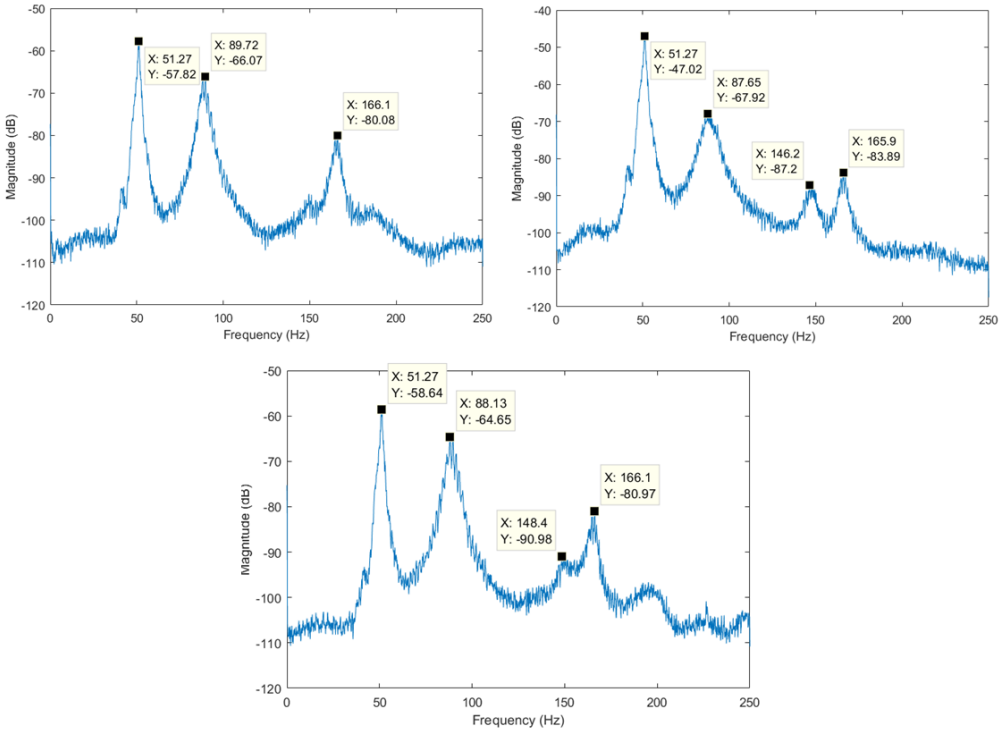

Figure 2.3: PSD plot from three sensor used for pit bridge measurement

As can be seen in figure 2.3, first, second, and fourth natural frequency peak are clearly apparent in each of sensor PSD plot. On the other hand, the third natural frequency peak cannot be seen in first sensor and barely noticeable in third sensor. This could be caused by the placement of first sensor is quite close to the node point of third mode shape. Moreover, some natural frequency peaks detected in one sensor is located slightly different compared to the same peak detected in other sensor. Thus, sending FFT result based on only local peak information tend to give erroneous result. However, based on observation, the variation of one natural frequency detection between each sensor is minimal and they are close to true natural frequency location. Therefore, if FDD process is applied only to a small but sufficient area of frequency points around the true natural frequency locations instead of whole

frequency points, it is still possible to obtain correct natural frequencies. This reduce the size of required FFT result data. If FFT operation is applied locally in each sensor node, this will effectively reduce the data transmission to central node. Based on this observation, a three phase distributed FDD is proposed. The details of each phase will be explained in following subsections. In order to explain the method easily, the same previous example, pit bridge measurement result, is used throughout the remaining explanations.

2.3.1 First Phase of Distributed FDD

The first phase objective is to find set of frequency points area to be used for finding true natural frequency locations in the second phase. For convenience purposes, this set of area is called **peak areas**. The complete process of first phase is illustrated in diagram on figure 2.4.

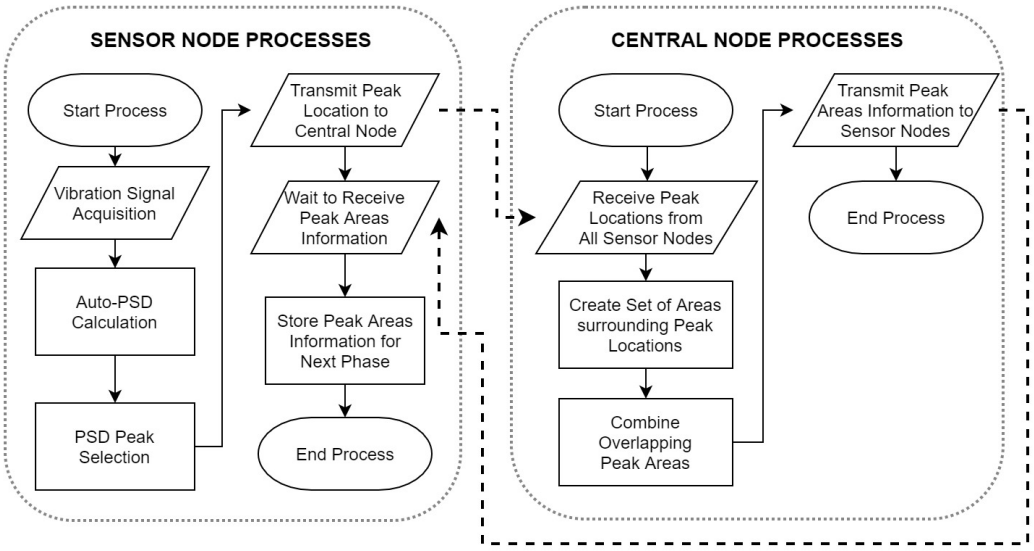

Figure 2.4: First Phase Process of Distributed FDD

Each sensor node will perform sufficient number of windowed FFT calculation to estimate auto power spectral density (Auto-PSD) using Welch's method [20]. Higher number of window tends to produce better auto-PSD with lower noise in cost of processing time. Then, peak selection is applied to obtained Auto-PSD locally in sensor node. The obtained peaks represent candidates of true global natural frequencies. They are used for determining peak areas that will be needed for FDD process in second phase. The peak selection process is illustrated in figure 2.5.

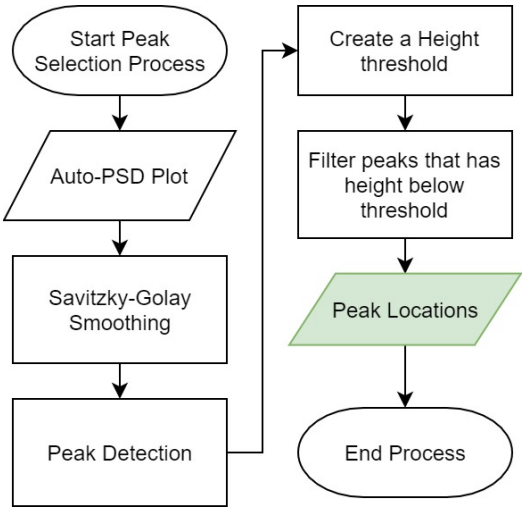

Figure 2.5: First Phase Peak Selection Process

The PSD plot is smoothed using Savitzky-Golay [21] smoothing filter first to reduce noises and jagged part of the plots. Savitzky-Golay is used as it can produce good smoothing while maintain peak shapes compared to simple moving average smoothing. The peak detection algorithm, based on the work of Yoder [22], is applied to the smoothed plot to obtain peak candidates. The algorithm is sufficiently reliable to detect peaks in noisy signal or plot. However, it is not perfect, there is still possibility that some noises or false positive are detected. Thus, additional process is added. Since, natural frequency peaks tend to be dominant height-wise, all of peak candidates are filtered based on their height. If the height is sufficiently high, it is passed. The height threshold is based on a percentage of the highest peak height. This thresholding result in set of peak locations that acts as potential natural frequency locations.

After obtaining local peak locations, each sensor node send them to central node and wait for further reply from sensor node. Upon receiving those data from all of the sensor node, central node will create set of peak areas surrounding each peak locations. The peak areas width should be set at sufficient value. Area width that is too large will force each sensor node send too many unnecessary data in next phase. However, peak area width that is too narrow will falsely treat two peaks that actually points to same natural frequency, but located slightly differently, as two different peaks. This, in effect, produce two peak areas that result in wrong natural frequency determination. After constructing these areas, every overlapping peak areas are combined such that in the end a minimum number of peak areas area

obtained. The process is best explained using example plot in figure 2.6.

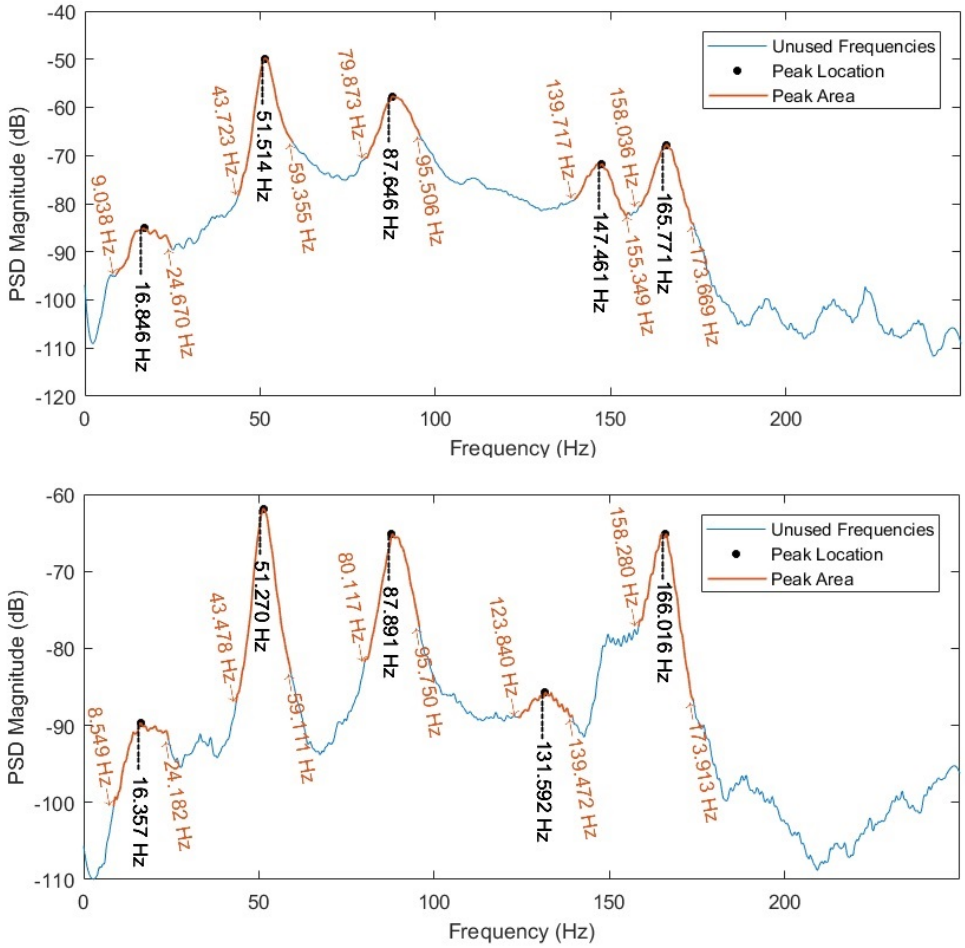

Figure 2.6: First Phase Peak Area Construction Example from Two Sensors

Both sensor detect five peak locations from their PSD plot. However, each of their peaks are in slightly different location. A peak areas with a width of 15.633 Hz is created in each peak locations, that will result in 10 peak areas, 5 from each sensor result. As can be noted, first peak area of first sensor (9.038 Hz - 24.670 Hz) is overlapping with first peak area of second sensor (8.549 Hz - 24.182 Hz). Central node will combine them into single peak area that is located fro 8.549 Hz to 24.670 Hz. This also apply to the remaining peak areas. In the end of combination, only 5 peak areas are resulted. This combined peak areas information are sent back to all sensor node for second phase.

