An Architecture for On-Board Frequency Domain Analysis of Flight Vehicle Vibration Signals

Submitted in partial fulfilment of the requirements

for the award of the degree of

Doctor of Philosophy

by

R Srinivasan
(Roll No: 715049)

Under the supervision of

Prof. B. Lakshmi

Dr. Tessy Thomas

Department of Electronics & Communication Engineering

National Institute of Technology Warangal

Telangana, India - 506004

2023

Dedicated

To

My late Parents
Shri G Ramadorai & Smt R Mangalam,

Wife Smt B Rajeswari

Daughter Ku S Subhashini

and

Foster Son Chi RM Lakshman

Approval Sheet

This thesis entitled **An Architecture for On-Board Frequency Domain Analysis of Flight Vehicle Vibration Signals** by **R Srinivasan** is approved for the degree of **Doctor of Philosophy**.

Examiners

Research Supervisors

Prof. B. Lakshmi
Department of ECE
NIT Warangal, India-506004

Dr. Tessy Thomas
Defence R&D Orgn.
Hyderabad/Bengaluru

Chairman & Head

Prof. P. Sreehari Rao
Department of ECE
NIT Warangal, India-506004

Place:

Date:

Declaration

This is to certify that the work presented in this thesis entitled **An Architecture for On-Board Frequency Domain Analysis of Flight Vehicle Vibration Signals** is a bonafide work done by me under the supervision of **Prof. B. Lakshmi** and **Dr. Tessy Thomas** and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my own ideas and even considered others ideas that are adequately cited and further referenced the original sources. I understand that any violation of the above will cause disciplinary action by the institute and can also evoke panel action from the sources or from whom proper permission has not been taken when needed. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea or data or fact or source in my submission.

Place:

Date:

R Srinivasan
Research Scholar
Roll No.: 715049

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL, INDIA-506004

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the thesis work entitled **An Architecture for On-Board Frequency Domain Analysis of Flight Vehicle Vibration Signals** is a bonafide record of work carried out by **R Srinivasan** submitted to the faculty of **Electronics & Communication Engineering** department, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in **Electronics and Communication Engineering, National Institute of Technology Warangal, India-506004**. The contributions embodied in this thesis have not been submitted to any other university or institute for the award of any degree.

Prof. B. Lakshmi
Research Supervisor
Department of ECE
NIT Warangal, India-506 004.

Place:

Date:

Acknowledgements

First, I take immense pleasure to convey my sincere gratitude to my supervisor Prof. B. Lakshmi for her perpetual encouragement and supervision. Her guidance has oriented me in a proper direction and supported me with promptness and care. I extend my sincere gratitude to Dr. Tessy Thomas, Outstanding Scientist & DG (Aero) for all her encouragement to take up this research work and her patience to keep me motivated.

I take this privilege to thank all my Doctoral Scrutiny Committee members Prof. P Srihari Rao, Prof. L. Anjaneyulu, Prof. C. B. Rama Rao, Dr. P. Muralidhar, Prof. N. Subrahmanyam (Dept. of EEE), Prof. N. V. S. N. Sarma, Prof. T. Kishore Kumar, and Prof. N. Bheema Rao for their detailed review, constructive suggestions and excellent advice during the progress of this research work. I would also like to thank all the faculty of the Dept. of ECE who helped me during the course.

I would like to convey my sincere gratitude to my colleagues Shri.S Y Veerabadra Reddy, Shri. P Sampath Kumar, Shri.A Karthik and Shri.Abhishek A Baghat of RCI, DRDO for their invaluable suggestions and support during the course of this work. I would also like to extend my heartfelt appreciation to my family, colleagues, scholars, friends, and well-wishers for their continuous support and motivation. A special thanks to the Director ASL, DRDO and the Director, RCI, DRDO for granting to do research work at NIT warangal. I am grateful to the management of NIT, Warangal for providing me with an opportunity to carry out my work at the institute.

R Srinivasan

Abstract

The dynamic properties of the airborne structures play a crucial role in the stability of the vehicle during flight. The Natural Frequency, Mode Shapes and Damping Factor are the dynamic properties of the structures which are estimated by Modal Analysis. The configuration and design of the Launch vehicle depends on the application, purpose and other parameters like range, type of payload, launch platform and user requirement. To maintain the stability of the vehicle and drive the vehicle to the intended target, auto-pilot algorithms are coded onto the On-Board computer along with necessary Control Actuation Systems and avionic sub-systems. Random vibrations are encountered by the flight vehicle structure and the sub-systems due to propulsion load, aerodynamic load, acoustic load, control forces, wind gusts, etc. These forces strongly interact with the flight structures and cause mechanical vibrations throughout the length of the launch vehicle and also at the interface with the payload. The Modal and Spectral behaviour of the vehicle structure is one of the important factors in finalising the auto-pilot algorithm to ensure stability of the vehicle and the control margins for effective performance of the control systems during the flight. It is ensured during the mission design, that the modal parameters of the structure does not enter the control loop through control-structure interaction.

Currently, the model and spectral behaviour of the launch vehicle is handled in the auto-pilot algorithm based on the Structural Analysis of the flight structure, Ground Resonance Test data and simulation of flight trajectory in Hardware-In-Loop-Simulation test bed. The modal frequency values and the damping factor of the structure from the ground tests are used in the design of the notch filter in autopilot. The results of the ground resonance test and the random vibration tests are compared with the simulation results to fine tune either the model or the test conditions or both in many occasions. Fur-

ther corrections to the algorithm are carried out based on post-flight analysis of telemetry data. This recursive simulation and testing process on ground is a tedious one and always based on few assumptions on the flight trajectory conditions and the associated errors in the models. This problem is addressed in this work by an alternative efficient method of the adaptive tuning of the auto-pilot parameters in real time during the flight of the vehicle. This requires to perform on-board analysis of the flight vehicle vibration signals in real-time and utilise the data to dynamically tune the autopilot notch filter parameters during the flight, instead of using the pre-stored values in the look-up table.

Mathematical models are developed for different techniques available in the literature to process random vibration signals sensed during the trajectory of the flight vehicle. These models are implemented and tested in Hardware-In-Loop-Simulation setup to verify performance with real-time and post-flight data. The performance comparison is carried out to identify suitable model for on-board application. It is observed from the analysis of results that Welch Method with either Hann or Blackman-Harris Window along with 25% overlapping of data for better accuracy is best suited for the on-board analysis of flight vehicle vibration signals. A memory-based architecture is proposed to implement the selected Welch method based mathematical model. This proposed architecture is designed for hardware realisation of different steps viz. Windowing, Overlapping, Domain conversion, PSD Computation and Identification of Frequency Maxima required for the random vibration signal analysis. The proposed architecture is modelled using Verilog HDL, simulated and synthesised using Xilinx Design Vivado tools targeting Kintex-7 series FPGA device. The synthesized netlist is validated by performing the Post synthesis functional and timing simulations. The proposed architecture is implemented on Kintex-7 Prototype board and the functionality is verified with stimuli and real time doppler signal of known frequency. The performance of this proposed implementation with different noise floor levels is also verified. A dedicated hardware is designed using the modules developed employing the proposed architecture along with the associated pre- and post-processing electronics. The hardware is tested for the functionality and performance with field inputs from the accelerometers mounted on the test objects. It is observed from the field test results that the spectrum signatures are closely matching with the the reference controller output, except for a small variation in the amplitude to the tune of 0.00102 g^2/Hz even when two dominant frequencies are at close proximity. The maximum difference between

the reference and estimated dominant frequency is 4.9 Hz in one measurement amounting to 1.9% of reference value and 0.24% in full scale range of 2000 Hz. For all other test cases, the accuracy of estimation over the full-scale range is less than 0.2% and over the reference value is less than 2%, whereas the resolution (ΔF) of 1024-point FFT over the range is 1.93 Hz. These values are well within the acceptable limit of the mission design requirement. When the low frequency modal parameters are processed, the accuracy of the measurement and the closeness to the correct structural frequency will improve further.

The conventional auto-pilot algorithm presently used is with wide-band/cascaded notch filter with definite value. As the flight vehicles are getting more and more optimized, there is a greater need for better control margins. To meet the required stability margins for high manoeuvrability vehicles, dynamically tuned Notch Filter is proposed in this work to avoid control-structure interaction. The improvement in the Notch filter characteristics and the control margins are compared among the conventional method and the proposed scheme. Formidable improvements in the control margins are observed using the proposed scheme.

Keywords : Modal Analysis, Frequency Domain, FFT, Windowing, Overlapping, Power Spectral Density, HILS, Verilog HDL, FPGA,Auto-pilot, Notch Filter

Contents

List of Figures

List of Tables

List of Abbreviations

VHDL	Very High Speed Integrated Circuit Hardware Description Language
RTL	Register Transfer Level
OBC	Onboard Computer
INS	Inertial Navigation System
FCS	Flight Control System
FPGA	Field Programmable Gate Array
SoC	System on Chip
FFT	Fast Fourier Transform
PSD	Power Spectral density
MUSIC	Multiple Signal Classification
ICP	Integrated Circuit Piezoelectric
ADC	Analog to Digital Converter
BRAM	Block Random Access Memory
ARMA	Auto Regressive Moving Average
VSPU	Vibration Signal Processing Unit

Chapter 1

Introduction

All physical entities have mass and elasticity causing them to vibrate when external forces are applied to the system for excitation, termed as forced vibration. An object can also vibrate under the action of inherent forces in the system in the absence of any external force, known to be free vibration. Vibrations of these objects are generally classified as linear or non-linear. One or more frequencies of oscillation under free vibration are known as natural frequency of the object. When the frequency of the external excitation coincides with the natural frequency of the object, then the resonance condition occurs resulting in damage to the structure because of oscillations with large magnitude. The vibration characteristics of the structure such as natural frequency and mode shapes can be determined by Modal Analysis. Dominant frequencies and associated energy content along with overall RMS value can be analysed using spectral analysis of vibration signals. Most of the physical signals vary with time and are random in nature and vibration experienced by the sub-systems and structure in a launch vehicle is no way different.

1.1 Flight Vehicles

A launch vehicle or a flight vehicle is an article propelled by combustion of propellants in a pressure chamber. The plume expelled out through nozzle propels the article in the forward direction following Newton's law. Generally, a launch vehicle consists of the following major sub-systems, namely,

- Motor with Solid or Liquid propellent
- Nozzle, either fixed or flex
- Control System
- Avionic Sub-systems like On-Board Computer, Inertial Navigation system, Telemetry systems etc.
- Payload

For a given initial velocity, using kinematic equations of motion, it is very easy to calculate the trajectory and the range of projectile. A launch vehicle or missile is designed for delivering a payload or a warhead, thousands of kilometers away with a small circular error probability. But the parameters like distance, exit velocity, launch angle or acceleration due to gravity may not be very accurate. Similarly, atmospheric drag, effect of wind etc also may not be accurately known. Feedback control and guidance systems are designed to achieve this task instead of depending on unrealizable accuracies of various parameters to meet the objectives [1]. The system performance depends heavily on accurate sensors and actuators. In present day scenario, there is an ever-growing demand for launch vehicles in order to place the desired subsystem at the desired location by minimising the cost function [2], which includes the cost of the mission article, effort on final error minimisation, operational cost, etc. During the trajectory of the flight vehicle, the navigation system provides current position of the missile. The guidance computer finds out error between the actual position and the desired instantaneous position of the vehicle to generate control system commands for correcting position as shown in Fig.1.1.

Control sensors sense the current missile orientation and On-Board Computer generates demands on Actuation System to generate required forces and moments acting on the missile, so that the missile position is corrected as per guidance requirement. Fig.1.2 shows general sketch of three feedback control loops. Though the final mission objective of launch vehicle or missile depends on accuracy of guidance loop, its performance depends on accuracy and speed of the response of autopilot loop which in turn depends on accuracy and speed of the response of actuator loop. Hence, while designing, say, an autopilot loop, the designer needs to meet the requirements of guidance loop and also lay down specifications on actuator loop. Response of a feedback control system is decided by poles and zeros of closed loop transfer function. When a large number of poles and

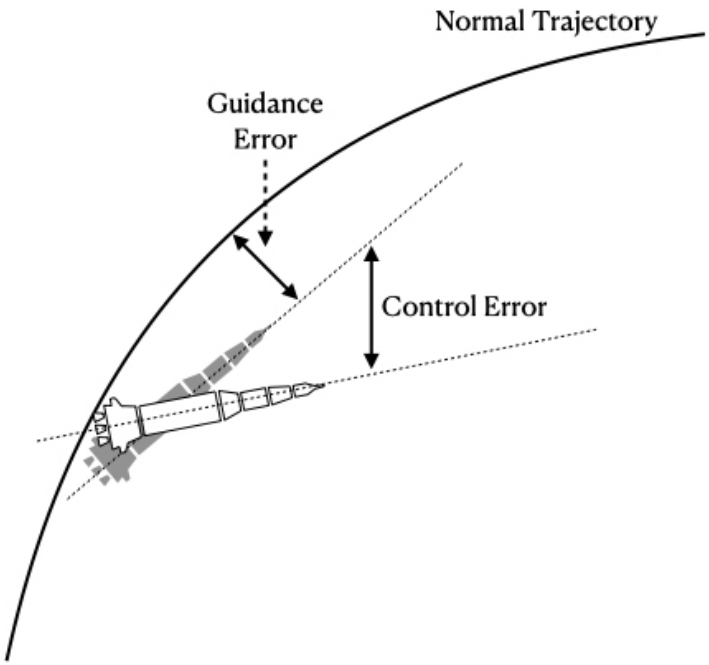

Figure 1.1 Missile Orientation and Trajectory

zeros are present, it is difficult to quickly assess performance of the system. Hence, for preliminary design one generally attempts to find an equivalent second or third order system for which system response characteristics can be quickly assessed. This helps in fixing various design parameters. Flight vehicle characteristics, flexibility of structure and its contribution to stability of flight vehicle and design of auto-pilot to cater stability margin requirements and Notch filter coefficients are one of the important areas of study and analysis.

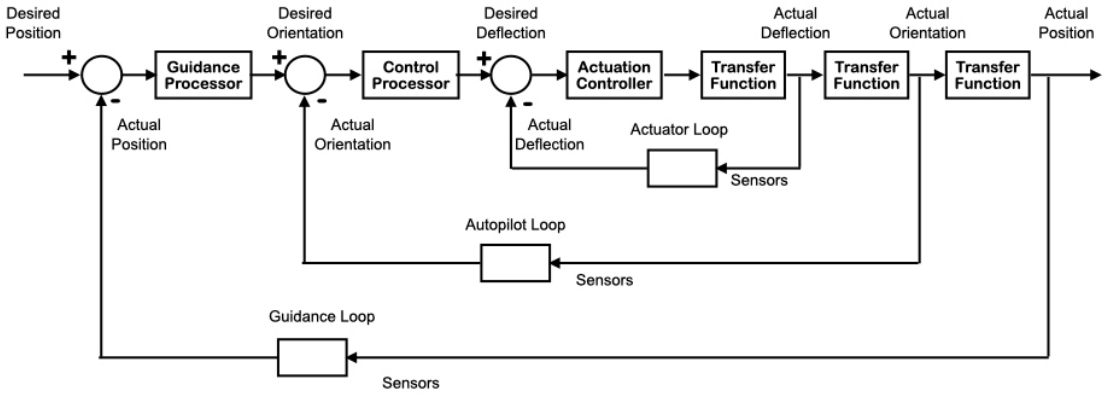

Figure 1.2 Guidance, Control and Actuator Loops

1.1.1 Flexibility in Flight Structures

Modal vibration data describes the dynamic characteristics of any flight system. A natural mode of vibration is excited due to aerodynamic disturbances caused during flight. These data include modal mass, flexural stiffness and structural damping associated with the modes. Knowledge of these data helps us to understand the dynamic behaviour of flight vehicle due to external excitement. It is also essential in determining inflight loads for spin stabilization and control system coupling which will save the flight vehicle from catastrophic failure [3]. Fig 1.3 shows full scale model of a flight vehicle with length of different sections and bending characteristics with first and second mode.

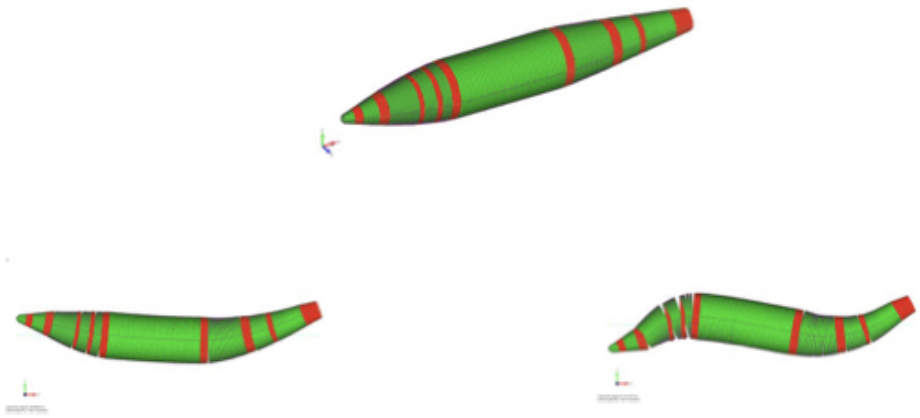

Figure 1.3 Full Scale Model and Mode Shapes

The modal frequencies of flight vehicle varies during the course of trajectory due to change in Mass, CG Properties, variations in material characteristics due to rise in temperature and other factors during flight. As it is impractical to evaluate these effects analytically [4] ,the problem had been treated empirically and standards are derived. Design of flight control system is carried out using the following steps [5]:

Step 1. Control system gain schedule design assuming vehicle as a rigid body

Step 2. Checking stability margins after incorporating structural flexibility.

Step 3. Design suitable filters to ensure stability of structural modes.

Step 4. Perturb vehicle data within expected bounds and check for instability.