2.3.2 Second Phase of Distributed FDD

The second phase objective is finding the true global natural frequencies using the FFT result data located in peak areas from first phase. The complete process of second phase is illustrated in figure 2.7.

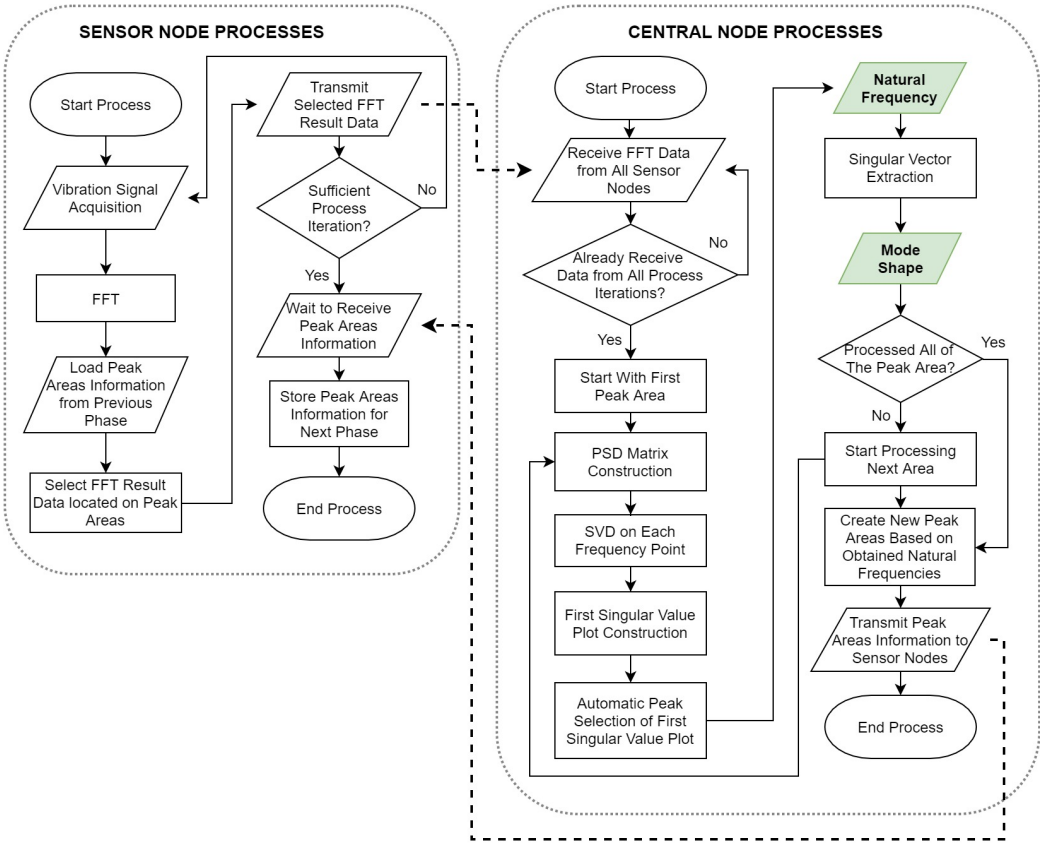

Figure 2.7: Second Phase Process of Distributed FDD

Sensor node start with acquiring vibration signal and convert them into frequency domain using windowed FFT. This FFT result is needed to be sent to central node for constructing PSD matrix in FDD process. However, not all of the FFT result data are selected. Only FFT result located in frequency points specified in peak areas from first phase are selected. For example, if there are only two peak area obtained located on 7 Hz - 23 Hz and 30 Hz - 46 Hz, then FFT result outside those frequency ranges are excluded. After FFT result data are selected, they are sent to the central node. This process is repeated several time according to the desired length of vibration signal to be processed. For example, if 2048 point windowed FFT without

overlapping is used for signal with sampling rate of 500 Hz, then each signal window has time duration of 4.096 seconds. If user desire the minimum length of vibration signal is 204.8 seconds, then there will be 50 windows of signal needs to be processed. Thus, 50 times of process iteration needs to be done. Longer vibration signal acquisition will results in lower noise in resulted singular value plot from FDD process, which means lower chance of false positive in natural frequency detection. After sufficient iterations, sensor node will stop the process and idle wait further reply from central node. As can be observed, since commonly there are only so many peaks detected from previous phase and the peak area size limited, the data transmission is greatly reduced compared to sending the complete FFT result data.

Upon receiving FFT data of whole iterations from all of the sensor node, central node will start the remaining FDD process to obtain modal parameters. All of the FFT result data are transformed into PSD matrix, one matrix for each peak area. The same FDD process as centralized one are applied. SVD is performed for each frequency points in the peak area, then singular value plot is constructed. The difference is in the peak selection. On normal centralized FDD, user need to select the peak manually. In this proposed solution, everything is done automatically, including the peak selection. The same peak selection process used in sensor node on first phase, illustrated in figure 2.6, is applied. The only difference is, instead of using PSD plot, it uses first singular value plot. After obtaining the peak locations, which correspond to natural frequencies, mode shapes are extracted from corresponding first singular vector. This process are done repeatedly until all of the peak areas are processed. The process and result can be further explained by example in figure 2.8.

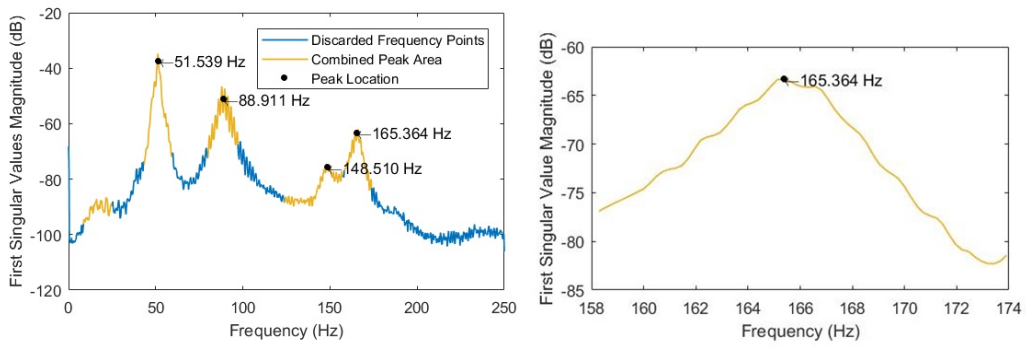

Figure 2.8: Second Phase Result Example. Selected Singular Values (left) and Smoothed Singular Values of One of the Peak Area (right)

Based on first phase result, there are five peak areas obtained, which their location in frequency points is shown as yellow line in figure 2.8. The blue line shows the frequency points where the FFT result data are discarded or not transmitted by sensor node. Since there are five peak areas, the modal parameter estimation process of central node is repeated five times, each iteration process different peak area. The example of smoothed singular value plot used in automatic peak selection is shown in right-side plot of figure 2.8. It can be noted that the first peak area is false positive. It is caused by at least one of the sensor detects local peak in auto-PSD plot around the area that does not correspond to the global natural frequency. However, by combining data from multiple sensor using SVD, the second phase process reliably determine that there is not any natural frequency peak located in the first peak area.

As mentioned previously, natural frequencies could be shifted due to various factors. As a consequence, peak areas are still needed to capture the correct modal parameters for next phase, as it is not possible to predict the natural frequency shift. Therefore, central node will make a new set of peak areas around the obtained natural frequencies, but using minimum area width. Since the frequency shift is commonly minimal, a narrow peak area such as 3 Hz is sufficient. Using example result in figure 2.8, there will be four peak areas, which are located in 50.039 - 53.039 Hz, 87.411 - 90.411 Hz, 147.010 - 150.010 Hz, and 163.864 - 166.864 Hz. This new peak area information are sent back to all sensor node for next phase.

2.3.3 Third Phase of Distributed FDD

The third phase is the steady state phase. After running the first and second phase at least once, the wireless sensor network will stay in this phase for long-term continuous modal parameter estimation. The complete process of third phase is basically the same as second phase process in figure 2.7, except in two things. First, since the peak areas area narrower than second phase, the data transmission is reduced even further. Second, since it is already certain that each peak area only correspond to one natural frequency peak, the peak selection process becomes simpler. The peak selection process is illustrated in figure 2.9.

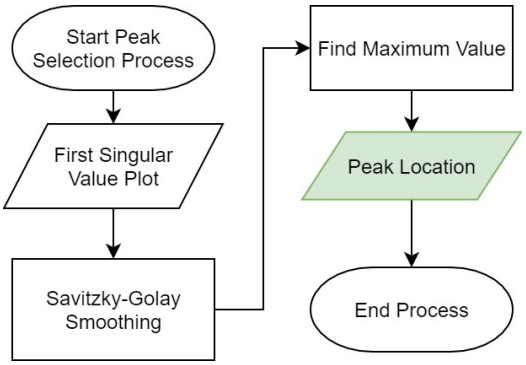

Figure 2.9: Third Phase Peak Selection Process

It starts with the same as subprocess as previous peak selection process, applying Savitzky-Golay smoothing to reduce noise peaks in first singular value plot. Then, peak location or natural frequency is obtained by simply finding the maximum value of smoothed singular value plot.

Upon processing all of the peak areas and obtaining all modal parameters, central node re-adapt the peak area to newly obtained natural frequencies. It is done such that in case there is a continuous natural frequency shifting, modal parameters can always be correctly estimated. Then, this adapted peak areas information are sent to all sensor node for third phase session.

Chapter 3

Experiment Result

In order to test the proposed distributed frequency domain decomposition, three operational modal analysis experiments are done in real bridge structure located on the campus of University of Twente. The three experiment follows similar scenario. Six wireless inertial sensors, ProMove-Mini, produced by Inertia Technology B.V. are used to measure the vibration of the bridge as acceleration under ambient excitation for a specific duration.

All of the bridge is measured using 500 Hz sampling rate to provide sufficiently broad frequency detection capability. The vibration measurement is done twice, so each bridge produce two set of vibrations data. The first vibration data is used for the first and second phase of distributed FDD and the other set of data is used for second phase. Each experiment uses 2048 point windowed FFT for their processes. Each window is filled fully with vibration signal, without zero-padding. Therefore, the FFT produces effective frequency bin resolution of 0.244 Hz. The windowing function used is Hanning and PSD estimation is done without overlap between each window.

Four windows are used to estimate auto-PSD in first phase of each experiment setup. Regarding PSD and singular plot smoothing, Savitzky-Golay filter with order of 3 and length of 13 data points is used for all experiments to make sure a good trade-off between smoothing and peak shape preservation. In order to separate the natural frequency peaks from insignificant false positive peaks, peak height threshold for all peak selection process in each of the experiments is set to 30 percents of highest peak height. Peak area width of second phase and third phase for all experiment are set to 15.6224 Hz (65 frequency bins) and 2.9311 Hz (13 frequency bins) respectively.

3.1 Pit Bridge

The first experiment is performed in a small pit bridge located in University of Twente. The construction of the bridge can be seen in figure 3.1. The bridge has width of 169.2 cm and length of 106 cm. The structure is made of steel with steel mesh deck.

Figure 3.1: The structure of the pit bridge of University of Twente

The six sensor are placed on top of deck with arrangement that follows figure 3.2.

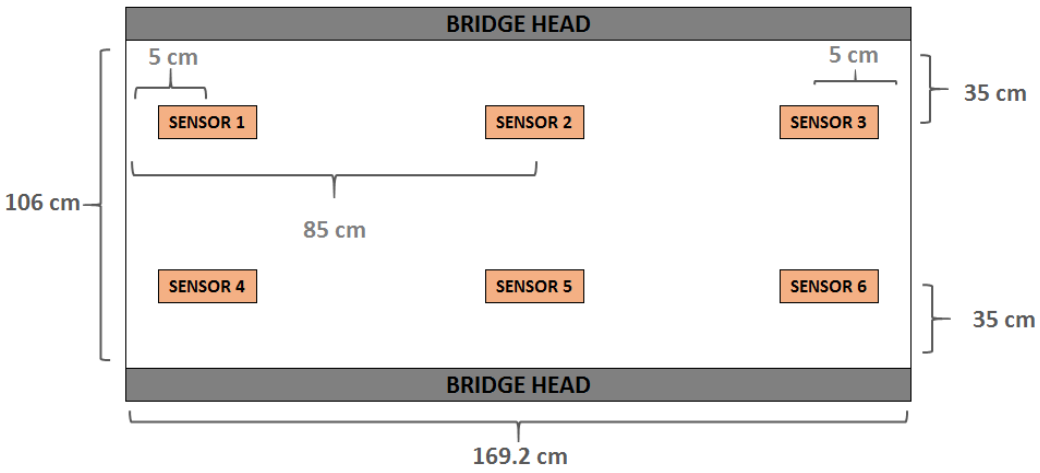

Figure 3.2: Sensor placement on the pit bridge of University of Twente

The vibration excitation sources are pedestrian and bicycle that crosses repeatedly on the bridge. Vibration is captured in one axis, on the normal direction of the bridge (vertical direction). First and second measurement are performed for approximately 237.57 seconds using 500 Hz sampling rate,

which produces 118784 samples each. The example of acquired vibration signal is shown in figure 3.3.

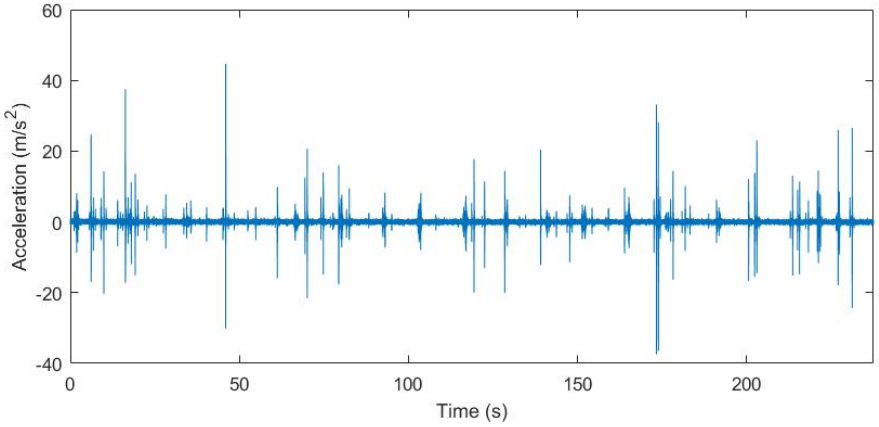

Figure 3.3: Vibration Signal Example of Pit Bridge Measurement

The vibration signal is processed with 58 windows of 2048 point FFT to obtain PSD matrix for second and third phase. Processing the obtained data using first phase and second phase distributed produce result as shown in figure 3.4.

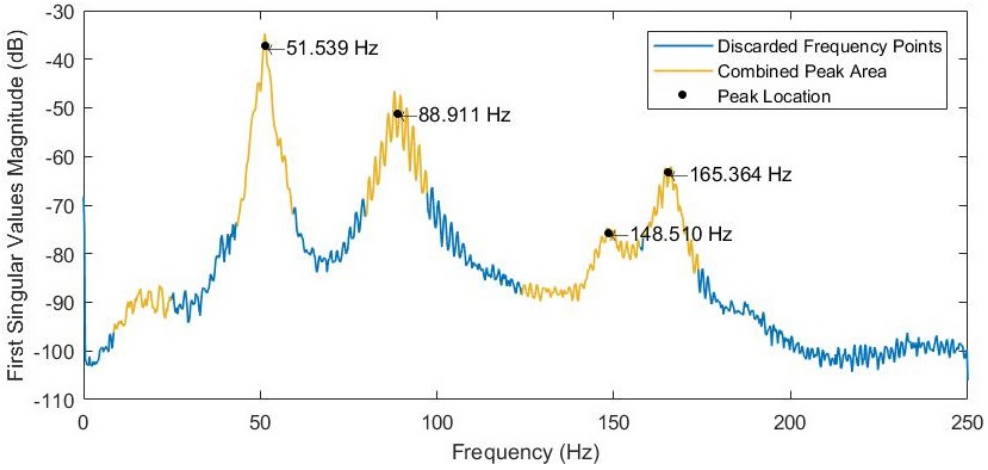

Figure 3.4: Second Phase Result Plot of Pit Bridge Experiment

Figure 3.4 shows that first phase produce 5 peak areas but four natural frequencies, as the first peak is false positive. However second phase is able

to determine that there is not any natural frequency in that peak area. As can be seen from the plot as well, not all frequency points are necessary to perform obtain the natural frequency. Therefore, there are already some data transmission reduction from second phase. Using natural frequency results obtained from second phase, a third phase distributed FDD is performed. The resulted modal parameters and its comparison to second phase result is summarized in table 3.1.

Table 3.1: Third Phase Result of Pit Bridge Experiment

Third Phase Natural Frequency (Hz)	**Second Phase Natural Frequency (Hz)**	**Natural Frequency Difference (%)**	**MAC Value Third Phase vs Second Phase**
51.050	51.539	0.948	0.991
87.934	88.911	1.099	0.973
148.754	148.510	0.164	0.976
165.608	165.364	0.148	0.995

Resulted mode shapes are illustrated in the figure 3.5.

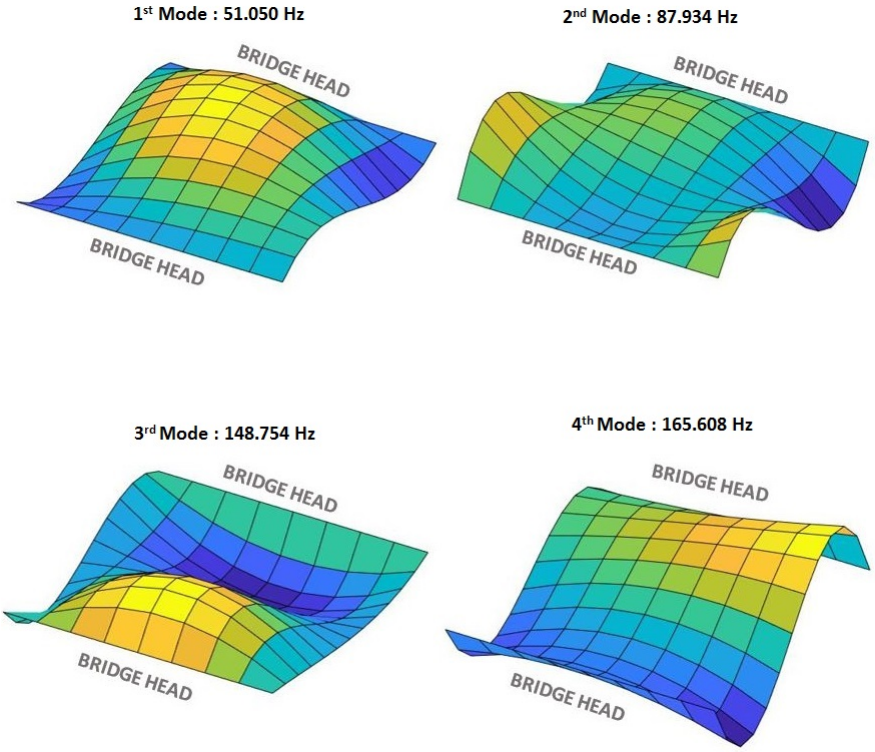

Figure 3.5: Mode Shapes of Pit Bridge Experiment

The result shows that the natural frequency difference between second phase and third phase is minuscule. The resulted mode shape also have good and unique shape. The high MAC values for all mode shapes shows that third phase produce consistent mode shape. These results also signifies that all of the peaks detected are true modal peaks, since they produce consistent result between two separate measurements. In order to find out how well the performance of this distributed FDD compared to normal centralized FDD, a further comparison result are made. The centralized FDD is performed by simply follow the normal FDD process and selecting peaks by manually select the highest value of peaks on the singular value plots. Obtained result is shown in table 3.2.

Table 3.2: Centralized FDD Comparison Result of Pit Bridge Experiment

Third Phase Natural Frequency (Hz)	**Centralized FDD Natural Frequency (Hz)**	**Natural Frequency Difference (%)**	**MAC Value Third Phase vs Centralized FDD**
51.050	51.050	0.000	1.000
87.934	87.934	0.000	1.000
148.754	148.754	0.000	1.000
165.608	165.852	0.147	0.998

The comparison results shows that except fourth mode, all of the third phase modal parameters are same as centralized FDD result. The highest frequency difference is only 0.147 percent and the lowest MAC value is still quite high, which is 0.998. Regarding the data size reduction, the table 3.3 summarized them.

Table 3.3: Data Size Comparison of Pit Bridge Experiment

Data Name	**Data Type**	**Data Type Size (byte)**	**Number of Data**	**Total Data Size (byte)**	**Data Saving Compared To Time Domain Signal (%)**
Time Domain Signal	Float	4	118784	475136	-
Second Phase FFT Result (58 Windows)	Complex Float	8	408	189312	60.16
Third Phase FFT Result (58 Windows, 4 peaks)	Complex Float	8	754	6032	98.73

The data size comparison only includes data that is sent from sensor node to be processed into modal parameters, thus first phase is omitted from the

comparison. The result in table 3.3 assumes that time domain signal is always represented by 32-bit floating point. The FFT result always produce complex number that represented in 32-bit floating point number. Since each complex number consist of two part, imaginary and real number, each data has twice the size of 32-bit floating point number. The data size for second phase and third phase is sum of 58 windows of FFT result data. Third phase data size is also the sum of four peaks, where each peak area consist of 13 frequency bins. The result shows that second phase reduce the data size needed to determine initial natural frequencies quite significantly. Third phase reduce it even further until the data size is only 1.27 percents of original time domain signal data size.