The results of a combined experimental and numerical investigation of the first two natural frequencies and mode shapes of airborne system to program the flight in such a way that the forcing functions do not excite the fundamental modes of the structure and hence, stable systems may be achieved. The control-structure interaction generally enters the control loop through sensor and actuator mountings. In spite of all the detailed studies, there bound to be lacunae in the results due to errors in modelling, simulation assumptions and precisely not simulating the actual flight conditions on ground due to practical limitations. On few occasions, the actual flights show unstable bending mode, leading to diverging oscillations and structural failure. The Ground Resonance Test of the whole flight vehicle structure plays very important role in determining the Mode Shapes, Natural Frequency and the Damping Factor of the Structure. The test structure is excited by performing sine sweep at a desired band and natural frequencies are found out. Then the structure is excited at two of its natural frequencies to find out the corresponding mode shapes and damping at each frequency [6, 7]. The control system design needs to ensure that this aero-structure-control interaction is stable and does not lead to diverging oscillation. Hence, the stability analysis of flexible modes is an extremely important part of the flight control systems design. This is repetitive process after each flight trial before a final configuration and the auto-pilot filter coefficients are arrived and verified.

1.1.2 Structural Analysis

Conventionally, the Modal and Spectral behaviour of flight structures are modelled for simulation and structural analysis is carried out. Subsequently ground tests are carried out, keeping ground test setup close to flight conditions. The natural frequencies of the structure and dominant frequencies under dynamic conditions are evaluated in ground tests. The damping factor derived from hammer test and modal frequencies at different flight conditions are stored in lookup table of on-board computer for use in autopilot algorithm during flight. The results of ground resonance test and random vibration tests are compared with simulation results to fine tune either the model or the test conditions or both in many occasions. This is a recursive process as explained in Fig 1.4 till satisfactory results without major uncertainties are obtained. Also, these simulation and ground tests data are compared with flight telemetered data for further fine tuning of model.

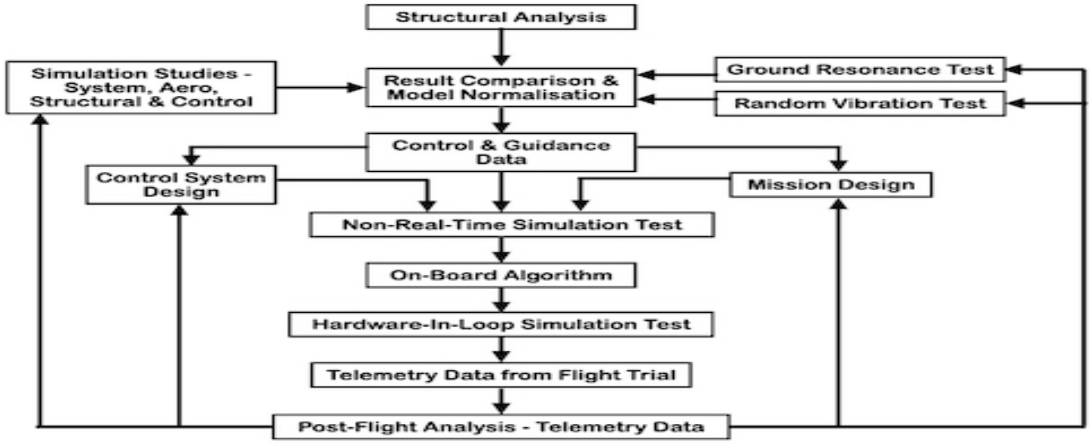

Figure 1.4 Simulation and Analysis Flow Diagram

It is observed through the study of existing methods that the recursive simulation and testing process on ground is a tedious one and always based on few assumptions on flight trajectory conditions and associated errors in mathematical models. Hence, an accurate knowledge of natural frequencies, including experimental verification, is desirable in vehicles with spin stabilisation programs. Modal information is also essential for proper positioning of guidance sensing devices and for investigating stability characteristics of vehicles with closed loop control systems. The orthogonal properties of mode shapes make them desirable functions for use in series solutions involving generalised coordinates, widely known as "Modal Form Solutions". The transient mass and structural characteristics of a typical multistage flight vehicle require that natural mode evaluations be made, at least for ignition and burnout times of each stage of flight and frequently at other points of investigation, such as Speed of Mach1, maximum dynamic pressure, and minimum stability. These stringent requirements involve substantial engineering effort and thereby justify the development and organisation of techniques for calculating modal data. They also emphasise importance of performing experimental investigations to ascertain suitability of the methods used.

1.2 Motivation

The Control-Structure interaction plays a crucial role in stability of vehicle. Mode shapes and natural frequencies of flight vehicle structure is a critical parameter in the

design of auto-pilot algorithm. Analysing flight vehicle structure for flexibility and dynamic analysis of flight airframe sections provides input for design of auto-pilot and flight structure integration. But the present method of conducting Ground Resonance Test, Airframe Random Vibration Tests and designing algorithms and procedures is an iterative process. This demands more time and resources, both for analysis and subsequent ground tests to validate design assumptions. But, ultimately adequacy of design is validated and confirmed during post flight analysis of trial data. Measurement chain inaccuracies and Simulation Models Errors are to be accounted for during analysis. The recursive simulation and testing process on ground is a tedious one and always based on few assumptions on flight trajectory conditions and associated errors in mathematical models. The best method is adaptive tuning of auto-pilot parameters in real time during flight of the vehicle. This calls for spatially separated sensors along structure and associated signal processing electronics which are immune to failures and has relatively fast update rate. The flight vehicles for both space and defence applications are getting highly optimized to suit operational requirements. Hence, there is a need to perform on-board analysis of vibration signals, to have better controllability with in-flight updation of control and auto-pilot filter coefficients. Of late, flight vehicles depending on type and purpose, have become highly optimised with respect to its Length/Width Ratio and manoeuvrability of vehicle calls for High Bandwidth control systems. Also, technology has advanced in areas of high precision sensors and processing electronics. To improve Stability margins and to design better control system overcoming unknown uncertainties, one option available is to measure modal frequency in real-time during the flight, which are functions of Vehicle Parameters and other trajectory conditions during flight and tune the Auto-pilot algorithms accordingly. It is observed from available literatures and review reports that a better method to be evolved and studied for circumventing the problems in the design of Control system. This method so-far had not been attempted in Missile applications, though a few measurements had been carried out for Aircrafts with rotary equipment. This method of In-flight identification of Modal frequencies becomes mission critical, and requires to update the Autopilot loop which has to be precise and fault-free. At the same time, it is observed that the same will improve stability and controllability of vehicle during flight, which needs to be addressed.

1.3 Research Objectives

The objective of this research is to design and develop an architecture for on-board frequency domain analysis of flight vehicle random vibration signals and body rates sensed by Inertial sensors so as to dynamically tune Notch filter to avoid control-structure interaction and thereby improving Stability Margins.

- The vibration data and body rates sensed by Inertial sensors, provide a good insight of structural behaviour of flight vehicle. A suitable model to be developed to capture and analyse these signals, which suits real time requirements of control system in the flight vehicle. Hence, the mathematical models are developed for various existing methods and frequency domain analysis of Random vibration signals is performed. These results are verified with Hardware-In-Loop-Simulation setup with known and random signals.

- An architecture is proposed for the hardware realization of validated mathematical model and HDl model are to be developed. These models have to be simulated and synthesized using Xilinx Design Vivado tools in sync with existing missile avionics data structure. A dedicated hardware ported with proposed architecture has to perform functions and yield expected results. The performance is defined by accuracy of the output. Trade-off to be made between execution time and output sensitivity, depending upon acceptable limits of real-time processing in Flight Vehicle applications.

- The proposed architecture is to be validated with real-time Vibration sensor inputs using Hardware-In-Loop simulation and Testing technique. The identified dominant Modal frequency is used to dynamically tune Notch filter to filter out body rates generated due to Natural Mode shape of Flight Vehicle during its trajectory motion. This will avoid control-structure interaction and hence, control margins needed can be maintained and the vehicle stability is improved. The Attenuation characteristics and the Phase-lag of the Notch Filter with the proposed scheme is computed. The ultimate effectiveness of proposed scheme is verified by comparing with conventional filter characteristics plots and the improvement in the control margins are also verified.

The contributions of the thesis are summarised as follows:

- **Mathematical Models, Verification and Validation in Hardware-In-Loop-Simulation Test Bench**

  Mathematical models are developed for chosen methods among prevailing frequently used methods, which are suitable for On-board applications of Flight vehicles. The output of mathematical models is verified and validated in Hardware-in-loop-Simulation test setup. The contributions of this work are briefly described as

  - **T**he Random vibration signals can be analysed either in Time-domain or frequency-domain depending on application. The Models developed for Periodogram, Welch Method and MUSIC Algorithm are simulated, verified with known and random inputs. The models are validated in HILS Platform for closeness with respect to standard reference analyser outputs. The Identified Dominant Frequencies are compared with dominant frequencies as recorded from Dewetron Reference Controller for both Summative sinusoids with additive Gaussian Noise and Vibration Sensor Response Data. Conclusions are drawn from comparative results to proceed with the hardware realization of the proposed model. The accuracy of models and Execution time are presented in this study. Also, the models are verified by analysing post-flight data and validating with predictions. The results of Post flight data analysis are also presented.

- **Proposed Architecture, Hardware Implementation and Field Trials**

  The validated mathematical models are converted to Architecture for implementation. The architecture is designed and modelled in HDL. The contributions of this work are briefly described as

  - **I**n this work, a memory-based architecture is proposed for the hardware realization of the proposed mathematical model. This architecture is designed to realize the Windowing Function, Frequency Domain Conversion, Computation of instantaneous Power Spectral Density spectrum or averaged-out spectrum based on overlapping criteria and identification of Frequency Maxima in hardware. All the modules of the proposed architecture are modelled using Verilog

HDL and functional simulations are performed using Xilinx Design Vivado tools. The Verilog netlist is synthesized targeting Kintex 7 series FPGA device and post synthesis functional and timing simulations are performed to validate the model. The proposed architecture is ported on to hardware with associated pre and post processing electronics for input and extraction of output. The output is compared with expected results of mathematical model for verification and with reference controller output for validation. The sensitivity of architecture is tested by varying the floor level of Input signal noise. The hardware is also tested for multiple Random vibration response signals and results are presented to prove the performance.

- **Proposed Scheme for Dynamic Tuning of Auto-pilot Notch Filter**
  To meet required stability margins for high manoeuvrability vehicles, dynamically tuned Notch Filter is proposed to avoid control-structure interaction. The filter coefficients are to be tuned dynamically based on vehicle attitude and structural behaviour. The contributions towards this are briefly described as

  – **I**n this work, the criticality of Notch Filter in the design of Auto-pilot algorithm and necessity for real-time update of Filter coefficients for effective control and better stability is presented. A post flight body rates as measured by gyros and variations in identified modal frequency over a period of time are analysed. The general notch filter characteristics presently available and improvement in performance with dynamic tuning is compared and results are presented. The results depict improvement in Attenuation characteristics of Notch Filter when phase lag is maintained and improvement in phase lag when attenuation levels are maintained. The achieved stability margins are presented.

1.4 Thesis Organization

The rest of the thesis is structured as follows:

Chapter 2: Identification of Dominant Frequency in Flight Vehicles presents the detailed study of relevant technical literatures on methods of structural and vibration analysis, Instantaneous frequency identification and architectures for Frequency Domain transformations and analysis.

Chapter 3: Random Vibration Signal Analysis presents the different Mathematical Modelling for the Frequency domain analysis of Random vibration signals and validation of models in Hardware-In-Loop-Simulation setup with known and random signals. The validation results of the models with Real-time data and Post-Flight Data are also presented.

Chapter 4 Architecture for Frequency Domain Analysis presents the proposed architecture developed to perform Windowing, Domain Conversion, Overlapping, PSD Computation and identification of Dominant Frequency component required for the analysis of flight vibration signals. The proposed architecture is modelled using Verilog HDL, simulated and synthesised using Xilinx Design Vivado tools targeting Kintex-7 series device. The synthesised netlist is validated by performing the Post synthesis functional and timing simulations. The proposed architecture is implemented on Kintex-7 Prototype board and the functionality is verified with stimuli and real time doppler signal of known frequency. The performance of the proposed implementation with different noise floor levels is also verified and results are presented.

Chapter 5: System Design and Field Verification presents the functional diagram of the system design using the modules developed employing the proposed architecture along with the associated pre- and post-processing electronics. The test setup for the field testing of the system and comparison of analysis results with standard reference controller are presented. The performance measurement parameters and the acceptance criteria are also explained.

Chapter 6: Auto-Pilot and Dynamic Tuning of Notch Filter presents the Conventional Method of Auto pilot to handle the flexibility issues and the Filter charac-

teristics, along with the updated model based on the proposed method. The comparison of the Filter performance parameters computed employing the conventional and updated model are presented. The effect of deploying updated model designed using the proposed architecture on the control margins of the vehicle is presented.

Chapter 7 Draws conclusions from the earlier chapters and concludes the thesis with future scope of work.

1.5 Conclusions

In this chapter, the basics of the flight vehicle, Vibrations encountered and the associated Structural flexibility issues of airborne vehicle is described. A brief overview of the entire research work along with the motivation behind this research and its objectives are presented. The next chapter presents the review of the technical literatures related to the present work.

Chapter 2

Identification of Dominant Frequency in Flight Vehicles

Dynamic properties of structures such as natural frequency and mode shapes can be determined by Modal Analysis. Dominant frequencies and associated energy content along with overall RMS value can be analysed using spectral analysis of vibration signals. These structural analysis of the flight structures during the configuration provides very crucial and critical inputs for mission and control system design.

2.1 Parameter Estimation

Most of the physical signals vary with time and are random in nature. These random signals are no-periodic and non-deterministic. These signals cannot be treated in the same way as deterministic signals and analysed by assuming a periodicity of infinite time as these signals do not obey the Dirichlet condition [8]. This frequency variation with time cannot be obtained using Fourier transform as this transform simply expands a signal as a linear combination of single frequencies that exist over all time [9]. A single component time variant signal x_t is

$$x_t = \sum_{k=1}^{\infty} (\beta cos(2_k + \varphi_k) + \in_t \tag{2.1}$$

where β is the amplitude, f is the frequency, φ is the phase and $\in_t$ is zero-mean random noise. Instantaneous Frequency, i.e. frequency at a particular time can be described as:

$$I_n(f) = n^{-1} \left| \sum x_t e^{-2\pi f t} \right|^2 \tag{2.2}$$

Information about propagation in an unknown environment may be derived from instantaneous frequency estimation of signal. Instantaneous frequency estimation is highly useful in applications such as underwater acoustics, radar, active vibration control etc. There are three main parameter estimation problems which involve frequency estimation [10]:

- Single tone frequency estimation: Signal is a single, constant-frequency sinusoid. This is the simplest frequency estimation problem.
- Multi-harmonic frequency estimation: Signal composes sum of harmonically related sinusoids. This case is important because rotational or periodic phenomena rarely generate sinusoidal waveforms.
- Multi-tone frequency estimation: There are several tones of unrelated frequencies present.

In Statistical Signal Processing and Time series Analysis, primary focus is on estimating sinusoidal signal frequency when it is coupled with environmental noises. For a random process of length 'k' and time series of $x_1, \ldots, x_k$,

$$x_t = \sum_{k=1}^{\infty} (\beta cos(2_k + \varphi_k) + \in_t \tag{2.3}$$

accurate instantaneous frequency estimates $\mathrm{I}_n(\mathrm{f})$ can be obtained by maximising the Periodogram as a continuous function of 'f' as

$$I_n(f) = n^{-1} \left| \sum x_t e^{-2\pi f t} \right|^2 \tag{2.4}$$

The asymptotic standard error of this estimator can be expressed as $\mathrm{O}\left(n^{-3/2}\right)$.

2.2 Vibration in Flight Vehicles

Flight vehicles used for placing payload on to the targeted trajectory encounter propulsion forces, aerodynamic forces, acoustic and shock loads during the launch of spacecraft. These forces strongly interact with flight structures and cause mechanical vibrations throughout the length of flight vehicle and also at interface with payload. When steady state accelerations are subtracted, these mechanical vibrations are generally categorised [11] as

- Sinusoidal vibrations 5-100 Hz
- Random vibrations 20 - 2000 Hz
- Shock loads, accelerations 100 - 5000 Hz

The shock loads have very short time periods in comparison to time cycles associated with lowest and lower natural frequencies of launch vehicle and are generally represented by Shock Response Spectrum. Acoustic excitation of launch vehicle causes mechanical vibrations of random nature on top of random mechanical vibrations. Many of the failures of satellites/payloads in initial days are due to vibro-acoustics during vehicle launch phase [12]. The estimation of dynamics response in launch vehicle, primary structure, secondary structures, instruments, and equipment etc. is called dynamic analysis. MIL-STD-810G emphasizes tailoring of flight components' design, test limits of conditions and test methods. The Environmental Engineering Guidelines, the laboratory test methods and the Climatic region guidance are envisaged in MIL-STD-810G and applicable test method IV for Missile and Launch vehicles defines the details of spectrum. Generally, flattened spectrum is used for vibration analysis of air-borne systems, with provision for fine-tuning the same using shaped spectrum based on flight requirements and flight data. Besides dynamic responses, modal properties of complete launch vehicle and subsystems, such as, natural frequencies, mode shapes, effective masses are also of great interest when discussing vibration in launch vehicles.

The methods by which modal frequencies can be identified are broadly classified as,

- Frequency domain or Time domain

– Bayesian or Non-Bayesian

The known theoretical properties such as correlation function or spectral density of measured vibrations are used with Statistical estimators in Non-Bayesian methods. When number of modal frequencies in vibration data is large, time domain approach provides best results. However, limitation is that these estimates are within the range of analysis. This does not take into account residual effects of those frequencies lying outside the range. On the other hand, Frequency Domain approaches provide best results when frequency range of interest is limited and number of modes is relatively small. But Frequency resolution, Leakage and High modal densities are disadvantages of Frequency domain analysis. These problems, however, can be addressed by increasing the order of model.