3.2 Steel Arch Bridge

The second experiment is performed on a pedestrian steel arch bridge located in University of Twente campus. The structure of the bridge can be seen in figure 3.6. The structure itself contain two identical bridge (left and right) that does not have any connection to each other. Each part of the bridge has arc length of 499 cm and width 99.6 cm. The deck of the bridge has similar shape as previous pit bridge, comprises of steel meshes.

Figure 3.6: The structure of the steel arch bridge of University of Twente

Six sensor are used to measure the vibration of the bridge. Since there is not any connection between each part of the bridge, they should be treated as separate structure. Thus, the measurement and excitation is done only on one side of the bridge, as can be seen in the figure 3.7.

Figure 3.7: Sensor placement on steel arch bridge of University of Twente

All of the sensors are placed by tightly strap them into the mesh structure of bridge deck. The detailed sensor placement diagram can be seen in figure 3.8.

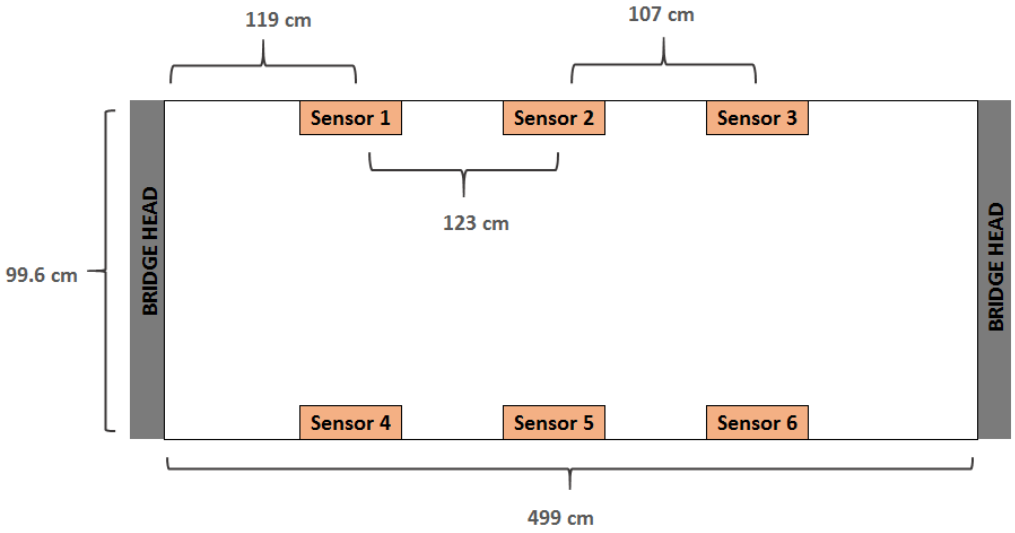

Figure 3.8: Detailed sensor placement diagram on steel arch bridge

The vibration excitation sources are pedestrians and bicycles that crosses the bridge. Vibration is measured in normal direction of the bridge. It is done twice, with each measurement length of approximately 352.26 seconds using sampling rate of 500 Hz. This resulted into two set of data, each has length of 176128 samples. The PSD matrix for second and third phase are constructed using 86 windows of FFT on acquired vibration signals. The example of acquired vibration signal is shown in figure 3.9.

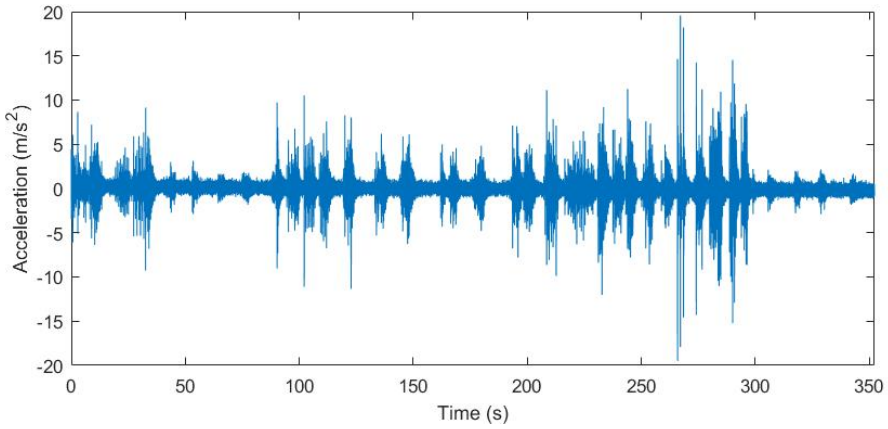

Figure 3.9: Vibration Signal Example of Steel Arch Bridge Measurement

Processing the obtained data using first phase and second phase distributed produce result as shown in figure 3.10.

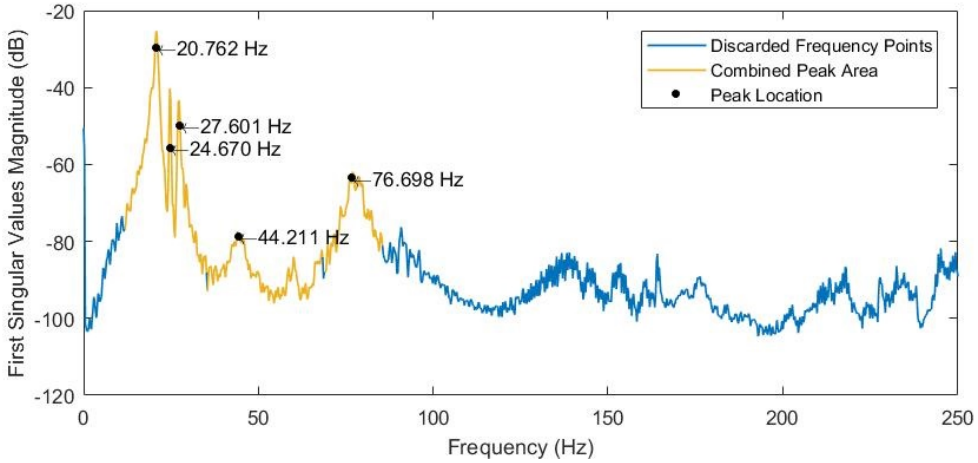

Figure 3.10: Second Phase Result Plot of Steel Arch Bridge Experiment

Figure 3.4 shows that there is not any false positive peak area. Each peak area consist at least one natural frequency. However, it can be noted that the first three peaks are not exactly detected on peak maximum value. This is caused by the smoothing process. Since the first three peak are quite narrow, it is rather affected by the smoothing even though minimal length of filter is used. However, omitting the smoothing process or shorten the filter length even further will not effectively suppress noise peaks from the plots.

As a consequences, higher chance of false positive will occurs. Using natural frequency results obtained from second phase, a third phase distributed FDD is performed. The resulted modal parameters and its comparison to second phase result is summarized in table 3.4.

Table 3.4: Third Phase Result of Steel Arch Bridge Experiment

Third Phase Natural Frequency (Hz)	**Second Phase Natural Frequency (Hz)**	**Natural Frequency Difference (%)**	**MAC Value Third Phase vs Second Phase**
20.762	20.762	0.000	0.985
24.670	24.670	0.000	0.966
27.113	27.601	1.770	0.990
43.967	44.211	0.552	0.888
77.675	76.698	1.274	0.969

Resulted mode shapes illustration is shown in figure 3.11.

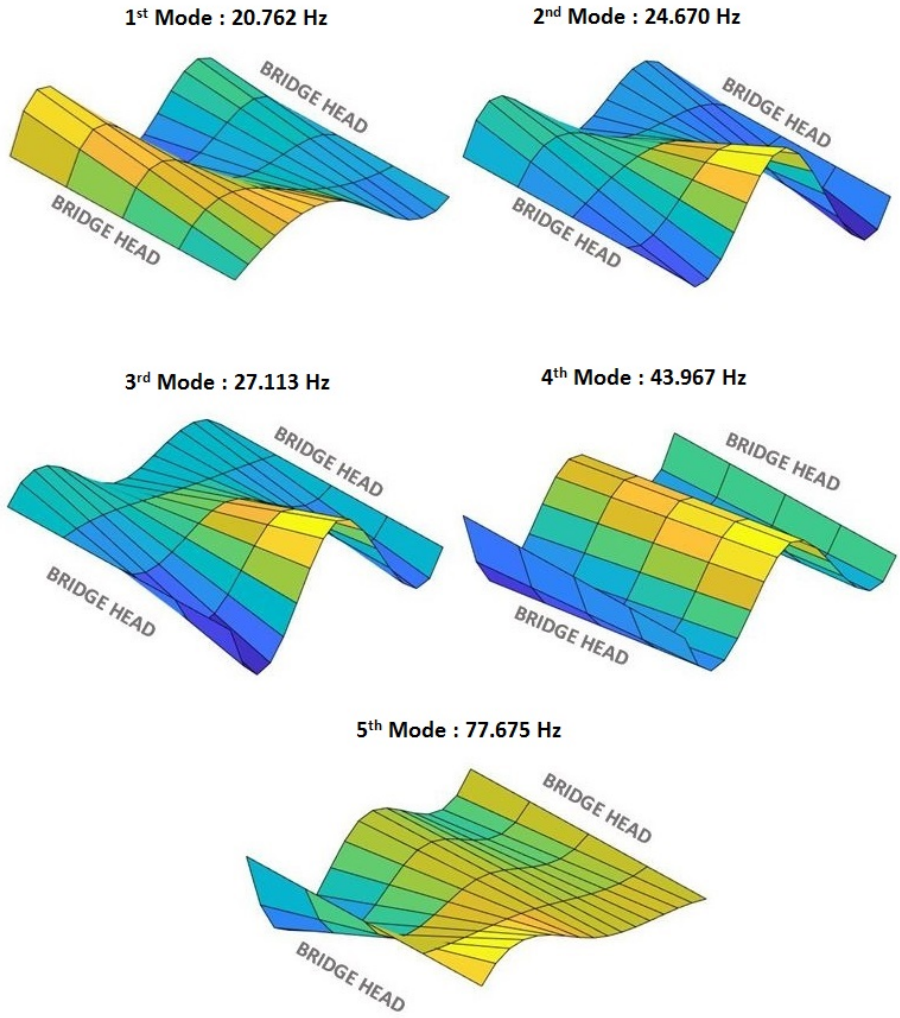

Figure 3.11: Mode Shapes of Steel Arch Bridge Experiment

The result shows that the natural frequency difference between second phase and third phase is minuscule. The mode shapes produced from third phase is also quite consistent with second phase result with MAC values above 0.96, except the fourth mode, which has MAC value of 0.888. However, the three first resulted mode shape are similar to each other. It is suspected that the cause of this similarity is uncaptured vibration in some part of the bridge. Since only six sensors used, and they are all placed on side of the deck, there is a possibility that there is a mode shape movement in the center of bridge or side railing beam. As can be seen in figure 3.12, the side railing is well connected to the bridge girder and could affect the global mode shape of the bridge. Since there is not any sensor placed in center part or railing of the

bridge, the potential movement on that part cannot be shown in obtained mode shapes.

Figure 3.12: Railing Connection on Steel Arch Bridge Experiment

A further comparison result between third phase and normal centralized FDD is also made. The comparison result is shown in table 3.5.

Table 3.5: Centralized FDD Comparison Result of Steel Arch Bridge Experiment

Third Phase Natural Frequency (Hz)	**Centralized FDD Natural Frequency (Hz)**	**Natural Frequency Difference (%)**	**MAC Value Third Phase vs Centralized FDD**
20.762	20.762	0.000	1.000
24.670	24.670	0.000	1.000
27.113	27.113	0.000	1.000
43.967	43.967	0.000	1.000
77.675	78.652	1.242	0.980

The comparison results shows that all of the modal parameters resulted in third phase are consistent with centralized FDD. Four of the modal parameters are basically the same as centralized FDD result. Even though the fourth mode shows lowest MAC value in comparison with second phase result, it actually point to the correct mode if compared to centralized FDD result of second dataset. The highest frequency difference is only 1.242 percent and the lowest MAC value is still very high, with value of 0.980.

Table 3.6: Data Size Comparison of Steel Arch Bridge Experiment

Data Name	Data Type	Data Type Size (byte)	Number of Data	Total Data Size (byte)	Data Saving Compared To Time Domain Signal (%)
Time Domain Signal	Float	4	176128	704512	-
Second Phase FFT Result (86 Windows)	Complex Float	8	26230	209840	70.21
Third Phase FFT Result (86 Windows, 5 peaks)	Complex Float	8	1118	8944	98.73

The data size reduction is shown in table 3.6. The result shows that second phase reduce the data size significantly. As same as result in pit bridge experiment, third phase reduce it even further until the data size is only 1.27 percents of original time domain signal data size.

3.3 Wood Arch Bridge

The third experiment is performed on an arch bridge with wood structure, except the deck, which is made up of multiple beams of asphalt. This bridge is larger than two previous bridge, with the arc length of 12.9 m and width of 2.1 m. The photos of the structure can be seen in figure 3.13.

Figure 3.13: Wood arch bridge of University of Twente

The same six sensors used in previous two experiments are used. They are taped tightly on top of the deck as shown in figure 3.14. The detail of arrangement of the sensor follows diagram shown in figure 3.15.

Figure 3.14: Sensor placed on top of Wood Arch Bridge

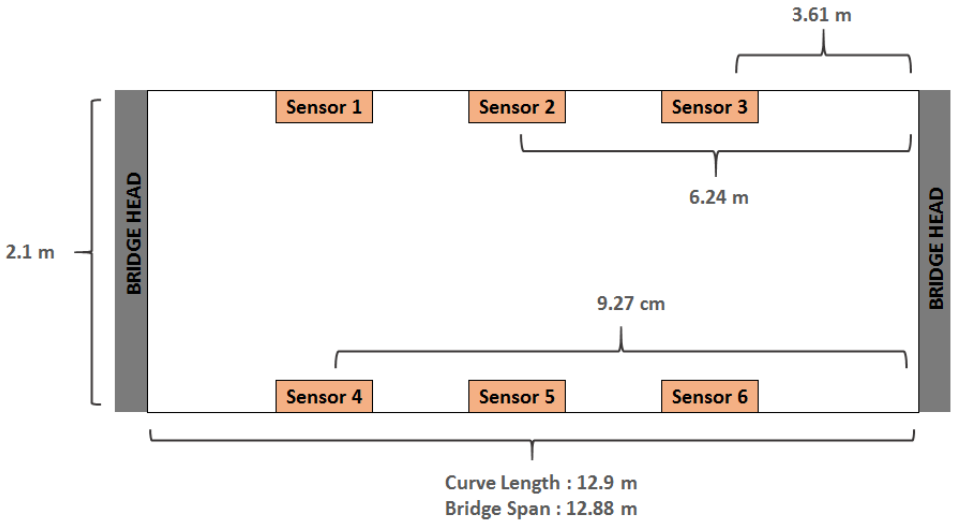

Figure 3.15: Diagram of sensor placement in Wood Arch Bridge

Also as same as previous experiments, the vibration of the bridge is excited by pedestrians and bicycles. Vibration is measured in normal direction of the bridge. The measurement is done twice. Each measurement has length of approximately 172.03 s using sampling rate of 500 Hz. The example of acquired vibration signal is shown in figure 3.16.

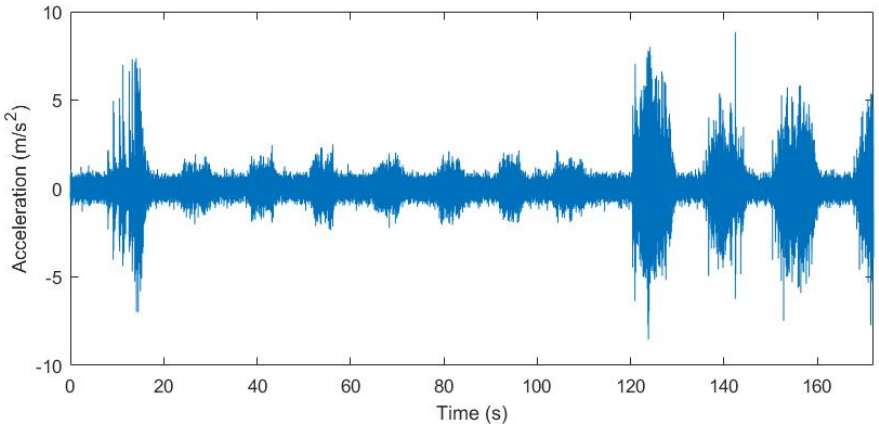

Figure 3.16: Vibration Signal Example of Wood Arch Bridge Measurement

PSD matrix for second and third phase are estimated using 42 windows of 2048 point FFT. Processing the obtained data using first phase and second phase distributed produce result as shown in figure 3.17.

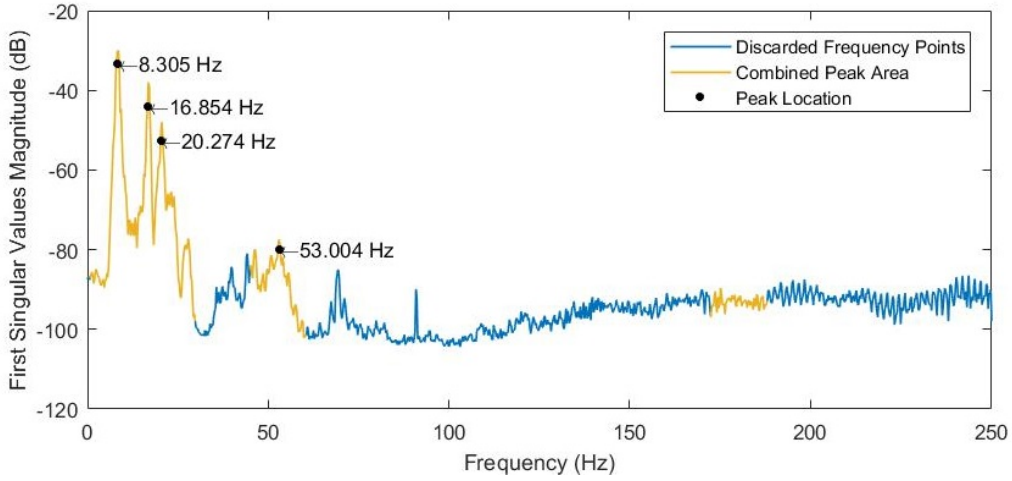

Figure 3.17: Second Phase Result Plot of Wood Arch Bridge Experiment

Figure 3.4 shows that there is one false positive peak area, which is the last peak area. However, second phase reliably determine that there is not any natural frequency peak in that area. This false positive peak area is caused by one of the sensor detect a local peak in its auto-PSD plot. In addition, there are two missed modal peaks located 27.85 Hz and 69.37 Hz. Upon checking

on second dataset (used for third phase), those two peaks also apparent, albeit weaker, as shown in figure 3.18.

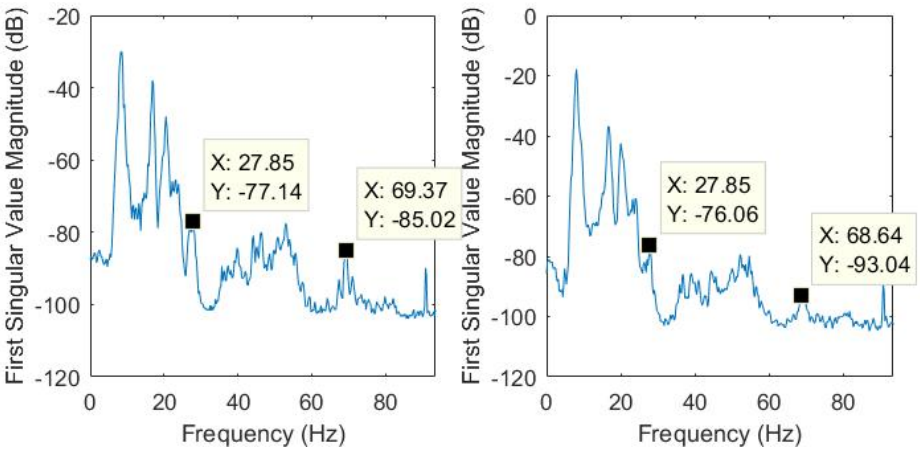

Figure 3.18: Missed Natural Frequency Peaks on Wood Arch Bridge Experiment. Peaks in singular value plot of first dataset (left) and second dataset (right)

Checking the mode shape of those peaks between two dataset result also shows that they correlate well, with MAC value of 0.9898 and 0.9115 respectively. This signifies that they are true modal peak. This problem is caused by the smoothing procedure in first phase and second phase. Those two peaks are suppressed by the smoothing procedure as they are narrow peaks with small height. Reducing the smoothing intensity could increase the possibility of those two peaks detected. However, it also increases the possibility of false-positive peak detection as the singular value plot is quite noisy. This noisy plot is caused by insufficient duration of signal acquisition. As can be noted, the wood arch bridge experiment has the shortest vibration acquisition duration. By increasing the vibration signal length, noise can be further suppressed and natural frequency peak becomes more apparent, which result in more reliable peak selection process.

Using natural frequency results obtained from second phase, a third phase distributed FDD is performed. The resulted modal parameters and its comparison to second phase result is summarized in table 3.7.

Table 3.7: Third Phase Result Wood Arch Bridge Experiment

Third Phase Natural Frequency (Hz)	Second Phase Natural Frequency (Hz)	Natural Frequency Difference (%)	MAC Value Third Phase vs Second Phase
8.061	8.305	2.941	0.965
16.854	16.854	0.000	0.997
20.029	20.274	1.205	0.957
52.272	53.004	1.382	0.911

Resulted mode shapes illustration is shown in figure 3.19.