2.3 Review of Frequency Identification Methods

The requirement of on-board analysis of modal and spectral vibration signals is gaining importance with high optimization of flight vehicles and need for better and on-board updating of control coefficients. Hence, providing a solution to analyse vibration signals on-board, generating Power Spectral Density and identifying dominant frequencies within the time bounds of data updates in flight vehicles are of paramount importance for better control and stability. In 1993, C. W. Acree, Jr. et al. [13] analysed flight flutter data using Fourier Transformation and computed spectral data using Chirp-Z transformation. The method is tested for initial test flight structure of XV-15 tilt rotor wing at baseline speed.

Hilbert Transform analysis in time-domain is proposed by M Feldman [14] to extract instantaneous undamped frequency and real non-linear elastic force characteristics. For Hilbert Transformation method, vibration signal should be a mono-component signal from a S-DoF system or from a multi-DoF system after special decomposition. Freevib method is used for non-linear frequency response function and Decomposition/Scaling Technique is used for force characteristic interpretation, which involves intense numerical calculations that may not be suitable for real-time on-board applications. Tadeusz Uhl et al. [15] proposed a method based on the linearisation of the relation between ARMA model parameter variance and the standard deviation of the modal parameters. The time

domain method is formulated for linear systems with time-varying parameters. A Morlet type wavelet is used in the proposed algorithm. Confidence intervals for all parameters were relatively small. M.Ruzzene, et al. [16] used the wavelet transform of system's free response for estimating the modal parameters. Juan R. Traperoa et al. [17] suggested the use of pure sinusoidal model in combination with the algebraic derivative method for parameter identification. This algebraic method is also capable of estimating more than two frequencies. However, its real-time implementation for actual signals could be computationally difficult. This method is generally suited for condition monitoring of mechanical moving structures. Jesús Ponce de León et al. [18] proposed the use of Complex Continuous Wavelet Transformation for fundamental frequency estimation. The method is applied for base frequency estimation of audio signals in single channel. Maximum likelihood Estimator for real valued signals is proposed by Nielsen, Jesper Kjær et al. [19] as a viable alternative to the autocorrelation-based methods for audio signal processing. Different techniques proposed in the literature for identification of instantaneous frequency of the signals are presented [20, 21], which are generally applicable for off-line analysis of vibration signals.

2.4 Frequency Domain Transformation

Most commonly used domain transformation techniques are Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Dual Tree Complex Wavelet Transform (DTCWT) etc. [22, 23]. The Wavelet transformations DWT and DTCWT are used for time-frequency analysis of random signals. DCT is widely used in signal processing and image compression. Whereas, for instantaneous frequency identification and associated spectral density, the DFT is the proven choice. Fast Fourier Transform (FFT) is one of the most efficient algorithms widely used in the field of modern digital signal processing to compute Discrete Fourier Transform (DFT) as it requires minimum number of resources while giving maximum efficiency in terms of the computation time and accuracy. The Discrete Fourier Transform (DFT) for an N-point input signal x(n) is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \tag{2.5}$$

$$W_N^{kn} = e^{-j2\pi nk/N} \tag{2.6}$$

where, k=0, 1,. . . ..N-1 and X(k) represents the frequency domain values obtained from the Fourier transform on x(n).

2.5 Algorithms for FFT

Different algorithms are proposed for the computation of FFT over the years and some of the widely used algorithms are

- Cooley-Tukey Algorithm
- Prime Factor Algorithm
- Winograd Algorithm
- Rader Brenner Algorithm
- Brunn's Algorithm

Cooley-Tukey algorithm [24] is the most commonly used and is the simplest mapping suitable for any number N. The DFT of composite size N = N_1N_2 is recursively broken down into smaller DFTs of sizes N_1 and N_2. Thus, it uses the divide and conquer method effectively along with O(N) multiplications by complex roots of unity traditionally called twiddle factors. Prime Factor Algorithm, also called as the Good-Thomas algorithm [25, 26] is not suitable for all numbers of 'N' because of the restriction that all the factors of N should be co-prime. Hence, PFA is used as the special FFT algorithm for numbers with co-prime factors.

Winograd Algorithm [27] factorises $Z^N - 1$ into various polynomials having coefficients of 1, 0 or 1, and thus requires few multiplications for its operation. Winograd showed that DFT can be computed with only irrational multiplications. which reduces the number of multiplications considerably at the cost of hardware. Since modern hardware architecture consists of multiplier blocks, number of multiplication operations in an algorithm is not a deterrent for optimisation. Subsequently, Richard Tolimieri [28]

modified the Winograd algorithm by designing multiplicative FFT algorithms with highly structured data flow. The number of real additions and real multiplications required by these algorithms are almost same that of Winograd algorithm. However, they have a better structure, which simplified their implementation. Two algorithms were suggested for N = p^2. One suitable for parallel computing and the other for a better operational count, suited for conventional serial machines.

In Rader-Brenner Algorithm [29], the complex multiplications are replaced by multiplication of complex number by a purely real or imaginary number. It is realised by computing an N-point DFT with N=2t. On the other side, the Bruun's algorithm is a Fast Fourier transform algorithm based on an unusual recursive polynomial-factorisation approach. It is proposed for powers of two by G. Bruun [30] and its operation involves only real co-efficient until the last computation stage. Table 2.1 presents the techniques employed for the computation of FFT by above mentioned algorithms.

Table 2.1 FFT Algorithms

Algorithm	**FFT Computation Technique**
Prime Factor Algorithm	Re-expresses the DFT but only for the case where N1 and N2 are relatively prime
Cooley-Turkey Algorithm	Divide N-pt DFT into M,N/M pt DFTs
Wenograd Algorithm	Factorizes $Z^N - 1$ into various polynomials
Rader-Brenner Algorithm	Computes N-pt DFT with N=2t
Brunn's Algorithm	Computes DFT of real co-efficient

2.6 Review of FFT Architectures

There are numerous architectures available in the literature [31–35] to realise FFT algorithms. The choice of architecture is dependent on application and resource availability. They can be broadly classified as shown in Fig. 2.1

- Pipelined Architecture

- Parallel Architecture
- Memory based Architecture

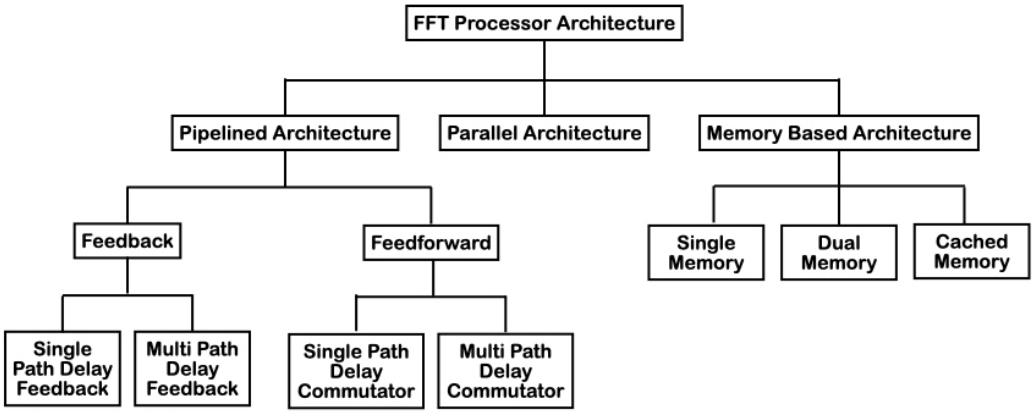

Figure 2.1 FFT Processor Architecture

FFT can be computed using Decimation-in-Time (DIT) or Decimation-in-Frequency (DIF) algorithms. In the case of DIT algorithms, the input sequence is stored in bit-reversed order and the output sequence will be in normal order. But in the case of DIF algorithms, the input sequences are in natural order and the output sequences are obtained in bit-reversed order.

2.6.1 Pipelined Architecture

Pipelined architecture is the classical method for the implementation of the FFT algorithm for the real-time applications. Pipelined architecture is proposed in the 1970s by L. Rabiner and B. Gold [31] for real time applications with a continuous input data. This provides a high throughput rate as compared to the other architectures. These structures use one butterfly unit between two commutators at each stage. The butterflies compute the data addition and subtraction, whereas the commutators are used to rearrange the data from one butterfly unit for use by the next stage. The general structure of the pipelined architecture is shown in Fig.2.2.

Pipelined architecture is broadly of two types, Feedback and Feed Forward. The input to the processor in both cases is sequential in real time. As Feed-forward archi-

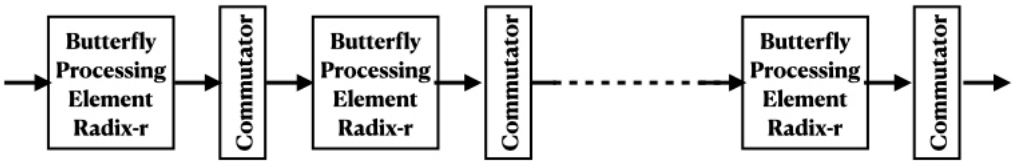

Figure 2.2 Generalised Pipelined Architecture

tectures results in more latency, Feedback architectures are preferred for real-time signal processing. In Single Path Delay Feedback architecture for N= 2i, a total of 'i' dynamic butterflies are to be used for a radix-2 algorithm. One input to the butterfly is the direct input and the other one is the delayed input through the feedback path. The output of one stage is subsequently passed on to the next stage. The last stage is to unscramble the output. The different architectures for the pipelined FFT processors are summarised below.

Radix-2 SDF:

Radix-2 Single-path Delay Feedback stores the butterfly output in the shift registers of the feedback path. At every stage, a single input data stream passes through the multiplier. This configuration uses the memory efficiently and requires (N-1) registers.

Radix-4 SDF:

Radix-4 Single-path Delay Feedback employs CORDIC iterations. This is radix-4 version of the radix-2SDF. However, the radix-4 butterfly is fairly complicated. It requires $\log_4$(N - 1) multipliers, $\log_4$(N) full radix-4 butterflies and storage of size (N-1).

Radix-2 MDC:

Radix-2 Multi-path Delay Commutator is notably simple and straightforward architecture for the implementation of radix-2 FFT algorithm. The input data stream is divided into two streams in parallel. Proper timing is maintained between the data elements those entering the butterfly processors. Control and delays are more important in this architecture. It requires $\log_2$(N - 2) multipliers, $\log_2$(N) radix-2 butterflies and (3N/2 – 2) registers (delay elements).

Radix-4 MDC:

Radix-4 Multi-path Delay Commutator is a radix-4 version of radix-2 Multi-path Delay commutator. It has a low utilisation factor of the resources, to about 25%, because of which it is used for few specific and special applications. It requires 3 $\log_4$(N) multipliers,

$\log_4$N full radix-4 butterflies and (5N/2-4) registers.

Radix-4 SDC:

Radix-4 Single-path Delay Commutator is relatively a complicated architecture using a modified radix-4 algorithm. It has a programmable 1/4 radix-4 butterfly units. This increases the utilisation factor of the multipliers in the butterflies. To reduce the memory requirements, it uses combined Delay-commutator. It requires (2N-2) memory and Log_4(N-1) multipliers.

The comparison of the hardware complexity of the above pipelined architectures is tabulated in Table 2.2. Pipelined architecture using the Cooley-Tukey algorithm was proposed

Table 2.2 Harware Complexity Comparison of Pipelined Architectures

	Radix-2 SDF	**Radix-4 SDF**	**Radix-2 MDC**	**Radix-4 MDC**	**Radix-4 SDC**
Registers	N-1	N-1	(3N/2)-1	(5N/2)-4	2(N-1)
Adders	2 $\text{Log}_2 N$	8 $\text{Log}_4 N$	2 $\text{Log}_2 N$	8 $\text{Log}_4 N$	3 $\text{Log}_4 N$
Multipliers	$\text{Log}_2(N-2)$	$\text{Log}_4(N-1)$	$\text{Log}_2(N-2)$	3 $\text{Log}_4 N$	$\text{Log}_4(N-1)$

by Groginsky and Works [33]. In this, two processors are used per FFT stage and the total storage requirement is N words. All the control inputs were generated by a binary counter to keep the control circuits simple and the first block of output data is received after the last sample in the input block of N is received. The biggest disadvantage of this architecture is the scrambled order in which the output data are received. Descrambling was possible with the introduction of additional memory blocks for re-arranging the output data to the required format. Subsequently, many modifications were suggested for the Pipelined FFT processors.

Shousheng He and Mats Torkelson [36] proposed the design and implementation of 1024 point pipelined FFT Processor with radix-2^2 algorithm with SDF architecture. Two types of butterflies, one for SDF and other one for both SDF and logic to implement the twindle factor multiplication are implemented in this proposal. The synchronisation control is made simple using the spatial regularity of radix-2^2 algorithm. T. Widhe et al. [37] had attempted to reduce the floor space of the Processing Element by using the radix-

8 architecture with Bit-serial implementation. This approach uses modular adder tree for the complex valued inputs and a dedicated ROM of size N/8 complex words to store the twiddle factor for each multiplier. Se Ho Park et al. [38] proposed an implementation method with sequential data processing for a single-chip 8192 complex point FFT. It is reported that around 55% of the chip area for the sequential processing was reduced by employing a DRAM like pipelined commutator architecture. The 16-point FFT is a basic building block of the entire FFT chip. The 8192-point FFT is designed by cascading six stages of radix-4 and one stage of radix-2. Yun-Nan Chang et al [39] presented a pipelined architecture based on radix-4 decimation-in-time algorithm. Digit serial arithmetic unit is the main processing element used in this architecture. Both the feedforward and feedback commutator schemes are combined to achieve maximum hardware utilisation and reduce the memory required. By exploiting the redundancy of the factors, the overall ROM size is reduced by a factor of 2. Bi and Jones [40] proposed modified Radix-4 architecture with four memory blocks, each storing N/4 complex data words, to receive the input data while another four blocks dump the received input data. C Chiu et al. [41] proposed details of a new low power Fast Fourier Transform (FFT) processor for use in digital television applications. This has been fabricated using a 0.6-µm CMOS technology and can perform a 64 point complex forward or inverse FFT on real-time video up to 18 Megasamples per second. Resource comparison of few of the pipelined architectures is as in Table 2.3. The evolution of radix-2^2 was a milestone in the design of pipelined FFT hardware architectures. Later, radix-2^2 was extended to radix-2^k. Radix-2^k was proposed for both single-path delay feedback (SDF) architectures and for feedforward ones, also called multi-path delay commutator (MDC). In feedforward architectures radix-2^k can be used for any number of parallel samples which is a power of two. Furthermore, both decimation in frequency (DIF) and decimation in time (DIT) decompositions can be used. The designs can achieve very high throughputs, which makes them suitable for the most demanding applications [42].

Shousheng He and Mats Torkelson [43] had derived a hardware-oriented radix-2^2 algorithm which has the multiplicative complexity of Radix-4, but has the same Radix-2 butterfly structure. Of the two types of butterflies it uses, one is identical to the Radix-2 SDF. The other one, also contains the multiplicative logic for the twiddle factor. R2^2SDF utilises Log_4(N-1) multipliers, $4Log_4$(N) adders and needs a memory of size (N-1). The

Table 2.3 Resource Requirements of Pipelined FFT Architectures

* - Bit Parallel. ** - Digit Serial. N - No. of FFT Points

Resources	**He and Torkelson [36]**	**Y N Chang and Parhi [39]**	**Bi and Jones [40]**	**C Chiu et al. [41]**
Commutator Scheme	Feedback	Mixed	Feedforward	Feedforward
Complex Adders	4 $\text{Log}_4 N$ *	8 $\text{Log}_4(N+1)$ **	8 $\text{Log}_4 N$ *	12 $\text{Log}_4 N$ **
Complex Multipliers	$\text{Log}_4 N - 1$ *	$3(\text{Log}_4 N - 1)$ **	$\text{Log}_4 N - 1$ *	$3(\text{Log}_4 N - 1)$ **
Data Memory	N	1.18N	2.75N	2.5N
Twindile Factor ROM	N	0.5N	N	N
Efficiency add mul	 50% 75%	 100% 100%	 50% 75%	 100% 100%

control complexity is simple in this architecture. Yun-Nan Chang [44] proposed a modified MDF FFT for single input data stream, by integrating the function of sequence converter and few data commutator, so as to reduce the total size of the data buffer. The last stage data commutator is integrated with the converter to provide the sequence conversion mechanism to provide a transformed output sequence with normal order.

2.6.2 Parallel Architecture

A parallel architecture is obtained by placing a Processing element for each of the butterfly operation which are interconnected to compute the FFT. If all the butterflies of one stage work in parallel, it is called as 'column parallel' architecture and this is feasible

only if the input samples are already stacked in the memory. Such parallel architectures are possible with DSP processors and multi-core processors which have a lot of readily available computing resources. The generalised parallel architecture and a radix-2 parallel architecture are shown in Fig. 2.3 and 2.4. Since the parallel architecture consumes huge area and power as compared to pipelined architecture, it is not an attractive option where the number of points 'N' is high. When the input data rate is higher than the computational speed of the processor, parallelism can be introduced to derive N-parallel architectures with SDF or SDC as the basic structure.