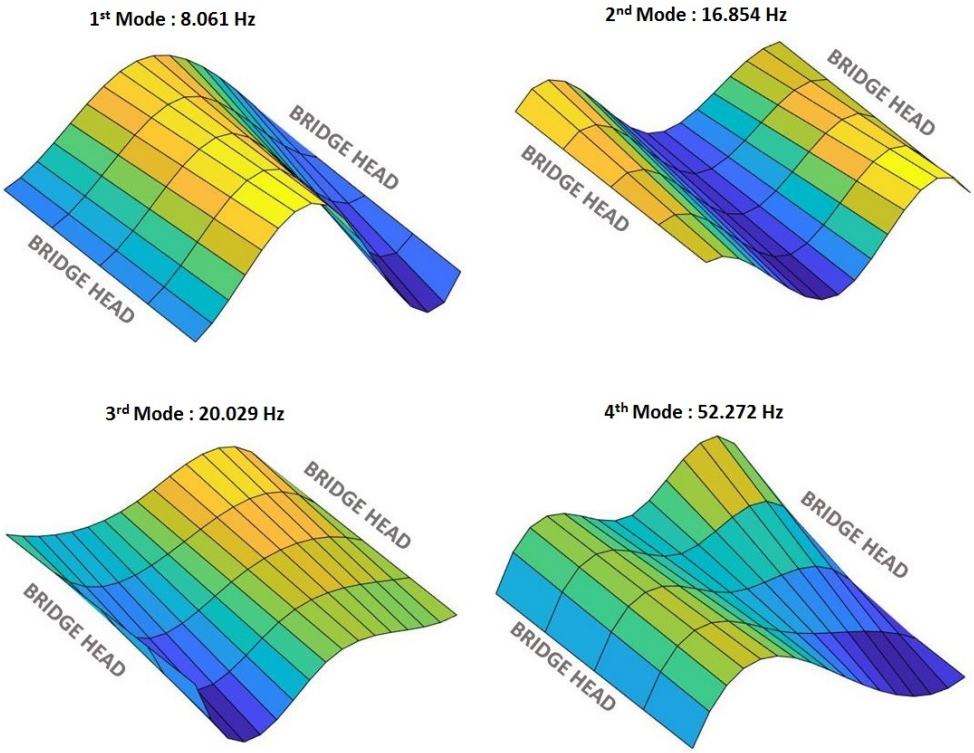

Figure 3.19: Mode Shapes of Steel Arch Bridge Experiment

The result shows that the natural frequency difference between second phase and third phase is insignificant, with maximum difference of 2.941 percents. The mode shapes produced from third phase is also quite consistent with second phase result with MAC values above 0.91. This means that all of the obtained modal parameters are correct since they are consistent between

two measurements. This is further proofed by the fact that all of the mode shapes are unique.

A further comparison result between third phase and normal centralized FDD is performed to see how accurate is the proposed distributed FDD. The comparison result is shown in table 3.8.

Table 3.8: Centralized FDD Comparison Result of Wood Arch Bridge Experiment

Third Phase Natural Frequency (Hz)	**Centralized FDD Natural Frequency (Hz)**	**Natural Frequency Difference (%)**	**MAC Value Third Phase vs Centralized FDD**
8.061	8.061	0.000	1.000
16.854	16.610	1.471	0.995
20.029	20.029	0.000	1.000
52.272	52.272	0.000	1.000

The comparison results shows that all of the modal parameters resulted in third phase are consistent with centralized FDD. Three of the modal parameters are as same as centralized FDD result. The one that is sightly differ, third mode, even shows high MAC value and minuscule natural frequency difference.

Table 3.9: Data Size Comparison of Steel Arch Bridge Experiment

Data Name	**Data Type**	**Data Type Size (byte)**	**Number of Data**	**Total Data Size (byte)**	**Data Saving Compared To Time Domain Signal (%)**
Time Domain Signal	Float	4	86016	344064	-
Second Phase FFT Result (42 Windows)	Complex Float	8	10500	84000	75.59
Third Phase FFT Result (42 Windows, 4 peaks)	Complex Float	8	546	4368	98.73

The data size reduction is shown in table 3.9. The result shows that second phase reduce 75.59 percents of data transmission by transmitting FFT result data that is located in peak areas. Third phase data reduction is as same as two previous experiments, the data transmitted in this phase is only 1.27 percents of original time domain signal data size.

Chapter 4

Practical Review

In order to asses the feasibility of the proposed distributed FDD to be implemented in resource constrained wireless sensor node, a resource usage analysis are performed. There are four main aspect that will be reviewed, which are processing power, memory usage, data transmission, and battery usage. In order to review the each of the resource usage, distributed processes of sensor node side are implemented in real microcontroller.

Since the nature of proposed distributed FDD involves lot of floating point computation and deals with huge data in sensor node side, it is desirable to use microcontroller that has floating point unit and large size of Random Access Memory (RAM). Based on this, STM32F407VG microcontroller made by ST Microelectronic, is chosen for the implementation. It has popular 32-bit ARM Cortex-M4 processor that include floating point unit in the architecture. It also has 192 Kilobytes of RAM, which fits perfectly with the requirement, and 1 Megabytes of Flash memory that could fit large application in it. Moreover, the microcontroller supports operating frequency of 168 MHz, which provides high computational speed capability.

The complete process implementation for all phases of distributed FDD on the microcontroller is illustrated in the figure 4.1. Important thing to take note is that there is not any complementary hardware used, such as real accelerometer and radio hardware used for the implementation. The implementation only focused on microcontroller side. Thus, some process part are not fully implemented.

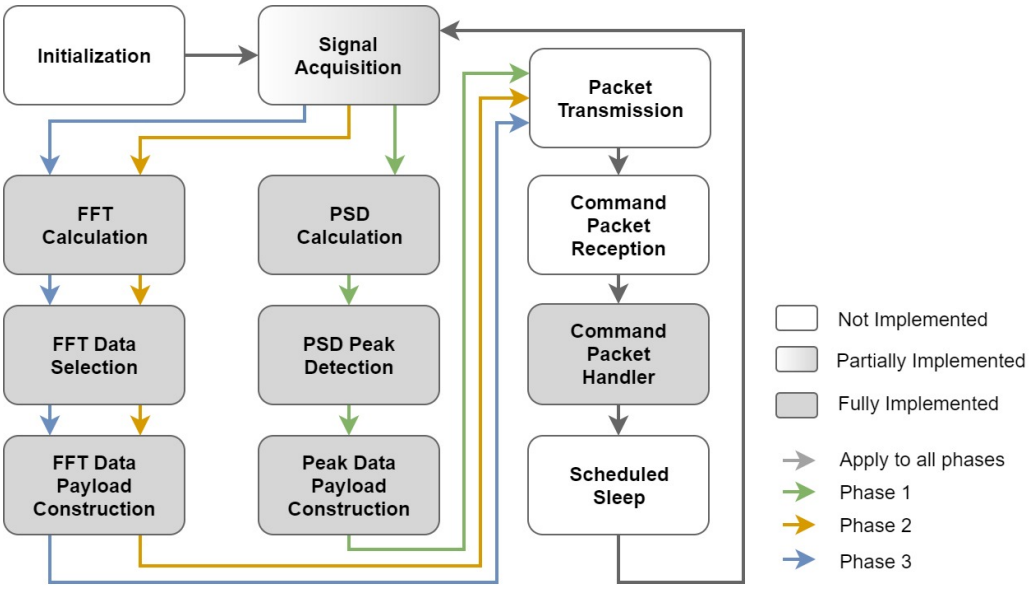

Figure 4.1: Process Diagram of Distributed Processes in Sensor Node Side

The diagram also explain which path of processes are taken for each phase of distributed FDD. The explanation for each process part as follows:

- **Initialization** is used for setting up external peripherals or hardware such as accelerometer and radio module. However, since there is not any external hardware used, this part of process is left empty and do nothing.
- **Signal Acquisiton** is the process where data is polled from accelerometer and arranged to be saved for further processing. However, this process is partially implemented since there is not any real accelerometer used. The data saving process is as same as the case when real accelerometer data is used. The difference is in data polling, where a dummy process is used to poll testing data. Despite that, the difference to real accelerometer data poling is expected to be minimal, since commonly accelerometer use fast SPI communication to transfer data to microcontroller. Signal acquisition is stopped when certain number of samples are already polled. In the previous experiment case, it is the length of signald window needed for FFT, which is 2048 samples.
- **FFT Calculation** is process of transforming the acquired vibration signal into frequency domain using windowed fast fourier transform. Acquired signal is applied by Hanning window before performing the

FFT process. A library provided by ARM called CMSIS-DSP is used for implementing the FFT.

- **FFT Data Selection** is process of selecting FFT result data that needs to be sent based on obtained peak area from phase 1 and phase 2.
- **FFT Data Payload Construction** is process of constructing packet data payload that will be sent via radio transmission to central node. This process involves creating header, splitting data into set of packets, and arranging data in packet payload.
- **PSD Calculation** is process of creating auto power spectral density plot in the first phase to determine peak location candidate for peak area construction. This process also involves multiple windowed FFT calculation similar to FFT calculation process part mentioned previously.
- **PSD Peak Calculation** is process to be run on the first phase for determining the peak locations in PSD plot.
- **Packet Transmission** is process where microcontroller instruct the radio hardware to send required data. This process is not different for each type of payload needed to be sent. Since there is not any real radio hardware used, this process is not implemented. It is expected that this process execution time is very short such that it is insignificant, since it only involves fast microcontroller to radio module communication.
- **Command Packet Reception** is process where microcontroller check and communicate with radio hardware for incoming packet from central node. Due to similar reason as Packet Transmission process, this process is not implemented and it is expected that the execution time is insignificant.
- **Command Packet Handler** is a process where received packets from central node are processed into set of meaningful data that will be saved for further processing. The kind of data involves are instruction to start first phase, and peak area information result from second and third phase. This process is fully implemented and tested using fake packet data that replicates real command packets.
- **Scheduled Sleep** is a process where microcontroller finished all of process of current phase, starting to initiate sleep mode where all of the hardware usage is minimized, and schedule wake up by using real time clock (internal or external). Commonly, the main RAM content

and computational register values are erased while microcontroller is in deep sleep mode. Therefore, an important data backup mechanism to permanent memory (such as flash) or backup SRAM (STM32F407VG have one) are performed before sleeping. Aside of that, the common operation involves setting up some microcontroller registers and real time clock. This process is not implemented and it is expected that the execution time is minuscule since the heaviest operation is only fast copying data from register or RAM to another memory location.

The clock speed of microcontroller is set to 168 MHz. All of the processes are written in a application that is run directly on top of STM32F407VG without any medium layer between such as operating system. The application is compiled by open source ARM-GCC compiler using O2 optimization flag. Each resource usage analysis is explained further in following subsections.

4.1 Processing Power

The metric that will be used to see how much the processing power toll on wireless sensor node is processing time. The longer processing time required to compute complete processes of each phases of proposed distributed FDD, the higher power consumption for microcontroller usage. If the processes are too heavy and takes very long time to finish, it is possible that the power consumption for computation is similar to or more than the power used for data transmission, which could defeat the purpose of data reduction on distributed process.

In order to measure time the execution time of each part of process, a special cycle counter register inside ARM Cortex-M4 is used. The execution time is approximated by polling the cycle counter inside this register and divide it by CPU clock speed. Since the execution time depends on various process setting and vibration signal acquisition length, the test follows scenario used in previous three bridge experiments. The obtained results are shown in table 4.1, table 4.2, and table 4.3.