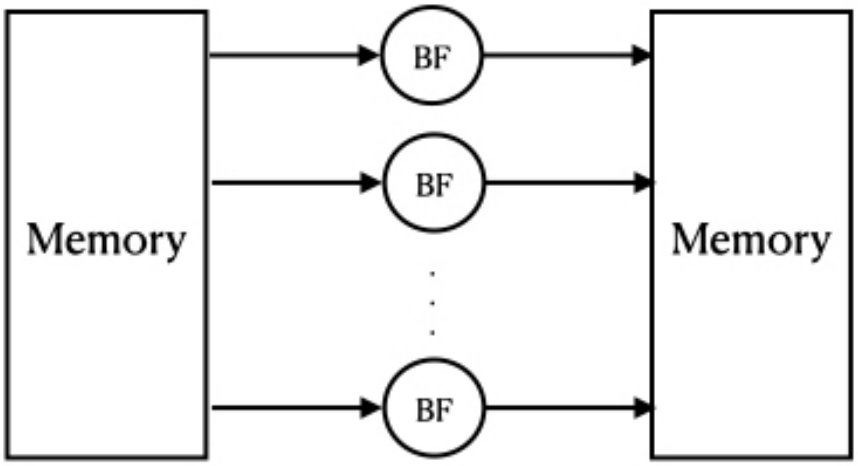

Figure 2.3 Generalised Parallel Architecture

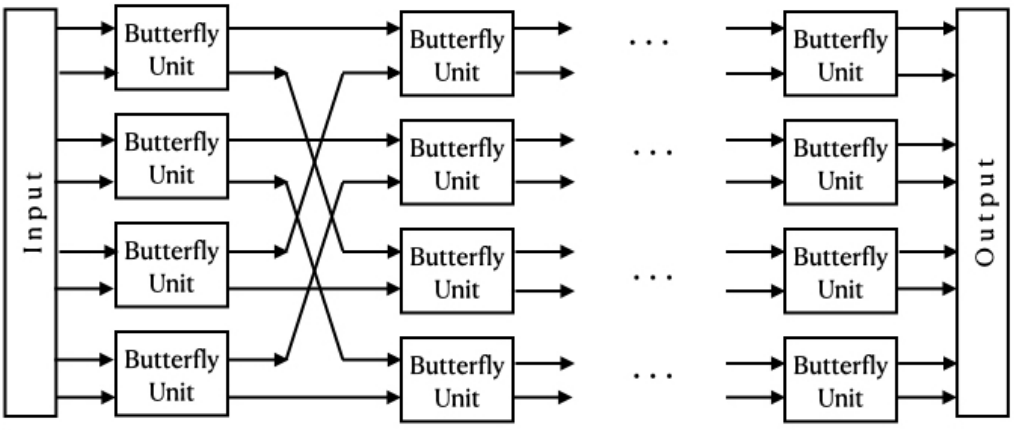

Figure 2.4 Radix-2 Parallel Architecture

Marshal E Pease [45] developed the modified version of the FFT and implemented in special purpose computer designed for this purpose. The computations were done in parallel for the successive pairs of data. The real and imaginary parts were also processed in parallel. Parallel shift of the data for the real and imaginary parts were carried out simultaneously using 'wired-in' shift network. The control system and the associated memory is too critical to store the common multipliers needed for the computation and the masking vectors presented for the selection of the multiplier sub-set. John E Welchel et al. [46] presented an FFT algorithm and implementation in a parallel processor. A recursive pipelined FFT processor along with a distributed Random-Access Memory and phase rotations are used for algorithm implementation and data shuffling. This architecture eliminates the need of the delay commutator switches. The Variable geometry algorithm is replaced by the Fixed Geometry processors as described by Pease [45] to improve throughput capability compared to the other FFT designs. The performance of these parallel architectures can be by improved by designing parallel-pipelined architecture to effectively use the advantages of both pipelined and parallel architectures.

2.6.3 Parallel-Pipelined Architectures

Mario Garrido et al. [42] proposed radix-2^k feed forward (MDC) FFT architecture to use for any number parallel samples which is a power of two. This architecture can be used for both Decimation-In-Time (DIT) and Decimation-In-Frequency (DIF) decomposition methods. Manohar Ayinala et al. [47] presented an approach to develop parallel pipelined architectures for the Fast Fourier transform using folding transformation and register minimisation techniques. For Complex valued Fourier transform, it takes advantage of under-utilised hardware in the serial architecture to derive parallel architectures without increasing the hardware complexity by a factor of parallelism. The architectures exploited redundancy in the computation of FFT samples to reduce the hardware complexity. These architectures are extended by adding multiple parallelism in power of 2 with the Radix 2^k architecture [48]. The hardware complexity comparison of the architectures for N Point FFT architecture with a parallelisation level of 4 proposed in the literature [42, 47, 48] is presented in Table 2.4.

Table 2.4 Hardware Complexity of Parallel-Pipelined Architectures for FFT

		Parallelisation Level	**Complex Multipliers**	**Real Adders**	**Delay Elements**
Radix-2	M Garrido [42]	4	$\text{Log}_2(N-3)$	$4\text{Log}_2 N$	N
	Syed [48]	4	$\text{Log}_2(N-3)$	$4\ \text{Log}_2(N-2)$	N-4
Radix-8	M Ayinala [47]	4	$2\text{Log}_8(N-1)$	$4\ \text{Log}_2(N-2)$	7N/4 - 4
	Syed [48]	4	$2\text{Log}_8(N-1)$	$4\ \text{Log}_2(N-2)$	N-4

Kenneth S. Stevens [49] proposed the parallel - pipelined methodology based on multirate signal processing techniques and asynchronous design style. This significantly reduced the power requirements as compared to the conventional design. This architecture is same as conventional Cooley-Tukey FFT formulation in multiplicative complexity. However, it proposed localised computations as opposed to global computing butterflies, thus resulting in implementation of parallelised architecture in a pipeline fashion.

The systolic phase rotation design and the distributed memory provides the flexibility for increasing the radix. However, being a high-performance design, power consumption and area are more critical in this architecture. With R4MDC architecture as a basic structure, a 4-parallel implementation, referred as a parallel-pipelined architecture, is used for a high throughput 64-point FFT for IEEE 802.11 applications by Wei Han et al. [50]. Power reduction can be attained with such architectures by reducing the frequency of operation for a given throughput. In 1984, H S Lee et al. [51] proposed the usage of two-dimensional systolic arrays designed using N processors. Similarly, another systolic array architecture proposed by Bridges et al. [52] needed (2N-1) processors. Subsequently WARP processors were developed for use in Systolic Array. Of the three variants of the WARP processors, WW-WARP (Wire Wrap WARP) is systolic array design with truly lock step sequencing of stages. The systolic architecture proposed by Kung & Webb [53] for the purpose of FFT computation required $\log_2(N)$ processors and N $\log_2(N)$ storage locations.

Valentin Boriakoff [34] proposed the implementation of one systolic linearly connected array with three inner-product step processors for each butterfly stage, making the operation of the entire ensemble a pipelined one. The clock rate is same for complete unit or ensemble. This architecture requires $3\log_2(N)$ number of processors and $5\log_2(N)+2N+2$ registers with Latency time $t(lat) = 2N+\log_2(N)+1$. The results report that $f(lat) = 2059$ time units and processor utilisation factor is 66.7% for 1024 point FFT. The major advantage is that there is no signal cross over as the communication is with the neighbour only and clock is the only common signal for the system. Though, Ja Ja & Owens [35] proposed the three unit scheme with shuffling memory, processing unit and control unit, the configuration translated into a system requiring 106 more memory locations. Also, the proposed scheme requires a few flip-flops and two or three logic gates for implementing the shuffle memory, making it complicated with respect to hardware requirements.

2.6.4 Memory Based Architecture

The memory-based architecture has a main processing core or processor with a computational memory for temporary storage. In single memory architectures, an arithmetic module is connected to a single memory. In the application of FFT computation, the butterfly input is read from the memory and processed output is written back to the same memory. In dual memory architecture, two memory blocks are connected to the arithmetic module separately. Data is read and written to different memory blocks. Radix-2 memory-based architecture is shown in Fig. 2.5. Cache memory Architecture is mainly used to increase the speed of the memory access, energy efficiency and for reducing the power consumption. This architecture is not widely preferred due to need of extra hardware and controller complexity. The nomenclature of the cached FFT is defined by following stages [32],

1. **Stage:** A stage is the part of an FFT where all N memory locations are read, processed by a butterfly, and written back.

2. **Super stage:** A super stage is the portion of the cached FFT algorithm, where all N-data word are loaded into a cache, processed, and written back to the main

memory, which can be regarded as a super stage.

3. **Group:** A group is the portion of a super stage where a block of data is read from the main memory into a cache, processed, and written back to the main memory.

4. **Pass:** A pass is the portion of a group where each word in the cache is read, processed with a butterfly, and written back to the cache once.

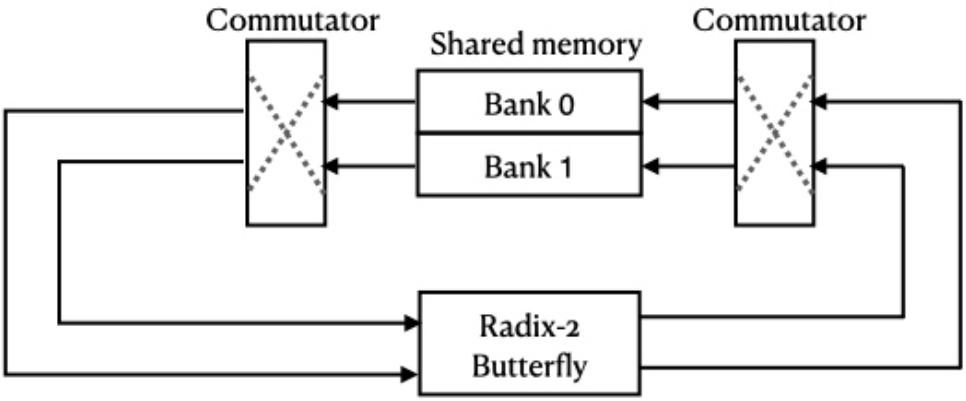

Figure 2.5 Radix-2 Memory based Architecture

To realise a variable length FFT Processor, Jen-Chih Kuo [54] adopted the cached memory architecture designed explicitly to operate on a processor with a hierarchical memory system. In the traditional FFT processing, the data are read and written back into the memory in every stage. As proposed, a super stage is used for processing more data and an intermediate cache is used in every super stage for fetching and storing the data. Jianming Wu [55] proposed a memory based architecture using two N-word memories for continuous flow FFT computation. It consists of a radix-4/2 butterfly unit, two main memories, a ROM for twiddle factor storage and a control unit. The control unit contains three functional components: data memory address generator, twiddle factor address generator and operation state controller. The butterfly calculation unit is responsible for processing of one radix-4 butterfly or two radix-2 butterflies. The attempt was to reduce the clock cycles required for the computation of FFT. Multiplierless architectures with memory element to compliment the multiplier functions [56–58] are proposed in the literatures.

2.7 FFT Co-processors

For many of the real-time processing requirements and applications, FFT functions are designed as a co-processor. This delivers a better resource utilisation in hardware implementation. Being a glue logic, the same is easy to handle and integrate with the main processor based on the application and need. The general performance metrics of a co-processor are

- Type of FPGA platform
- Data width
- Transformation length
- Type of algorithm
- Power consumption
- Resource utilisation and
- Max. clock frequency.

Performance Comparison of the FFT co-processor implementations available in the literature is shown in Table 2.5.

2.8 Analysis

With multiple architectures developed over the time and technological advances, implementing the FFT algorithm for the real-time applications like conditioning monitoring, structural analysis, audio mapping, SDR etc. are still a challenging task. The processing of the real-time analogue signals with windowing and post processing with overlapping algorithms adds to the processing overheads of the dedicated FFT processor. With increased number of bits in A-to-D conversion, the need of precise identification and estimation of instantaneous frequencies is the need of the hour. The throughput, cycle time, power and chip area are the most important performance metrics in the design of

Table 2.5 FFT Co-processors

S.No	FPGA Device	Data Width	Transform Length	FFT Algorithm	Power	Max Clk Freq. MHz
He H [59]	Altera Stratix IV	16 bits	1024 Points	NA	884 mW	275
Harikrishna [60]	Xilinx Virtex-E	Config.	Config.	Radix 2^2	NA	58
Derafshi [61]	Xilinx Virtex-4	16 bits	1024 Points	Radix 2^2	NA	100
Langemeyer [62]	Xilinx Virtex-5	Config.	Config.	Radix 2^3	NA	150
Altera IP Core (2015)	Altera Stratix IV	Config.	Config.	Radix 2	880 mW	375

the an FFT processor for use in real-time applications, especially for flight applications. A number of FFT designs have been presented in literature to reduce the chip-area consumed by multipliers. Co-ordinate rotation digital computer (CORDIC)-based designs [63, 64], distributed arithmetic (DA) based designs [65, 66] are to name a few. However, there is always an accuracy problem with designs based on co-ordinate rotation digital computer (CORDIC) since it takes several iterations to reach the desired result. On the other hand, DA based design is primarily suitable for computing larger length inner products. High radix implementation of the aforementioned techniques could be used for minimising the multiplications in FFTs, but increases the system complexity. Different approaches have also been presented to eliminate the complex multiplier completely by special constant coefficients using shifters and adders. This results in significant reduction in the number of operations, which helps to reduce the power consumption at the expense of increase in delay. To overcome the above mentioned issues, memory based implementation of multiplier would be an alternative approach. According to international technology roadmap

for semi-conductors (ITRS), memory will have dominating presence in the future System-on-Chip (SoC). It also provides high-throughput and reduced-latency implementation due to shorter memory access time than usual multiplication-time. With advancements in complementary metal oxide semiconductor (CMOS) technology scaling, the cost of memory has reduced making it faster and more power-efficient over its predecessors.

2.9 Conclusion

The literature survey presents an insight about designing a variable length FFT processor. It is found that most of the processors use mixed-radix architecture to realise variable N-Point FFT. The review of architectural aspects reveals that most of the researchers focussed on the variations in pipelined architecture or memory based architecture. Most of the FFT processors use a fixed point data format using 8 bits or 10 bits or 16 bits. SQNR is an important concern with small word lengths. For on-board applications of a flight vehicle, where the results of the FFT processor to be used in the closed loop, the architecture becomes further more critical as errors may lead to fatal failures of the mission. Hence, a combination of the possible architectures is to be designed after developing the mathematical model for the processing of the random vibration input signals in a flight vehicle. This mathematical model has to be simulated and tested with Hardware in Loop simulation (HILS). The details of the mathematical models developed for use in flight applications and the HILS results are presented in next chapter.

Chapter 3

Random Vibration Signal Analysis

The On-board analysis of the flight vehicle vibration signals is a very critical phenomenon, which needs a fail-safe and fully fault-tolerant system for implementation. Since, the frequency domain analysis method suits for the analysis the flight vehicle vibration signals to identify the dominant frequencies, robust Mathematical models for the transformation techniques and analysis are to be developed. These models are to be verified and validated with Hardware-in-loop simulation for analysing the performance of the model and to determine the feasibility for on-board applications.

3.1 On-Board Analysis of Flight Vibration Signals

The body rates are sensed by rate gyros in the Inertial Navigation System or by the strap-on gyro-systems. The modal and spectral vibrations of the flight structure are sensed by the accelerometers mounted along the vehicle body. These signals are transmitted to ground through telemetry system and the received signals on ground are analysed. Based on the analysis results, the auto-pilot algorithm is tuned for subsequent flight trials.

To circumvent the current iterative process of simulation and flight tests to fine tune the autopilot algorithm, the rates and vibration signals can be analysed on-board the flight vehicle employing the proposed signal processing flow shown in Fig.3.1 to compute the power spectral density and the dominant frequencies. These values can be used for real-time tuning of the Auto-pilot notch filters, which will effectively reduce or nullify the control-structure interaction for better controllability and provide better control margins.

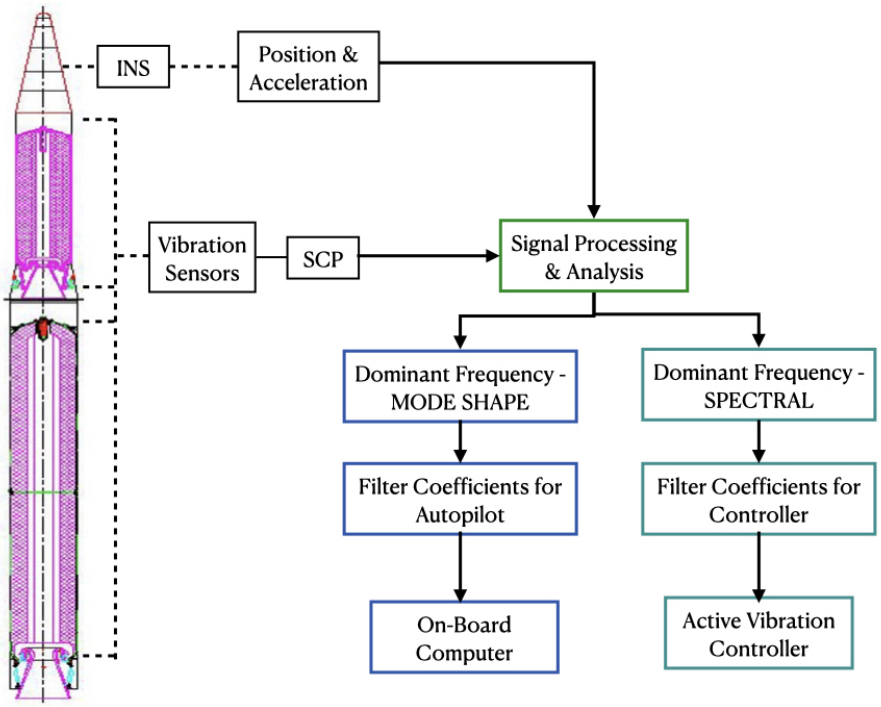

Figure 3.1 Proposed Signal Processing Flow

Since the auto-pilot coefficients are dynamic depending on the flight attitude and the prevailing aero-conditions, narrow-band and sharp filters can be designed which in turn will be more efficient compared to the broad band filter designed for the whole flight time to cover varying flight trajectory and attitude of the vehicle.

3.2 Analysis Methods

To design an architecture for processing the vibration signals suitable for on-board applications, initially the mathematical model is established. The mathematical model represents the process to be carried out and is simulated. In this, the pre-signal conditioning is not modelled and only the following process are modelled for simulation and verification.

- Necessary Window Function
- Storing and overlapping of the windowed input samples
- Frequency Domain Conversion

- Post processing for
  - calculation of Power Spectral Density
  - identifying the Amplitude maxima
  - corresponding dominant frequency

The objective of identification of the dominant frequency of the flight vehicle vibration signal in frequency domain is addressed by developing Mathematical Models using the following three methods,

- Periodogram using Fast Fourier Transformation
- Welch Method
- Correlogram using MUltiple SIgnal Classification(MUSIC) algorithm

Fast Fourier Transformation is a Periodogram method and Welch Method is an average of periodograms across time with rectangular window and formed from non-overlapping successive blocks of data. The MUSIC algorithm is a super-resolution approach for time-frequency analysis. Although the performance advantages of MUSIC are substantial, they are achieved at a cost of increase in computation time in searching over parameter space and storage requirements of array calibration data.

3.3 Model Simulation and Verification

The mathematical models developed are simulated with known sinusoids of frequencies 5 Hz, 50 Hz, 100 Hz, 500 Hz, 1000Hz, 1500 Hz and 2000Hz with an amplitude of 1 V. The summated signal added with random gaussian white noise is taken as the reference input for mathematical model simulation. The Summated sinusoidal signal with seven frequency components is added with Gaussian Noise as shown in Fig. 3.2. The simulation is carried out with 4096-point FFT in all the three mathematical models. Raw input data is analysed without any filtering or pre-processing. The simulation results obtained for these three models are presented in Fig.3.3, Fig.3.4 & Fig.3.5.

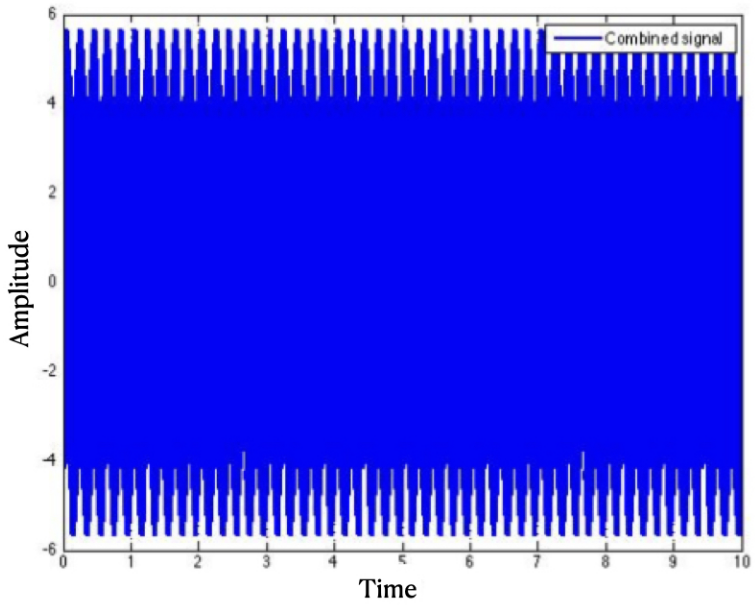

Figure 3.2 Summated Sinusoids with Gaussian Noise

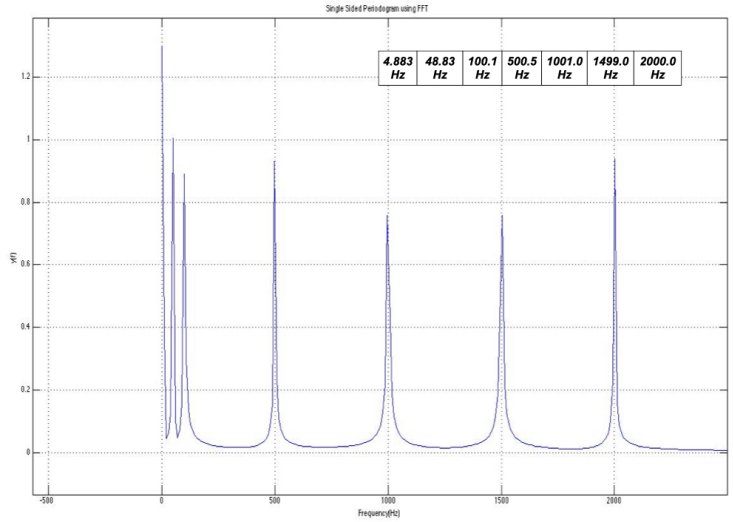

Figure 3.3 Spectrum using Periodogram without any preprocessing / filtering

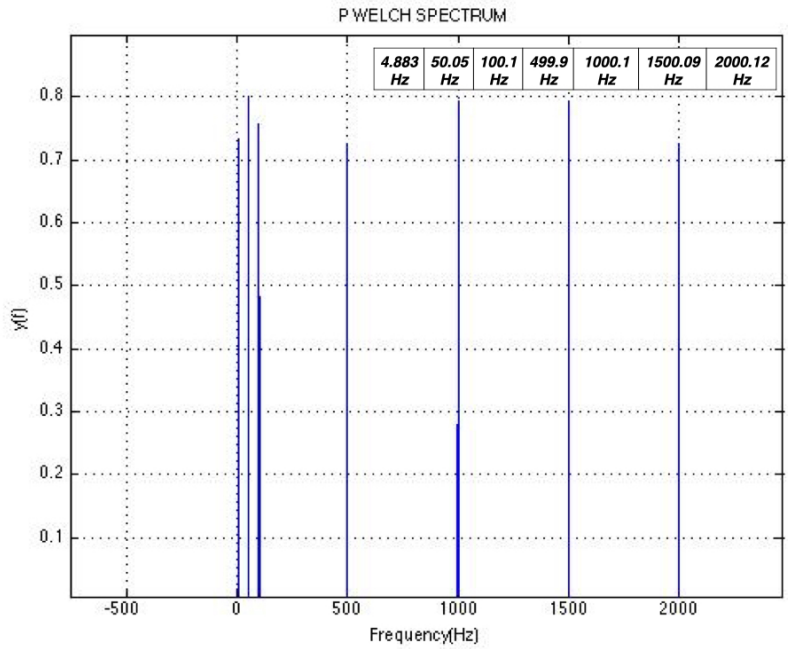

Figure 3.4 Spectrum using Welch Method

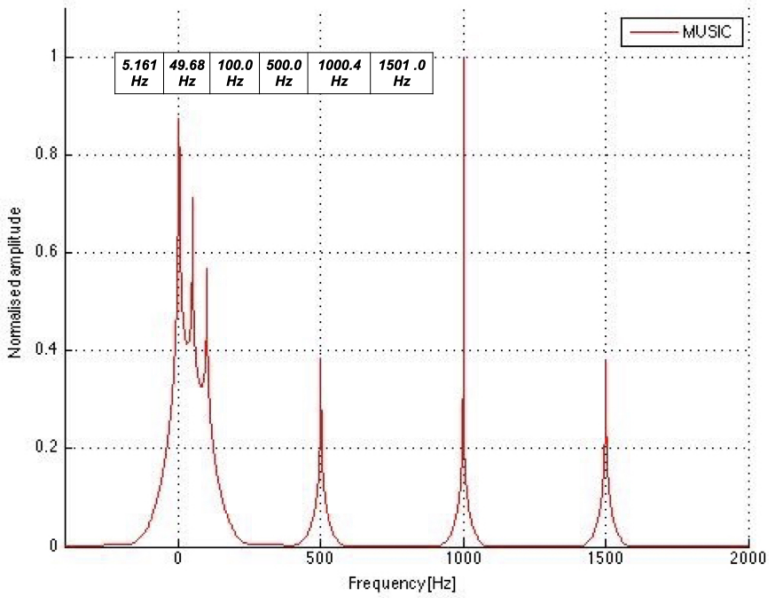

Figure 3.5 Spectrum using MUSIC Algorithm

The Table 3.1 presents the peak frequencies identified for three methods and the corresponding error with the reference signal frequencies.

Table 3.1 Mathematical Model - Identified Frequencies and Errors

Frequency (Hz)							
Input Spectrum	**5**	**50**	**100**	**500**	**1000**	**1500**	**2000**
Periodogram	4.883	48.83	100.1	500.5	1001.0	1499.0	2000.0
Error (Input-Periodogram)	0.117	1.17	-0.1	-0.5	-1.0	1.0	0.0
Welch Method	4.883	50.05	100.1	499.9	1000.10	1500.09	2000.12
Error (Input-Welch)	0.117	-0.050	-0.1	0.1	-0.105	-0.092	-0.122
MUSIC Algorithm	5.161	49.68	100.0	500.0	1000.4	1501.0	-
Error (Input-MUSIC)	0.161	0.320	0.0	0.0	0.400	1.000	-

It is observed from Table 3.1 that the maximum error is 1.0 Hz for Periodogram at two frequency points (1000 Hz and 1500 Hz) and one frequency point (1500 Hz) for MUSIC algorithm. Of all the three methods employed, the Welch method provides better frequency identification with minimum error. Maximum estimation error of 0.122 Hz is observed at 2000 Hz with Welch method. It is clearly observed that the Welch Method and the MUSIC Algorithm are providing better results compared to Periodogram with FFT. The mathematical models were also tested in Hardware-In-Loop-Simulation (HILS) setup with the same summated sinusoidal inputs.

3.4 Hardware-In-Loop Simulation and Validation

Data Acquisition Platform with NI's LabVIEW [67] is configured and used for HILS with necessary hardware for input signal simulation. The data flow diagram for HILS is as shown in Fig.3.6. The amplitude of the signal is selected as 1.0V for the simulation setup. This value is not varied while carrying out Hardware-In-Loop-Simulations as the

objective is to determine the dominant frequencies, irrespective of the amplitude of the input signal.

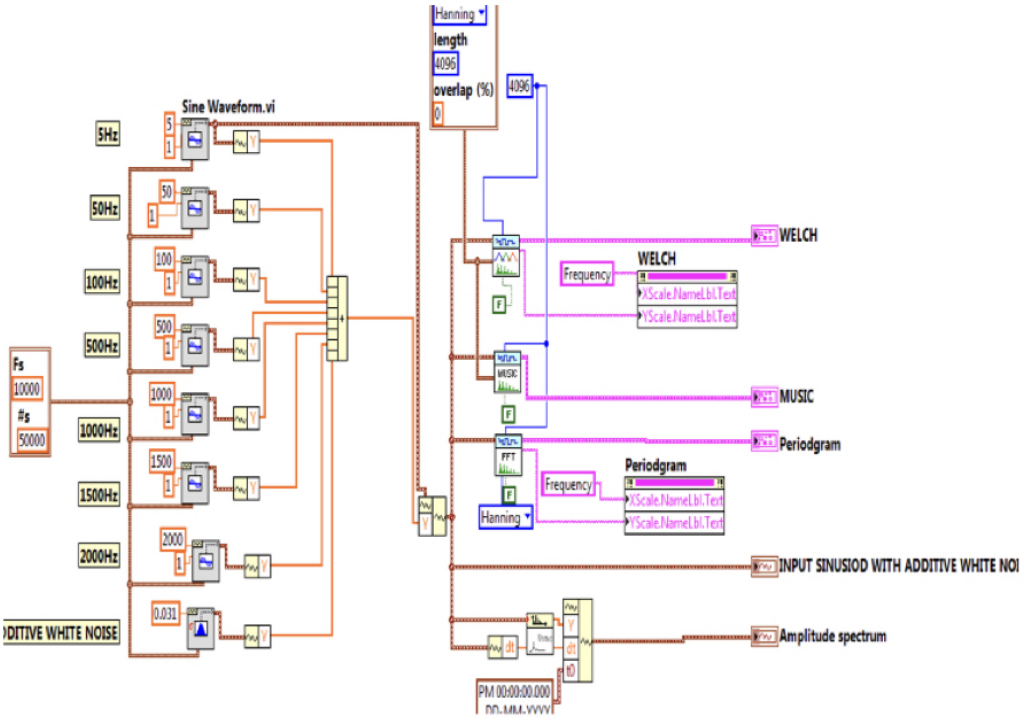

Figure 3.6 Data Flow Diagram for Hardware-In-Loop-Simulation

The time data of the input summated sinusoid added with white Gaussian Noise and the corresponding amplitude spectrum of the input signal are shown in Fig. 3.7. This summated sinusoidal signal with white gaussian noise is fed as an input to the HILS test setup as a raw signal without any filtering and pre-processing.

With the mathematical model implemented in the HILS Test setup for the summated sinusoidal input, the output of Periodogram, Welch Method and the MUSIC Algorithm are shown in Fig.3.8 – Fig.3.10. The identified frequencies by the three methods and the error from the known input signal frequencies are shown in table 3.2.

It is observed from the above Hardware-In-Loop-Simulation results that the Welch method is providing better closeness to all the frequencies considered for simulation points to the expected frequency peaks compared to the other two methods. The processing time for the MUSIC algorithm is more and this method is generally used for time-frequency analysis of the signals. MUSIC algorithm involves computationally intensive matrix decomposition operations and is not suitable for real-time applications in a flight vehicle.

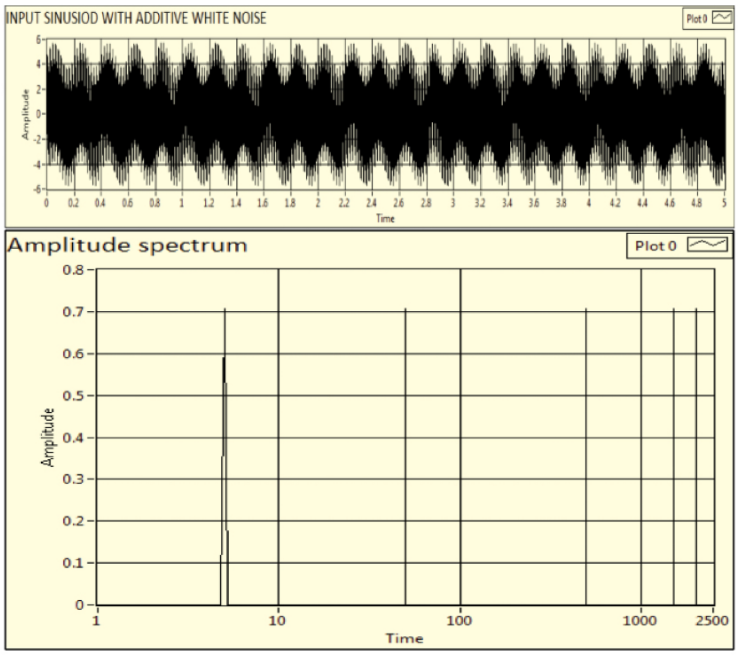

Figure 3.7 Input Signal Time Data and Amplitude Spectrum

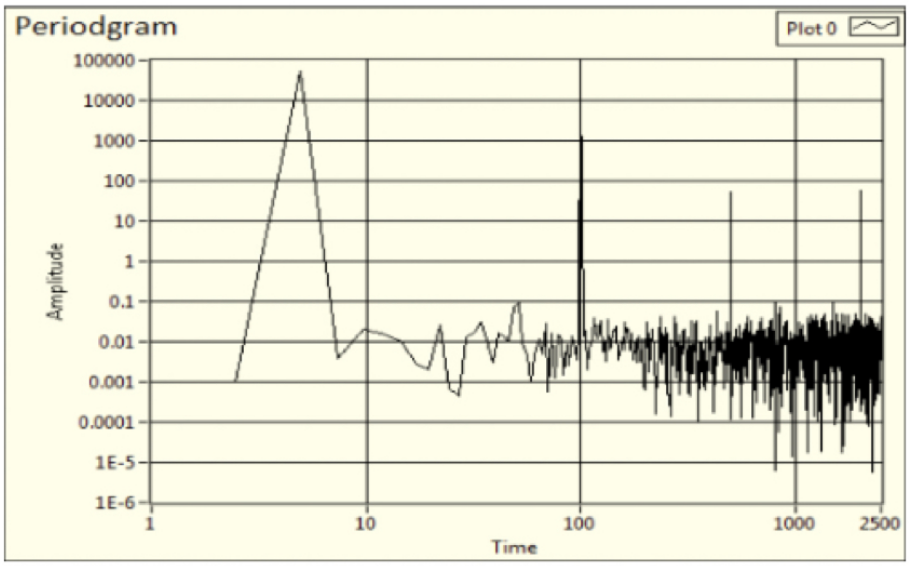

Figure 3.8 Spectrum using Periodogram

As the periodogram doesn't involve windowing and averaging method, the error in the estimation is more. From the statistical data obtained for the three methods using HILS platform, it is concluded to proceed with Welch method for frequency domain analysis of random vibration signal with suitable windowing techniques to calculate PSD and identify the dominant frequencies.