Table 4.1: First Phase Processes Execution Time

First Phase Process	**Pit Bridge Total Process Execution Time (μs)**	**Steel Bridge Total Process Execution Time (μs)**	**Wood Bridge Total Process Execution Time (μs)**
Signal Acquisiton	27.95	27.96	27.93
PSD Calculation	98780.00	98788.00	98856.00
PSD Peak Detection	1461.60	1465.10	1458.80
Peak Data Payload Construction	0.29	0.29	0.29
Command Packet Handler	1.28	1.28	1.27

Table 4.2: Second Phase Processes Execution Time

Second Phase Process	**Pit Bridge Total Process Execution Time (μs)**	**Steel Bridge Total Process Execution Time (μs)**	**Wood Bridge Total Process Execution Time (μs)**
Signal Acquisition	402.26	596.63	291.34
FFT Calculation	68005.00	100835.00	49245.00
FFT Data Selection	1194.05	1770.40	700.01
FFT Data Payload Construction	4409.04	6537.72	2528.57
Command Packet Handler	83.20	123.37	59.72

Table 4.3: Third Phase Processes Execution Time

Third Phase Process	**Pit Bridge Total Process Execution Time (μs)**	**Steel Bridge Total Process Execution Time (μs)**	**Wood Bridge Total Process Execution Time (μs)**
Signal Acquisition	402.39	596.69	291.40
FFT Calculation	68005.00	100835.00	49245.00
FFT Data Selection	420.33	623.41	251.80
FFT Data Payload Construction	1278.44	1895.53	713.79
Command Packet Handler	53.17	78.84	38.50

It can be seen from all of the table, overall, all of the processes is completed very quickly. The highest execution time is FFT calculation on steel bridge

experiment scenario, where the highest number of vibration signal windows (86 windows) are applied by windowed FFT. However, the execution only requires 100.835 milliseconds. The total execution time for all of the phase, in all of the bridge experiments, are below than 1 second. In this case, microcontroller spend more time idle waiting for accelerometer data, which can be batch polled, or idle waiting command packet from central node than processing the vibration signals. In both case, microcontroller can be put in lower energy consumption mode. Therefore, processing power is not a problem for sensor node development, seeing there are lot of cheap microcontroller with ARM Cortex-M4 processor aside of STM32F407VG. Of course the execution time can be increased if user set the required vibration signal duration to be longer, which means higher number of phase 2 and phase 3 process iteration. However, the total execution is very small such that even though the length is increased to 100 times longer, the required total execution time for each phase is still in unit of seconds.

4.2 Memory Usage

The memory usage of interest in this case is Random Access Memory (RAM) usage. Since majority of operations in the sensor nodes involves large data, RAM bounds to be the bottleneck of the implementation. Using GNU ARM Print Size utility, information about RAM usage can be obtained immediately after compilation time. However, it cannot determine which are variables that contribute to the total RAM usage and their portion of usage. In this case, simple manual calculation is needed. All of three bridge experiments scenario have same RAM usage, since it mainly only depends on the FFT related process setting (window size and number of point of FFT), which are set to same values for all experiments. The resulted RAM usage details is shown in table 4.4.

Table 4.4: RAM Usage Detail

Data Name	**Data Type**	**Type Size (Bytes)**	**Number of Data**	**Total Data Size (Bytes)**
Vibration Signal	Float	4	2048	8192
Auto PSD Data	Float	4	1024	4096
Peak Locations	Integer 16-bit	2	20	40
FFT Result Data	Complex Float	8	1024	8192
Peak Area Location Indices	2 x Integer 16-bit	4	20	80
Number of Iteration	Integer 16-bit	2	1	2
Stack Size	-	-	-	4096
Others	-	-	-	986
			Total RAM Usage	25684

Vibration Signal is used to store acquired vibration from accelerometer, it holds 2048 samples of data, which is equal to the window size for FFT and PSD calculation. Auto PSD Data holds the auto-PSD result calculation. Since auto-PSD always produces real result and half of the frequency points are unused, it is sufficient to hold only 1024 samples of floating point number. FFT result data contains data resulted from FFT operation of all of the distributed FDD phases. Peak locations are used to store the location of peak obtained from first phase. Peak area location indices is used to store information about peak areas obtained from central node as a result of second and third phase. Each of its data contain one number that signifies in which location the peak area start, and one number that signifies where the peak area should end. Both Peak Area Location Indices and Peak Locations are set to contain a fixed number of data, up to maximum 20 data. Number of Iteration store the information from central node about how many process iteration needed to finish second or third phase. This number of iteration correspond to the length of vibration signal that is needed to be processed for complete modal analysis in second or third phase. In microcontroller, stack size should be set manually to a fixed size. Understanding that some of the functions involve large data operation, in order to avoid stack overflow, it is set to 4096 bytes. There are some others data that uses 986 bytes which cannot be tracked, this could consists of global variables from external libraries or board and processor startup routine.

The total RAM usage is 25.684 kilobytes, which is significantly lower than STM32F407VG RAM size. This leaves lot of space to be used for another unimplemented processes or heap if needed. This size of RAM usage also pose no problem if implementation needs to be done in other ARM Cortex-M4

based microcontroller. Microcontroller that uses ARM Cortex-M4 processor often includes large RAM in it. To give some example, NRF52 from Nordic Semiconductor has 64 kilobytes of RAM and SAM4S from Atmel provides 160 kilobytes of RAM.

One thing to take a note is that the RAM usage is independent from the required length of vibration signal. Previous Wood Arch Bridge experiment, that involves vibration signal length of 172.032 seconds (42 windows of 2048 samples) use the same RAM size as Pit Bridge experiment that uses 352.26 second long of vibration signal (86 windows of 2048 samples). No matter how long the required signal duration to be processed, they basically use the same set of processes and data storage over and over again for each windows until sufficient number of windows processed. Each iteration, the old data is replaced by newly produced data as it no longer serves any use. However, the RAM usage can be changed if different number of point of FFT used. In current experiments case, 2048-point FFT is used. If it changed into 4096-point FFT, the RAM usage could increase to almost twice the size of RAM usage in 2048-point FFT scenario. Therefore, it is advisable to use as minimum number of point of FFT as possible, as long as it meets the frequency resolution requirements.

4.3 Data Transmission

The data reduction comparison that has been shown in previous chapter illustrates how significant the proposed distributed FDD able to reduce the data needs to be transmitted. However, in real practice, the data transmission will always be larger. This is due to packet header and payload size limitation for single packet transmission. Both of them are unique to the communication protocol and the radio hardware being used. Thus, in order review the practicality of proposed distributed FDD from data transmission perspective, a specific radio hardware should be selected for hypothetical testing scenario.

Since proposed distributed FDD still involves lot of data transmission, even after data reduction, a wireless protocol and radio hardware with sufficiently high data rate is needed. It also need to be able to cover large area as there is a possibility the system will be used in large-sized bridge. One protocol that comes up in mind when think about large area coverage is LoRa network. However, the data rate of LoRa is too slow. The typical maximum data rate is around 5468 bits per-second. The slow data rate makes LoRa energy inefficient for sending large data. Based on these requirements, ZigBee is

a good protocol candidate to be used in real application. It has high data rate, typically 250 kilobits per-second (31.25 kBps). It can cover large area by utilizing its capability to construct a mesh network. Therefore, ZigBee is used for data transmission review. A popular ZigBee radio module, Xbee ZB is chosen. It has maximum outdoor range capability of 120 m and indoor range of 40 m in single hop.

Xbee ZB have payload size limitation of 84 bytes for each packet. It also needs a 13 byte headers for ZigBee protocol. Another 3 byte custom header is needed for distributed FDD usage. The first byte signifies what type of packet is transmitted or received, second byte signifies the total number of packets is sent in current communication session, and third byte signifies the packet number. Last two bytes are used in case transmitted data are splitted into multiple packets. Using these two bytes, central node can track the order of packets and easily merge the them into one single set of data again. As can be observed, the data transmission size tends to be larger, especially on FFT result data. Payload size limitation force the FFT result data to be splitted into multiple packet. Each packet contain its own header. The resulted final transmission size are shown in table 4.5 for Pit Bridge experiment, table 4.6 for Steel Arch Bridge experiment, and table 4.7 for Wood Arch Bridge experiment. The original data size is based on previous chapter experiment results.

Table 4.5: Data Transmission Details for Pit Bridge experiment

Packet Type	**Data Size (bytes)**	**Header Size (bytes)**	**Number Of Packet**	**Total Transmission Size (bytes)**	**Transmission Duration (ms)**
First Phase Peak Locations	10	16	1	26	0.80
Second Phase FFT Data	189312	16	2254	225360	6935.38
Second Phase Peak Areas	20	16	1	36	1.11
Third Phase FFT Data	6032	16	72	7168	221.54
Third Phase Peak Areas	16	16	1	32	0.98

Table 4.6: Data Transmission Details for Steel Arch Bridge experiment

Packet Type	**Data Size (bytes)**	**Header Size (bytes)**	**Number Of Packet**	**Total Transmission Size (bytes)**	**Transmission Duration (ms)**
First Phase Peak Locations	10	16	1	26	0.80
Second Phase FFT Data	209840	16	2499	249808	7689.23
Second Phase Peak Areas	12	16	1	28	0.86
Third Phase FFT Data	8944	16	107	10640	329.23
Third Phase Peak Areas	20	16	1	36	1.11

Table 4.7: Data Transmission Details for Wood Arch Bridge experiment

Packet Type	**Data Size (bytes)**	**Header Size (bytes)**	**Number Of Packet**	**Total Transmission Size (bytes)**	**Transmission Duration (ms)**
First Phase Peak Locations	10	16	1	26	0.80
Second Phase FFT Data	84000	16	1000	100000	3076.92
Second Phase Peak Areas	12	16	1	28	0.86
Third Phase FFT Data	4368	16	52	5200	160.00
Third Phase Peak Areas	16	16	1	32	0.98

As can be seen in second phase and third phase FFT data transmission, there is an increase of total transmission size due to packet splitting and header, which are around 19 percents compared to original data size. Inspecting transmission duration, which is based on assumption that data is transmitted in full rate of 32.5 kBps, shows that second phase spend significantly higher transmission time than third phase. However, since first phase and second phase can be run only once throughout the sytem life, third phase data is more interesting to be observed. In long term, only third phase is run repetitively to obtain frequently update modal parameters data. Due to high data rate of Xbee, the transmission of third phase FFT data can be done briefly. If third phase is done only once per hour, the active duty cycle

is very low such that battery life can be preserved well. How long a battery life preserve will be further explained in next section.

4.4 Energy Consumption

Mainly there are three main energy consumer in sensor node, which are microcontroller, radio module, and accelerometer (or other vibrational sensor). In some cases, external real time clock is used to schedule sleep and wake up. However, STM32F407VG has internal RTC to support its energy saving mode. As same as other resource usage review, it is needed to make a use case using real hardware data, as without it is not possible to estimate energy consumption. Microcontroller and radio module hardware choice is already clear. For convenience purpose, a popular digital output accelerometer, LIS3DH, made by STMicroelectronics is used for energy consumption review. Using datasheet of each hardware, a summary of current consumption is shown in table 4.8.