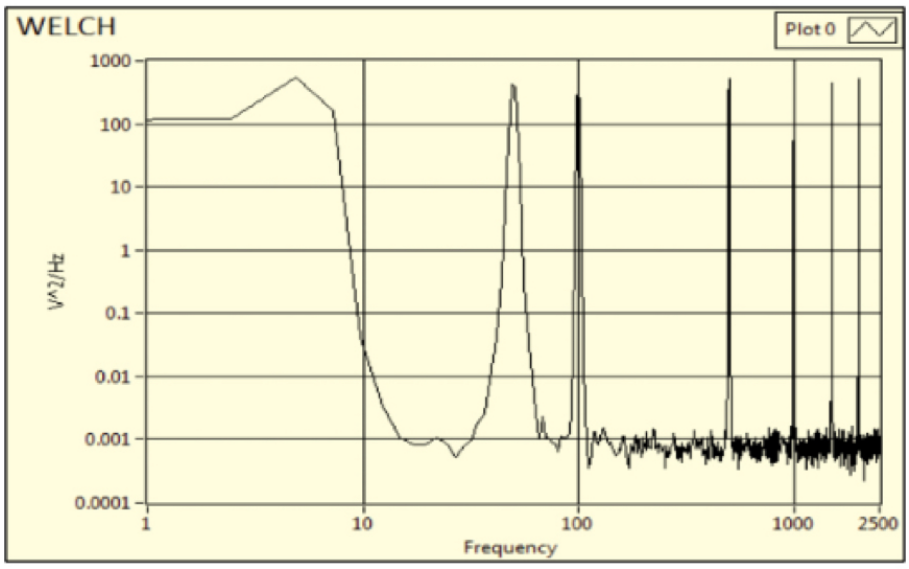

Figure 3.9 Spectrum using Welch Method

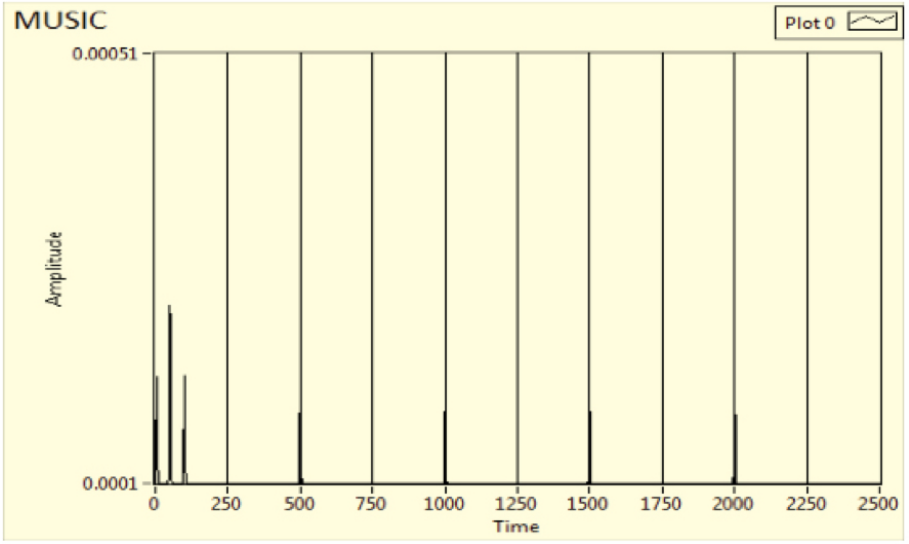

Figure 3.10 Spectrum using MUSIC Algorithm

Table 3.2 Comparative Frequency Estimates – HILS

Frequency (Hz)							
Input Spectrum	**5**	**50**	**100**	**500**	**1000**	**1500**	**2000**
Periodogram	4.871	48.5	100.176	500.488	1000.98	1501.49	1999.53
Error (Input-Periodogram)	0 .129	1.5 -	0.176	-0.488	-0.98	-1.49	0.47
Welch Method	5.06	49.627	100.092	499.767	1000.93	1499.07	1999.67
Error (Input-Welch)	-0.06	0.373	-0.092	0.233	-0.93	0.93	0.33
MUSIC Algorithm	4.913	48.837	100.093	500.051	1000	1499.07	1999.67
Error (Input-MUSIC)	0.087	1.163	-0.093	-0.051	0	0.93	0.33

3.5 Random Vibration Test Setup

The test section mounted on the Shaker with Head expander is subjected to random vibration for a level of 0.02 g^2/ Hz i.e. 5.35 g_{rms} for a duration of 200 secs in the longitudinal direction as per the input vibration signal profile shown in Fig.3.11 as described in test method IV of MIL- STD-810G. The Random Vibration Test setup is shown in Fig. 3.12.

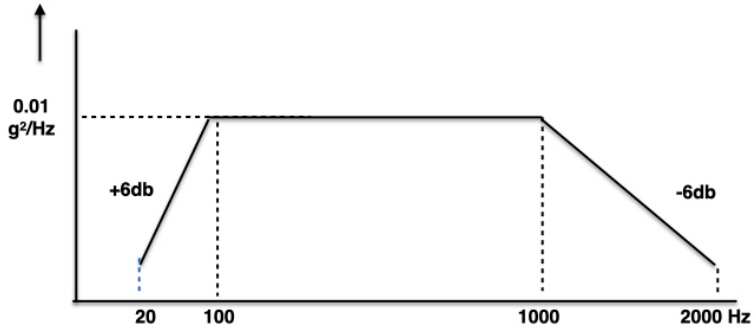

Figure 3.11 Input Spectrum

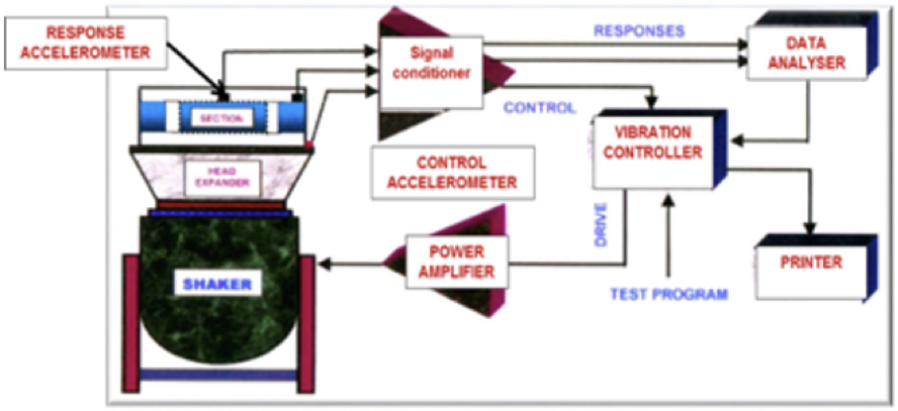

Figure 3.12 Vibration Test Setup

Multiple sensors to monitor vibration at different locations of the test section are mounted depending on the number of flight components integrated in the section. ICP Accelerometer 353B14 with the following specifications is used to measure the vibration response from the test article.

Make : PCB Piezotronics, NY

Sensitivity : 5 mV/g

Measurement Range : +/- 1000g peak

Frequency Range : 1-10000 Hz

Resonant Frequency : 70 KHz

Broadband Resolution : 0.01 g_{rms}

Nonlinearity : 1 %

Spectral Noise : 1Hz - 6400 µg/ Hz, 10 Hz - 1400 µg/ Hz

100 Hz - 360 µg/ Hz, 1 KHz - 128 µg/ Hz

Sensing Element : Quartz with Shear Sensing geometry

The reference controller with Data Analyser (DEWE-4010) has a throughput of 45 Mb/s and maximum sampling frequency of 10 KHz. The Data Analyser uses 1024-point analysis with Exponential type averaging of 64 lengths max. The window and overlap are set in AUTO as a standard practice. The number of points and averaging length are so decided that the output is of 6-sigma accuracy. The Data Analyser results are considered as reference input for all structural designs. The HILS validated model with known frequency signal is tested with the random vibration response data. The raw signal

data from one vibration sensor digitised in the Controller is taken as the reference input for processing through different algorithms. The processed PSD output from the Data Analyser is used as the benchmark for comparison with the outputs of the algorithms used. Fig 3.13 and 3.14 detail the time series of one response sensor data retrieved from the controller and the corresponding amplitude spectrum. The Power Spectral Density (PSD) plot of the same response is presented in Fig. 3.15.

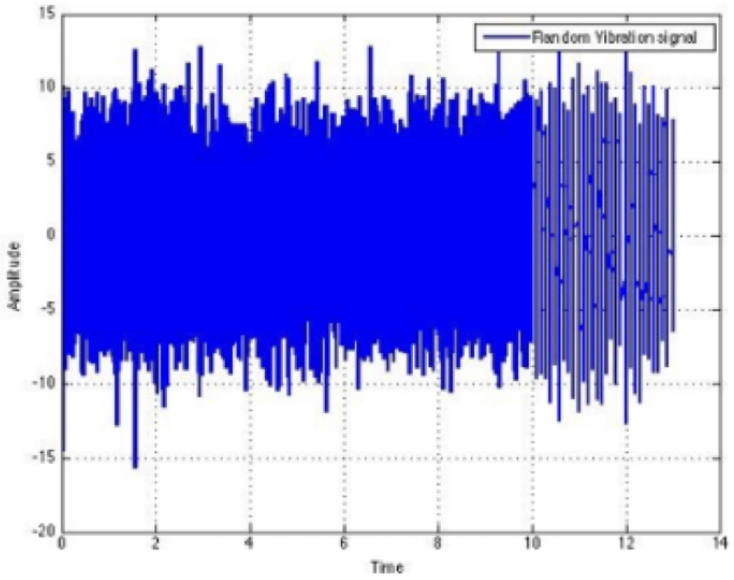

Figure 3.13 Response Sensor - Time Domain Series

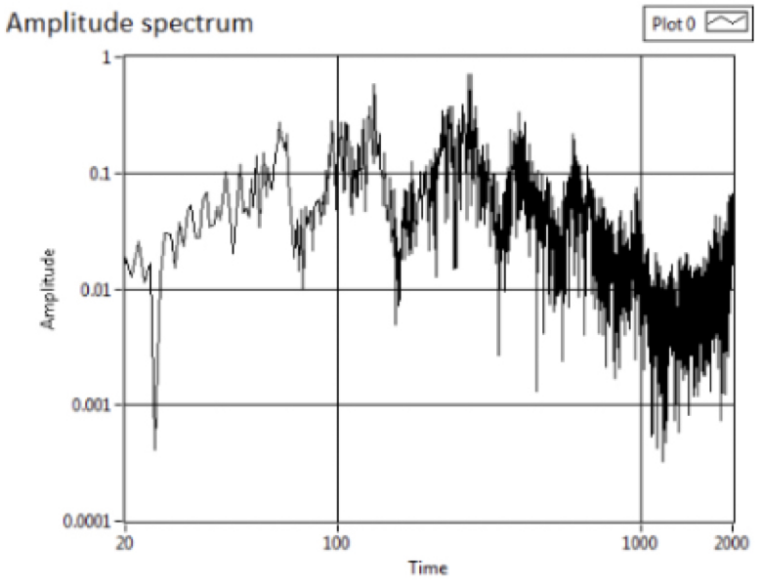

Figure 3.14 Response Sensor - Amplitude Spectrum

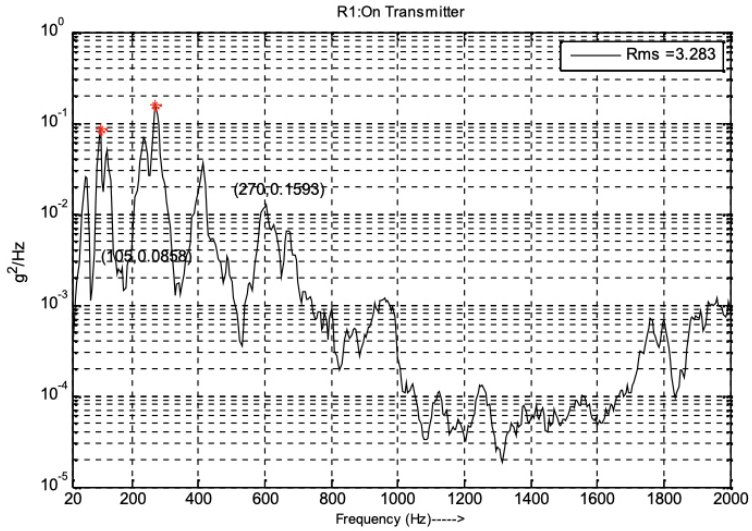

Figure 3.15 Response Sensor - Power Spectral Density

To verity both the Mathematical and HILS models with Random vibration data, the retrieved Response Sensor data from the reference controller are analysed. The single sided amplitude spectrum plot from the Mathematical model of FFT Spectrogram and the Welch Method are presented in Fig. 3.16 and the corresponding amplitude spectrums from the HILS model are presented in Fig. 3.17.

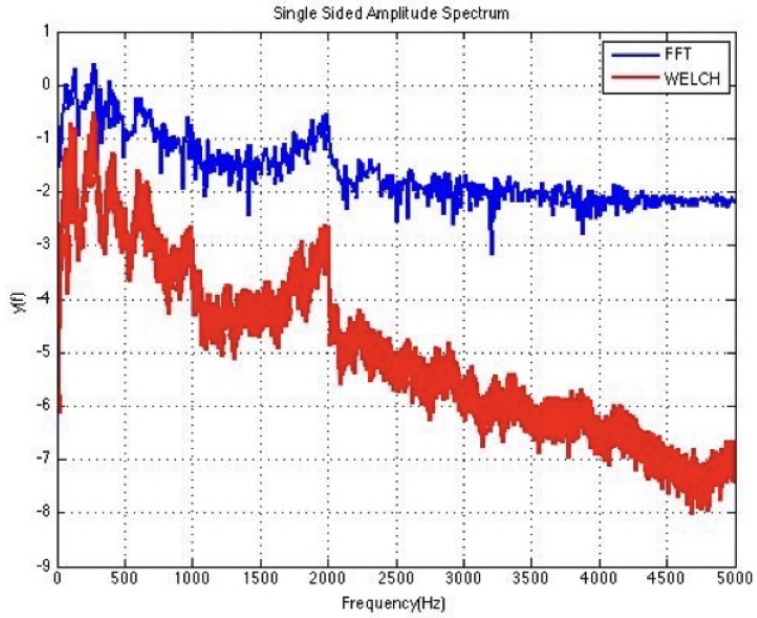

Figure 3.16 Random Vibration Response Signal processed with Mathematical Models- Periodogram and Welch Method

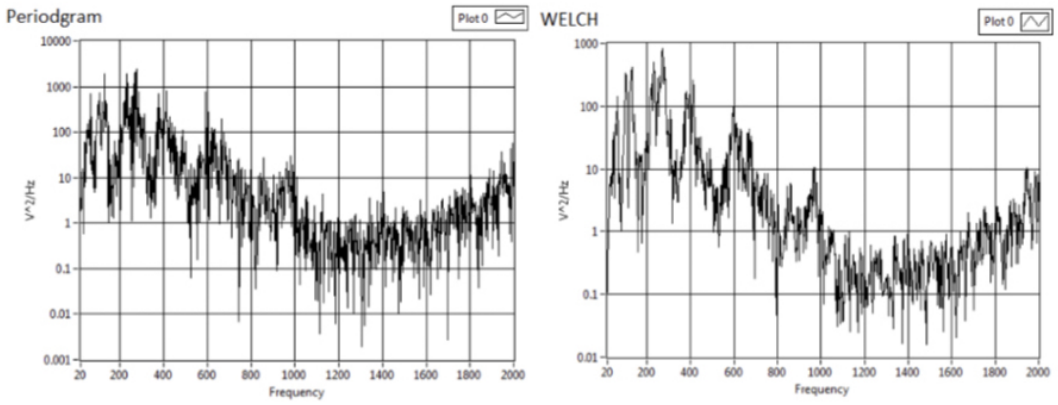

Figure 3.17 Random Vibration Response Signal processed in HILS Test setup - Periodogram and Welch Method

It is observed from Fig.3.16 that the power spectrum of the random vibration response signal with mathematical model are matching in signature, but the amplitudes are varying. This variation is due the scaling factor implemented in the model. With HILS Test setup, the resultant spectrum from both Periodogram and the Welch Method are matching in trend and signature as presented in Fig.3.17

Of the multiple sensors mounted on the test object, one sensor is selected such a way that the dominant frequencies are distributed across the range of interest. The Response Sensor Spectrum from the reference controller is shown in Fig 3.18. From the PSD computation of the sensor response, it is observed that there are two dominant frequencies, where the structure is getting excited i.e. at f1 = 293 Hz with an amplitude of A1 = 0.8882 g^2 / Hz and f2 = 839.8 Hz, with A2 = 0.7978 g^2 / Hz. The RMS value of the response is at 9.883 g_{rms} indicating an overall amplification factor of 1.85 induced due to mechanical interface and structural integrity of the test article.

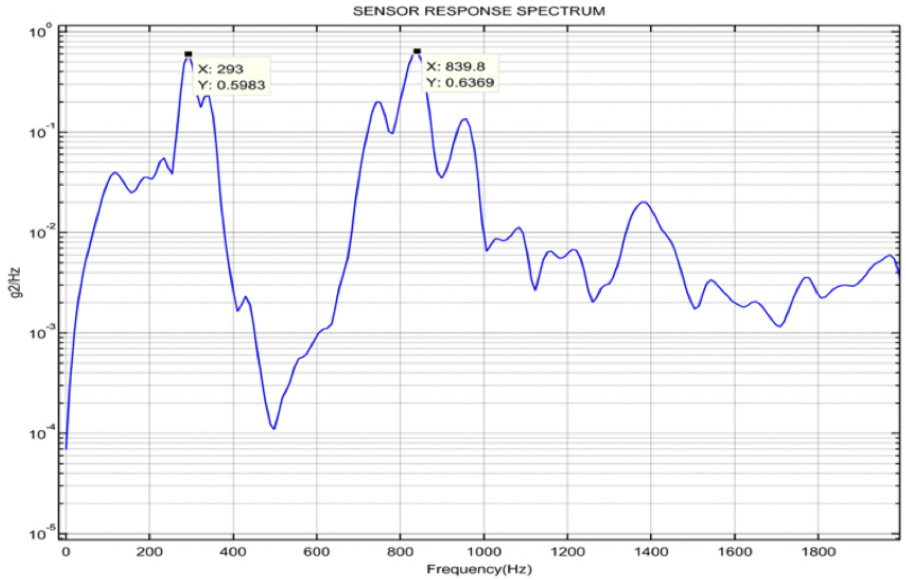

Figure 3.18 Response Sensor Spectrum from Reference Controller

3.6 Results and Analysis

Based on the simulation results of the known sinusoidal signals presented in section 3.3 and 3.4, application requirements and necessity & sufficiency of the methodology, Welch method with 4096-point analysis is chosen for further study and simulation. Three windowing methods viz. Hann, Hamming and Blackman-Harris are applied with 0%, 25% and 50% overlap over the sensor response data in frequency domain analysis. The study and simulations are carried out both using Mathematical Model and Hardware-in-Loop-Simulation setup. The raw data of the sensor response signal from the output of the ADC of the Data Analyser is retrieved and is used as the input array for processing in the selected algorithms with different windowing techniques and data point overlapping. The discrete data are first simulated in the mathematical model of the Welch algorithm to verify the performance and matching of the PSD profile with normalised amplitudes. Confirming the response profile in all the three methods of windowing with Welch method, the same simulation algorithm is used in the HILS platform. The overall sensor output spectrum of the PSD plot is similar in all the three windowing and three overlapping methods and are comparable with the reference controller PSD plots. However, variations in the amplitude are observed at dominant frequencies.

Firstly, the Welch method with Hann Window and three overlapping of 0%, 25% and 50% are processed in HILS setup. The power spectrum for all the three overlapping types and the dominant frequency zone at low frequency and high frequency ranges are zoomed-in and shown in Fig. 3.19, 3.20 & 3.21. It is observed that the overall spectrum and RMS value remains the same in all the three overlapping procedures. However, the number of dominant frequency peaks vary with percentage of overlap. As the energy is distributed at nearby frequencies, the amplitudes are marginally reduced. The number of peaks observed at both the frequency range of interest, the frequency in Hz and the average frequency with HANN Window are as presented in Table 3.3.

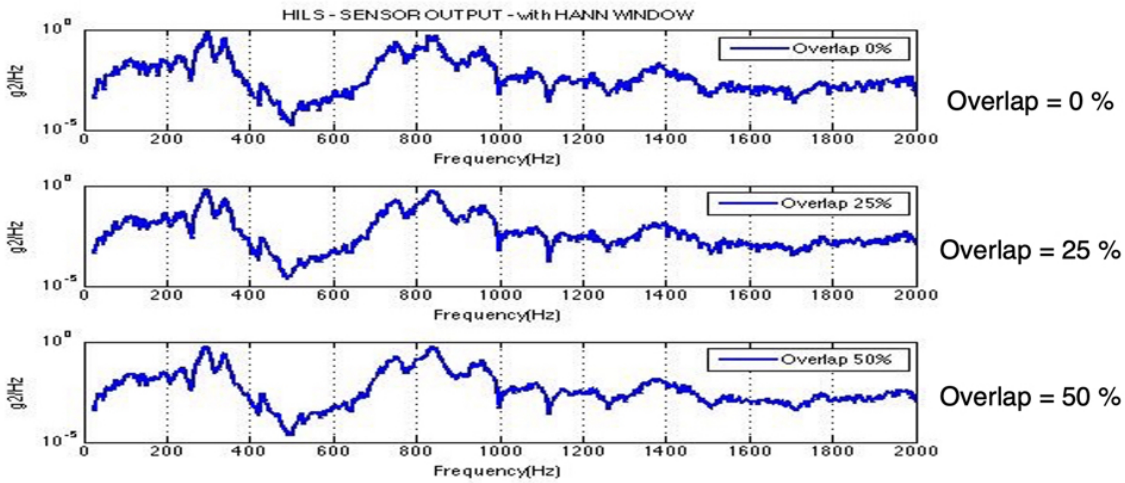

Figure 3.19 Power Spectrum (HANN) with three different overlaps

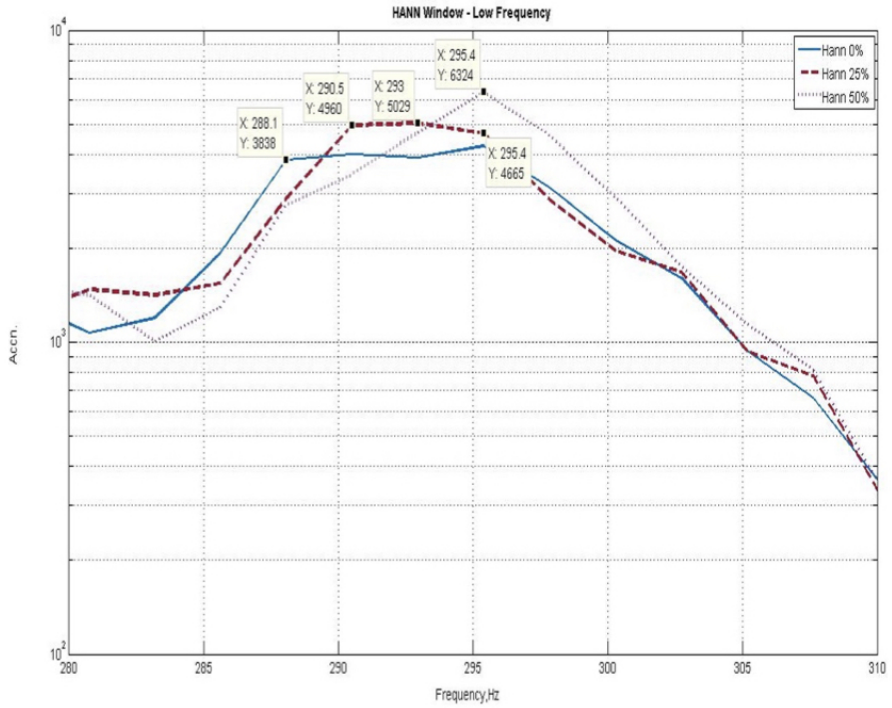

Figure 3.20 HANN Window Zoomed In plot - Low Frequency range

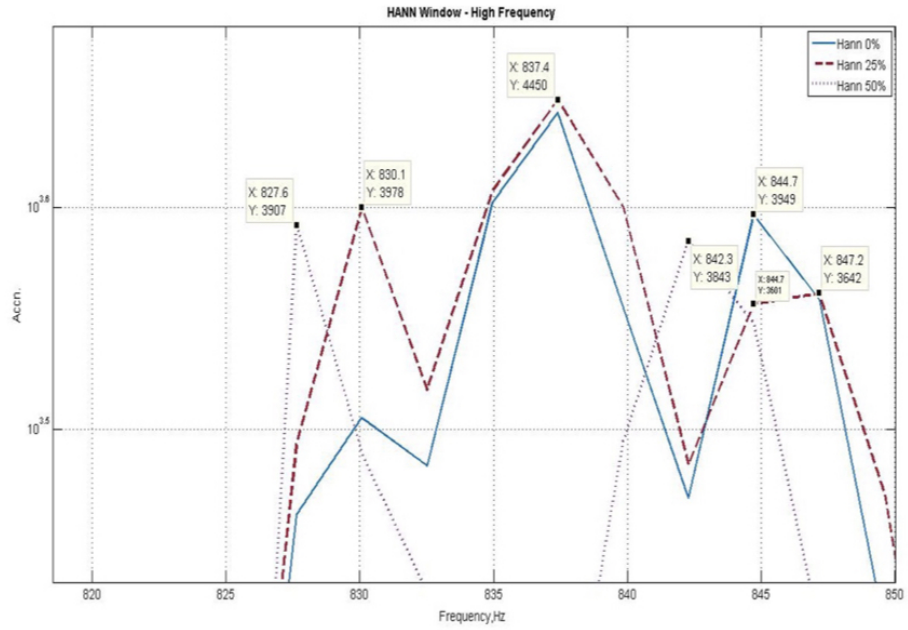

Figure 3.21 HANN Window Zoomed In Plot - High Frequency range

Table 3.3 HANN Window - Identified Frequencies

Overlap	No. of Peaks	Freq. Hz	Average Freq. Hz		No.of Peaks	Freq. Hz	Average Freq. Hz
0%	2	288.1 295.4	291.75		3	830.1 837.4 844.7	837.4
25%	2	290.5 295.4	292.95		3	830.1 837.4 847.2	839.85
50%	1	295.4	295.4		2	827.6 842.3	834.95

Similarly, the sensor output signal is processed through the Hamming Window with 0%,25% and 50% overlapping and the plots are presented in Fig.3.22, Fig.3.23 & Fig.3.24. The identified frequencies with HAMMING Window are presented in Table 3.4.

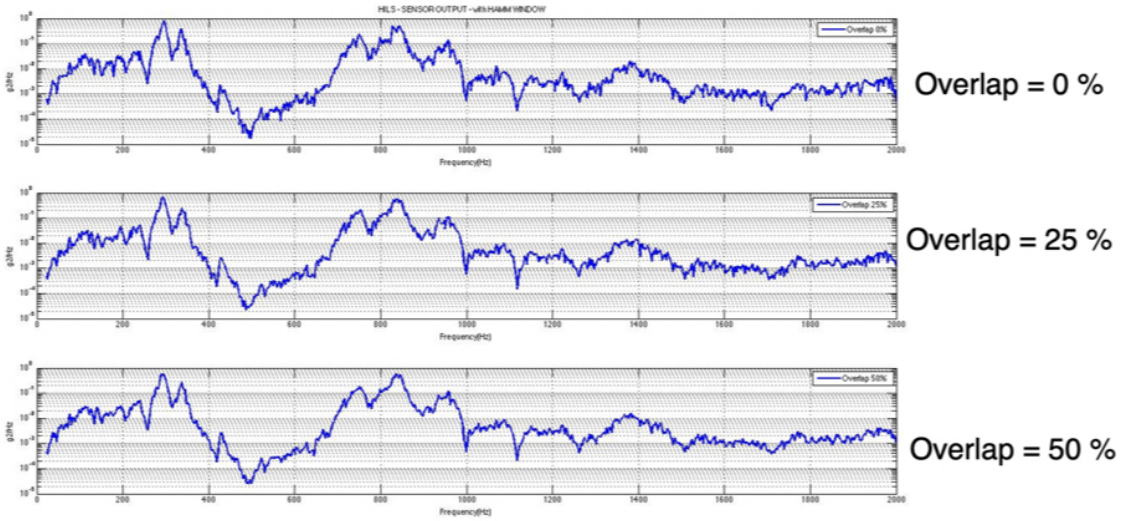

Figure 3.22 Power Spectrum (HAMMING) with three different overlaps

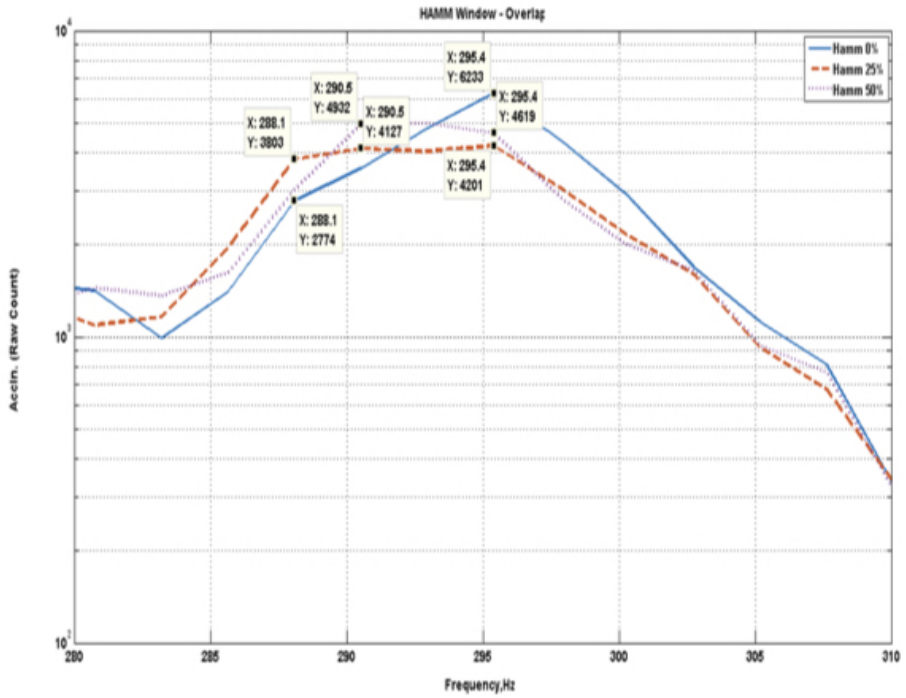

Figure 3.23 Hamming Window Zoomed In Plot - Low Frequency range

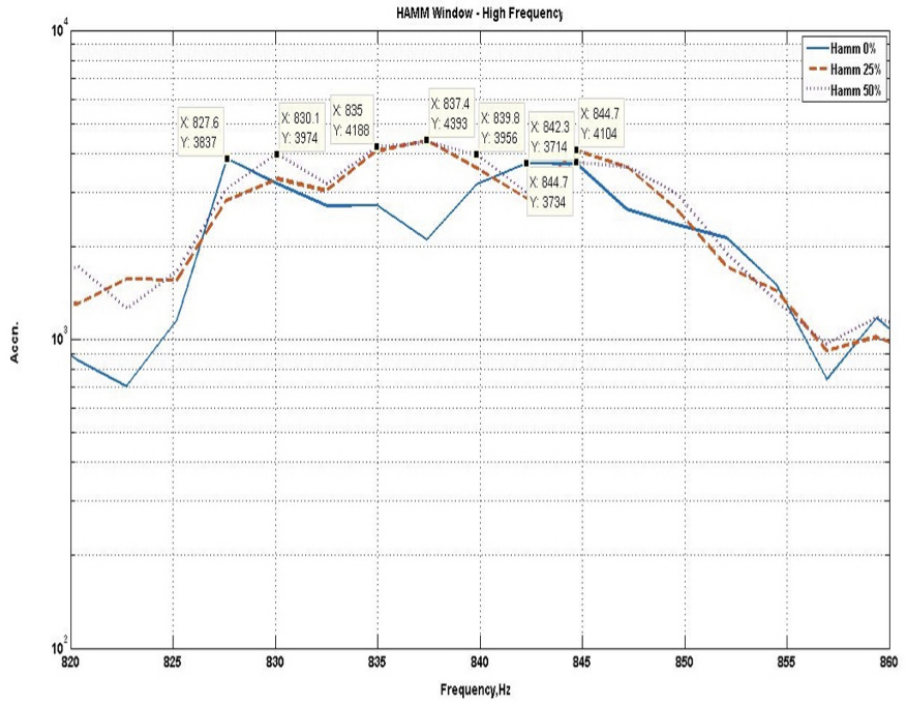

Figure 3.24 Hamming Window Zoomed In Plot – High Frequency range

Table 3.4 HAMMING Window - Identified Frequencies

Overlap	No. of Peaks	Freq. Hz	Average Freq. Hz		No.of Peaks	Freq. Hz	Average Freq. Hz
0%	1	295.4	295.4		3	827.6 842.3 844.7	838.2
25%	3	288.1 290.5 295.4	291.33		3	830.1 837.4 844.7	837.4
50%	2	290.5 295.4	292.95		3	830.1 835.0 844.7	836.6

Thirdly, Blackman-Harris window is configured with the three overlapping of 0%,25% and 50%. The sensor response is processed with the configured windowing and overlapping and the plots are presented in Fig. 3.25, Fig.3.26 & Fig.3.27. The identified frequencies with Blackman-Harris Window are presented in Table 3.5.