Table 4.8: Current Consumption Summary of Main Hardware

Hardware State	**Current Consumption (mA)**
Microcontroller Active	87
Microcontroller Sleep	0.0003
Radio Transmit	45
Radio Receive	40
Radio Sleep	0.001
Accelerometer Active	0.011
Accelerometer Sleep	0.0005

Using obtained sensor node process execution time, data transmission time, and vibration signal acquisition duration from previous sections and chapter, duty cycles of each hardware state in each bridge experiment scenario are calculated. This duty cycles data are used to find effective current consumption that will be used further to estimate battery life of sensor node. The cycle is set such that after first and second phase, every one hour, the third phase is activated to obtain updated modal parameters of the bridge. The resulted duty cycle and effective current consumption are shown in table 4.9 for first phase, table 4.10 for second phase, and table 4.11 for third phase.

Table 4.9: Current Consumption Details of First Phase

Hardware State on First Phase	**Pit Bridge Experiment**		**Steel Bridge Experiment**		**Wooden Bridge Experiment**	
	Duty Cycle (%)	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**
Microcontroller Active	0.0028	2.42	0.0028	2.42	0.0028	2.42
Microcontroller Sleep	99.9972	0.30	99.9972	0.30	99.9972	0.30
Radio Transmit	0.0007	0.30	0.0007	0.30	0.0007	0.30
Radio Receive	0.0009	0.37	0.0007	0.29	0.0007	0.29
Radio Sleep	99.9984	1.00	99.9986	1.00	99.9986	1.00
Accelerometer Active	0.4551	0.05	0.4551	0.05	0.4551	0.05
Accelerometer Sleep	99.5449	0.50	99.5449	0.50	99.5449	0.50
	Total	4.94	**Total**	4.86	**Total**	4.86

Table 4.10: Current Consumption Details of Second Phase

Hardware State on Second Phase	**Pit Bridge Experiment**		**Steel Bridge Experiment**		**Wooden Bridge Experiment**	
	Duty Cycle (%)	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**
Microcontroller Active	0.0021	1.79	0.0031	2.66	0.0015	1.28
Microcontroller Sleep	99.9979	0.30	99.9969	0.30	99.9985	0.30
Radio Transmit	5.7785	2600.31	6.4053	2882.40	2.5641	1153.85
Radio Receive	0.0008	0.33	0.0009	0.37	0.0008	0.33
Radio Sleep	94.2207	0.94	93.5937	0.94	97.4351	0.97
Accelerometer Active	6.5991	0.73	9.7849	1.08	4.7787	0.53
Accelerometer Sleep	93.4009	0.47	90.2151	0.45	95.2213	0.48
	Total	2604.86	**Total**	2888.19	**Total**	1157.73

Table 4.11: Current Consumption Details of Third Phase

Hardware State on Third Phase	**Pit Bridge Experiment**		**Steel Bridge Experiment**		**Wooden Bridge Experiment**	
	Duty Cycle (%)	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**	**Duty Cycle (%)**	**Effective Current Consumption (μA)**
Microcontroller Active	0.0019	1.70	0.0019	1.70	0.0014	1.22
Microcontroller Sleep	99.9981	0.30	99.9981	0.30	99.9986	0.30
Radio Transmit	0.1838	82.71	0.2728	122.77	0.1333	60.00
Radio Receive	0.0008	0.33	0.0008	0.33	0.0008	0.33
Radio Sleep	99.8154	1.00	99.7264	1.00	99.8658	1.00
Accelerometer Active	6.5991	0.73	9.7849	1.08	4.7787	0.53
Accelerometer Sleep	93.4009	0.47	90.2151	0.45	95.2213	0.48
	Total	87.22	**Total**	127.62	**Total**	63.85

Since there is a possibility of using multihop transmission in ZigBee and possibility of transmission loss, the radio transmit and receive state is modified for all of the phases. The original transmission and reception duration shown in previous section is multiplied by 30 to accommodate the possibility of using 5 hops with 6 times retransmission for each hop for worst case scenario when there are multiple packet loss. Current consumption data from all of the phases shows that the dominant energy consumer is indeed the wireless communication.

Based the total effective current consumption of all of the phases in tables above, battery life in hours can be estimated using following formula:

$$BatteryLife = \frac{E_{batt} \times \eta_{batt} - i_{tphase1} - i_{tphase2}}{i_{tphase3}} \tag{4.1}$$

where E_{batt} is battery capacity (in mAh), η_{batt} is battery efficiency (in percent), $i_{tphase1}$ is total effective current consumption of first phase (in mA), $i_{tphase2}$ is total effective current consumption of second phase (in mA), and $i_{tphase3}$ is total effective current consumption of third phase (in mA).

Using battery with capacity of 2000 mAh and battery efficiency of 70 percents, the estimated battery life of sensor node in pit bridge experiment, steel bridge experiment, and wooden bridge experiments are 1.83 years, 1.25 years, and 2.5 years respectively. These results show that even using a rather small battery, sensor node could achieve long lifetime by using proposed distributed FDD for operational modal analysis.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

In this thesis, a distributed Operational Modal Analysis for wireless sensor network has been designed. It is based on Frequency Domain Decomposition (FDD) algorithm. However, instead of applying FDD to the full frequency points (half of the sampling rate), the FDD is applied only to small frequency point area around natural frequencies. By performing FFT operation part of FDD locally in sensor node, data transmission size could be reduced significantly.

Operational modal analysis experiment on three real bridge is performed to assess the capability of the proposed distributed FDD. Overall, the result shows that it can reliably automatically extract true modal parameters out of vibration signal data, except on one occasion. In one of the bridge experiment, the process missed two out six modal parameters. The result also shows the size of data that needs to be transmitted from sensor node is reduced significantly, with size of only 1.27 percents of the raw time-domain vibration signal.

Moreover, practical feasibility review is also performed by implementing the process on real microcontroller, an ARM-Cortex M4 based STM32F407VG. Hypothetical hardware usage scenario is also used to inspect the resource usage further. The wireless communication is assumed to be performed via ZigBee protocol using Xbee ZB radio, and vibration signal is acquired via LIS3DH accelerometer. The result shows that all of the processes are executed quickly on the sensor node, with the highest execution time of 100.835 ms. The total RAM usage of 25.684 kilobytes also fits within the microcontroller RAM size and expected to be fit with other ARM-Cortex M4 based

microcontrollers. The energy consumption estimation , based on the all of the bridge experiment usage scenario, shows that sensor node lifetime could exceed one year using only 2000 mAh battery.

In conclusion, the proposed distributed FDD shows good promise for real wireless sensor network implementation as it can automatically and reliably detect modal parameters from real data while using minimum resources on wireless sensor node.

5.2 Recommendation

However, there are still some limitation on the process. Some of the process parameters are still set manually, especially in the peaks location and peak area selection. It is desirable if smoothing strength could be adapted automatically based on the intensity of noise in PSD plot or First Singular Value plot. If this can be applied, the smoothing process could produce higher quality of plot and reduce the possibility of false negative or false positive peak detection. A noise intensity metric based on the amount of local maxima and minima could be developed to pursue this.

Furthermore, current peak area is set into fixed width for all natural frequencies. This width could be unnecessarily too large. Width that is too large for frequency peaks that is closely spaced neighboring frequency peak will potentially produce erroneous result. The automatic peak selection could falsely select the neighboring natural frequency instead, if it is more dominant than the correct natural frequency. The peak area width should be adapted based on percentage of each natural frequency peak width itself.

Also, one thing that is learn from the experiment is the longer vibration signal acquired for FDD process, the higher quality of singular value plot. Acquired vibration signal that is too short in its duration will produce singular value plot that contains lot of noise. However, unnecessarily long vibration signal will affect the data transmission size, since it means more signal windows are applied by FFT calculation. Therefore, it is desirable to find a method to optimize the required length of vibration signal for the process.

References

[1] Scott W Doebling, Charles R Farrar, Michael B Prime, and Daniel W Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. 1996.

[2] Wei Fan and Pizhong Qiao. Vibration-based damage identification methods: a review and comparative study. *Structural Health Monitoring*, 10(1):83–111, 2011.

[3] Peter C Chang, Alison Flatau, and SC Liu. Review paper: health monitoring of civil infrastructure. *Structural health monitoring*, 2(3):257–267, 2003.

[4] MM Abdel Wahab and Guido De Roeck. Damage detection in bridges using modal curvatures: application to a real damage scenario. *Journal of Sound and vibration*, 226(2):217–235, 1999.

[5] Kai-Chun Chang and Chul-Woo Kim. Modal-parameter identification and vibration-based damage detection of a damaged steel truss bridge. *Engineering Structures*, 122:156–173, 2016.

[6] Charles R Farrar and David A Jauregui. Comparative study of damage identification algorithms applied to a bridge: Ii. numerical study. *Smart materials and structures*, 7(5):720, 1998.

[7] Peter Van Overschee and Bart De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[8] Soojin Cho, Jong-Woong Park, and Sung-Han Sim. Decentralized system identification using stochastic subspace identification for wireless sensor networks. *Sensors*, 15(4):8131–8145, 2015.

[9] Sung-Han Sim, BF Spencer, M Zhang, and H Xie. Automated decentralized modal analysis using smart sensors. *Structural Control and Health Monitoring*, 17(8):872–894, 2010.

[10] Rune Brincker, Lingmi Zhang, and P Andersen. Modal identification from ambient responses using frequency domain decomposition. In *Proc. of the 18*‘International Modal Analysis Conference (IMAC), San Antonio, Texas*, 2000.

[11] Julius S Bendat and Allan G Piersol. Engineering applications of correlation and spectral analysis. *New York, Wiley-Interscience, 1980. 315 p.*, 1980.

[12] Jin-Hak Yi and Chung-Bang Yun. Comparative study on modal identification methods using output-only information. *Structural Engineering and Mechanics*, 17(3_4):445–466, 2004.

[13] Bart Peeters and Guido De Roeck. Stochastic system identification for operational modal analysis: a review. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):659–667, 2001.

[14] Randall J Allemang and David L Brown. A correlation coefficient for modal vector analysis. In *Proceedings of the 1st international modal analysis conference*, volume 1, pages 110–116. Orlando: Union College Press, 1982.

[15] A Cunha, Elsa Caetano, Rune Brincker, and Palle Andersen. Identification from the natural response of vasco da gama bridge. In *XXII International Modal Analysis Conference*, 2004.

[16] Jian-Huang Weng, Chin-Hsiung Loh, Jerome P Lynch, Kung-Chun Lu, Pei-Yang Lin, and Yang Wang. Output-only modal identification of a cable-stayed bridge using wireless monitoring systems. *Engineering Structures*, 30(7):1820–1830, 2008.

[17] AT Zimmerman, RA Swartz, and JP Lynch. Automated identification of modal properties in a steel bridge instrumented with a dense wireless sensor network. *Bridge Maintenance, Safety Management, Health Monitoring and Informatics*, pages 1608–1615, 2008.

[18] Andrew T Zimmerman, Michihito Shiraishi, R Andrew Swartz, and Jerome P Lynch. Automated modal parameter estimation by parallel processing within wireless monitoring systems. *Journal of Infrastructure Systems*, 14(1):102–113, 2008.

[19] Gregory Hackmann, Weijun Guo, Guirong Yan, Zhuoxiong Sun, Chenyang Lu, and Shirley Dyke. Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):63–72, 2014.

[20] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[21] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[22] N Yoder. Peakfinder. *Internet: http://www. mathworks. com-/matlabcentral/fileexchange/25500*, 2011.