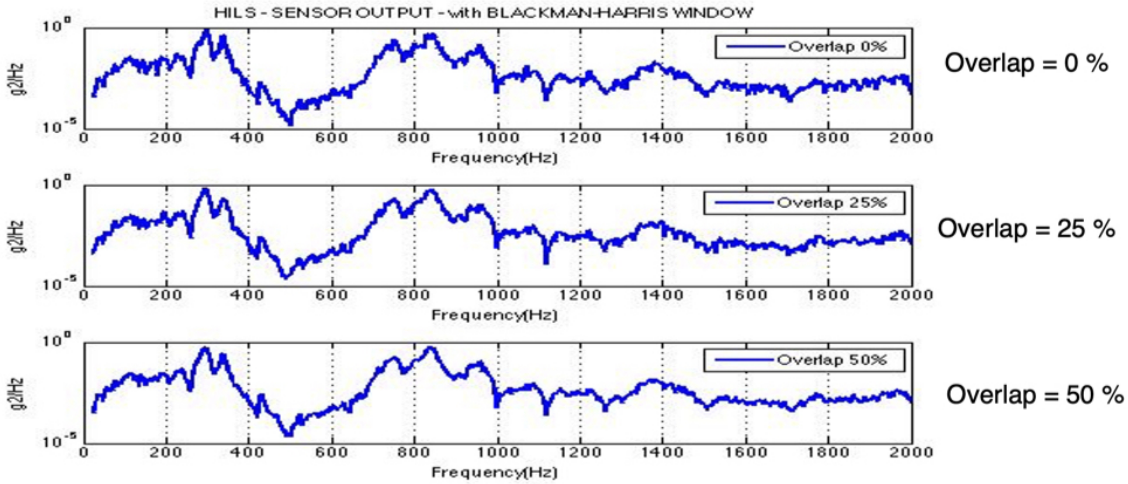

Figure 3.25 Power Spectrum (BLACKMAN-HARRIS) with three different overlaps

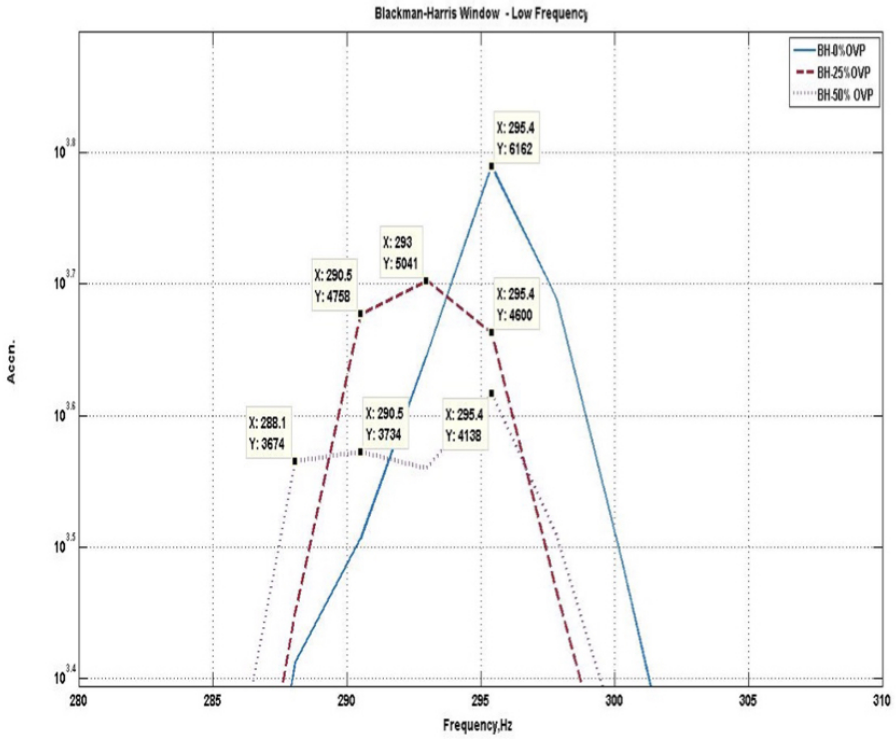

Figure 3.26 Blackman-Harris Window Zoomed In Plot – Low Frequency range

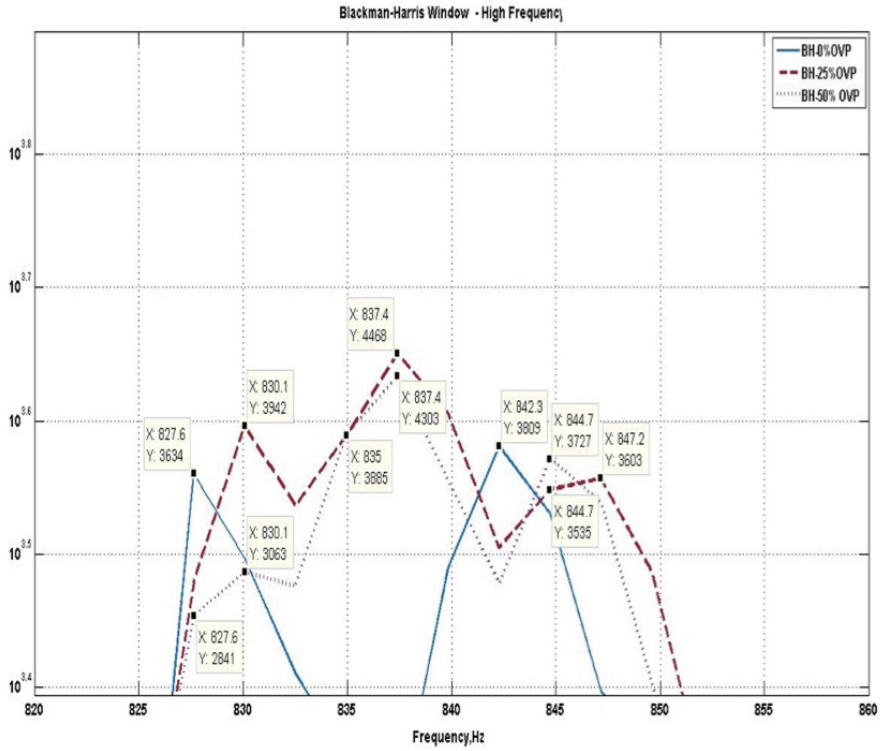

Figure 3.27 Blackman-Harris Window Zoomed In Plot – High Frequency range

Table 3.5 BLACKMAN HARRIS Window - Identified Frequencies

Overlap	**No. of Peaks**	**Freq. Hz**	**Average Freq. Hz**		**No.of Peaks**	**Freq. Hz**	**Average Freq. Hz**
0%	1	295.4	295.4		2	827.6 842.3	834.95
25%	3	290.5 293 295.4	292.96		3	830.1 837.4 847.2	838.23
50%	3	288.1 290.5 295.4	291.33		3	830.1 837.4 844.7	837.4

Flight performance parameters are telemetered to ground during the flight trials and recorded at the Ground Telemetry Stations, as described in Chapter 1. Among the sensors mounted on the flight vehicle, one modal vibration sensor data is selected randomly. This process is repeated for two more flight trials. The range of On-board vibration sensor is +/- 5g and the bandwidth is 150 Hz. First 50 secs of the received data from lift-off is considered to verify and validate the models. Three flights data are analysed using the HILS setup and the identified data points are compared with the modal analysis and post-flight ground reference analyser. The analysis is carried out with 512-point FFT and with averaging window of 1. The variation of Modal frequency with time, observed from the HILS model is compared with those obtained in post-flight reference analyser and pre-flight predictions. It can be noted from the plots (Fig. 3.28, Fig.3.29 & Fig.3.30) that the identified modal frequency values of the HILS model are well within the predicted range and closely matching with post-flight reference analyser for all the three flights, namely, Flight1, Flight2 and Flight3.

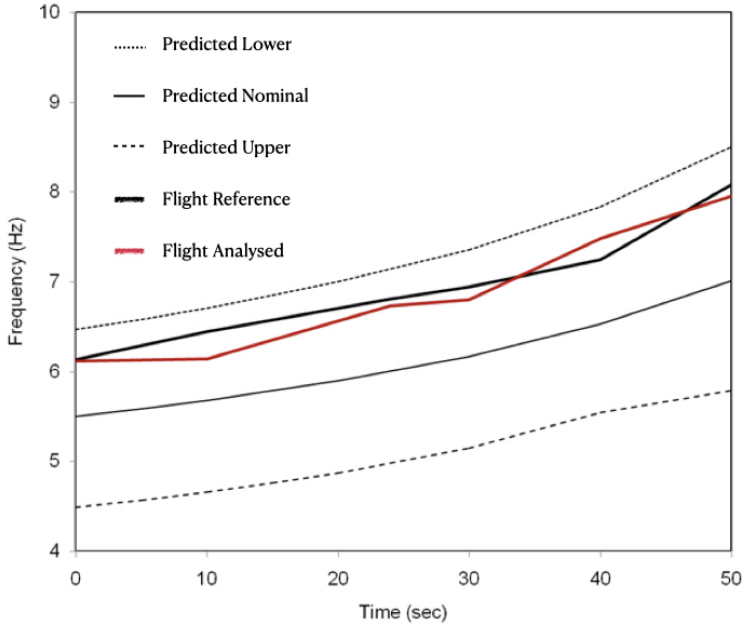

Figure 3.28 Modal Frequency - Flight1 Telemetered Data

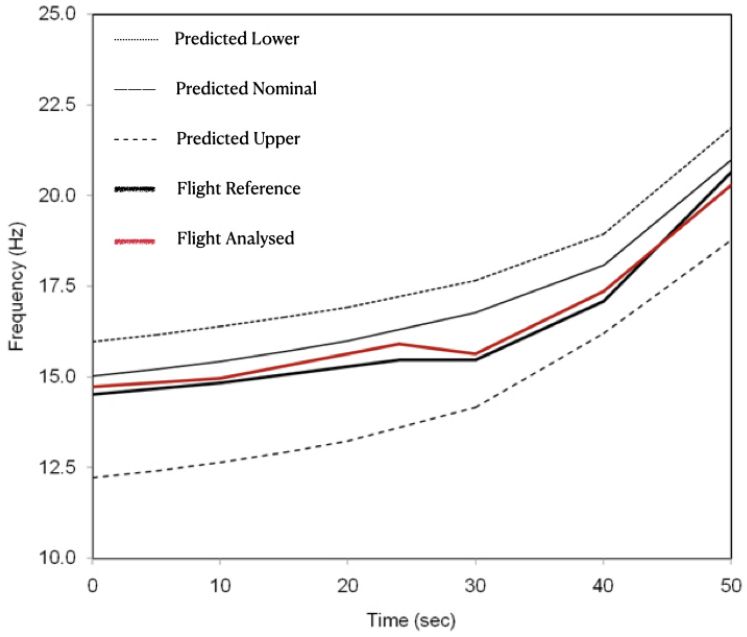

Figure 3.29 Modal Frequency - Flight2 Telemetered Data

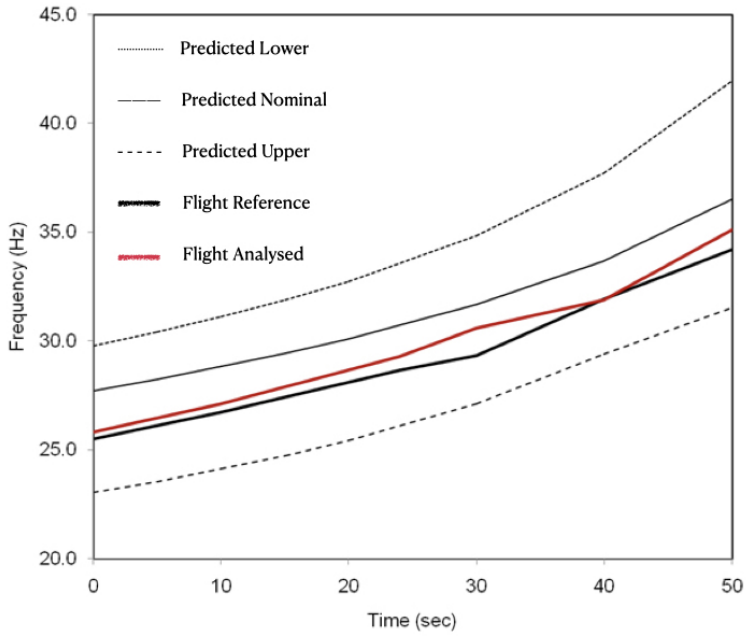

Figure 3.30 Modal Frequency - Flight3 Telemetered Data

With reference to the identified dominant frequencies at 293 Hz and 839.8 Hz by the reference controller, it is observed from Table 3.3, Table 3.4 and Table 3.5 that the 25% overlapping provides a closer estimation for Hann Window and Blackman-Harris Window. The identified dominant frequency values vary widely in the case of Hamming window. In addition, it may also be observed that the estimation error is less with Hann Window and Blackman-Harris Window techniques compared to the Hamming window. In general, the resolution of frequency (ΔF)increases with increase in value of N in N-point FFT. It is also observed that the number of peaks increased in the frequency range of interest with increasing overlapping(0%, 25 % and 50%). However, the amplitudes at dominant frequencies are lower than that of reference spectrum. The reason for the same can be attributed to the distribution of energy over the nearby frequencies. But it is seen that the overall PSD spectrum of the sensor output remained the same. From the analysis of flight telemetry data, it is observed that the identified frequency values are within the predicted bounds and the proposed algorithm is yielding expected results. Table 3.6 presents the execution time reported for each of the models simulated using the HILS platform. This execution time does not include the time required for A-to-D conversion and other pre-processing/post processing requirements for the specific application. The pre-processing

is carried out in the external electronics and charge amplifiers. As the processing circuit design is common for all the three methods (Hann, Hamming and Blackman-Harris) and the time taken will be the same irrespective of the algorithm used, the hardware loop-time can be considered constant.

Table 3.6 Execution Time for Windowing and Overlapping

Overlap	HANN	HAMMING	BLACKMAN-HARRIS
0%	1047-1052 μsec	1047-1051 μsec	1049-1053 μsec
25%	1050-1059 μsec	1051-1055 μsec	1049-1077 μsec
50%	1090-1096 μsec	1092-1107 μsec	1089-1096 μsec

Filters are not considered necessary as all the sensor responses are compensated signals. During the controller setting and sensor calibration, offset is defined and limited to omit the noise component introduced by the cables and environment. Statistical bounds are limited to the closeness to the dominant Frequency points and the corresponding amplitudes with the reference PSD plot.

3.7 Conclusion

The study and analysis of flight vehicle vibration signals presented in this chapter is considered to design an architecture for on-board analysis of vibration signals and PSD computation. It is concluded from the simulation analysis that the Welch method is suitable for analysis of flight vibration signals compared to Periodogram and MUSIC algorithm considered for study. The architecture has to be developed with the choice of window type and the overlapping % as configurable parameters. Hence, a suitable architecture is proposed in the next chapter for flight vehicle applications to extract instantaneous PSD and identify the dominant frequency from the body rates sensed by the accelerometers.

Chapter 4

Architecture for Frequency Domain Analysis

It may be noted from the mathematical modelling and HILS test results presented in Chapter 3 that most suitable method for on-board analysis of flight random vibration signal is the Welch method with necessary Windowing and Overlapping functions to calculate the Power spectral density and thus closely identifying the dominant frequencies. These identified dominant frequencies and their variations with time during the time of flight can well be used to fine tune the control and autopilot parameters in real time flight to have better controllability and mission design. Based on the mathematical model simulation and HILS results, a suitable architecture is to be developed for flight vehicle applications to extract instantaneous PSD and identify the dominant frequency from the body rates sensed by the accelerometers. For the on-board application of a flight vehicle, a scheme is developed to suit the existing avionics configuration, electrical connectivity, data update rate and control system requirements.

4.1 Spectral Analysis

The generalised block diagram for the spectral analysis of flight vibration signals is shown in Fig 4.1. Major functional units for spectral analysis are the sensors to sense the body rates, Input filters and ADCs, Domain Conversion block, Computation of PSD and Maxima and interface to On-board Computer.

Accelerometer signals are converted to digitized input with Anti-aliasing filter and A-toD converter. The window function is needed to reduce the spectral leakage during

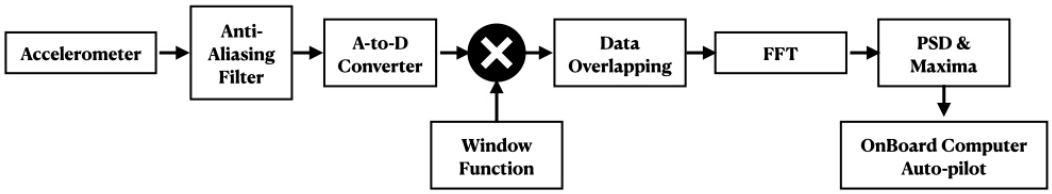

Figure 4.1 Generalised block diagram for Spectral Analysis

conversion from time-domain to frequency-domain. Overlapping of the input signal is to include more previous data points during analysis and FFT function translates the time domain signal into frequency domain for further computations to calculate power spectral density and identify the dominant frequency.

In this work, an architecture is proposed to implement Window Function, Data Overlapping, FFT, computation of PSD and Identifying the Frequency Maxima. The design and implementation of these blocks to perform frequency domain analysis of flight vehicle random vibration signals are presented in the following sections.

4.2 Windowing

In signal processing parlance, windowing refers to the process of modifying the input samples before transforming into the frequency domain by Fourier transforms, to reduce the spectral leakage and picket fence effect [68–72]. Many windowing functions are derived over the time for use in shaping and processing of the input signals [73]. Windowing is the process of smoothening the signal end points by multiplying the time data by finite length spectrum with smoothly varying amplitude that zeros out at the edges. The smooth ending at zero makes the endpoints of two successive data meet, avoiding sharp transitions. The loss in the total signal power induced by this type of weighing can be avoided by applying a correction factor. For the given signal x(t), applying a window w(t) results in the signal x'(t) as

$$x'(t) = x(t) * w(t) \tag{4.1}$$

Many types of windowing techniques are available to control the signal leakage. But

only a few are used in vibration signal analysis by the practitioners, because of suitability and efficiency. The three windowing techniques proposed in the literature [74] to analyse the spectral data of the flight mechanical structures are

- Hann Window
- Hamming Window
- Blackman-Harris Window

The Hann window can be viewed as the sum of a rectangular window and single cycle of cosine of function, that is

$$w\left(n\right) = 0.5\left(1 \; - \; \cos\frac{2\pi n}{N-1}\right) \tag{4.2}$$

and the zero-phase version is,

$$_0\left(n\right) = 0.5\left(1 \; + \; \cos\frac{2\pi n}{N-1}\right) \tag{4.3}$$

The ends of the cosine just touch zero, so the side-lobes roll off at about 18 dB per octave. The Matlab simulation result of the Hann windown is presented in Fig.4.2

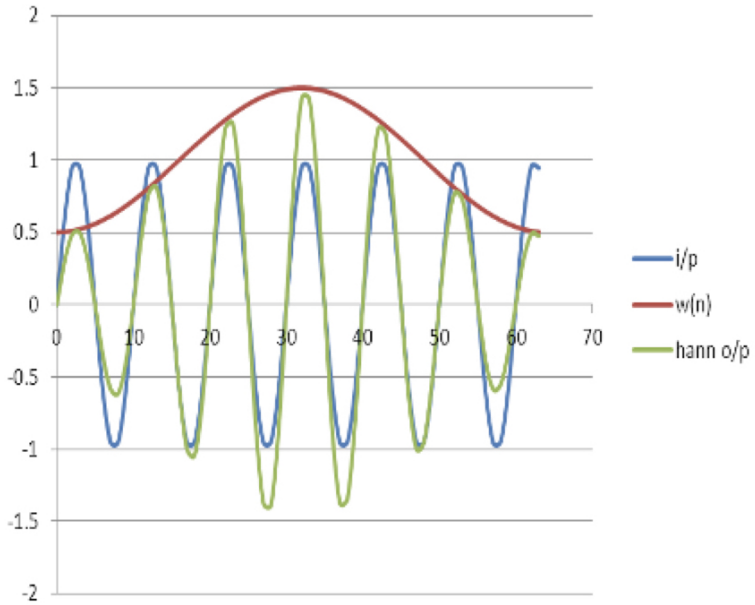

Figure 4.2 Simulated Hann Window Function

Hamming window nearly cancels the nearest maximum side-lobes in its transform using the form

$$w\left(n\right) = \alpha - \beta\left(1 \; - \; \cos\frac{2\pi n}{N-1}\right) \tag{4.4}$$

where $\alpha \; = \; 0.54$, $\beta \; = \; 0.46$

The constants are approximations of values α=25/46 and β = 21/46, which cancel the first sidelobe of the Hann window by placing a zero at frequency $(5\pi/N-1)$. Approximation of the constants to two decimal places substantially lowers the level of sidelobes satisfying the equiripple condition. In the equiripple sense, the optimal values for the coefficients are α=0.53836 and β=0.46164. The Matlab simulation result of the Hamming window function is presented in Fig.4.3.

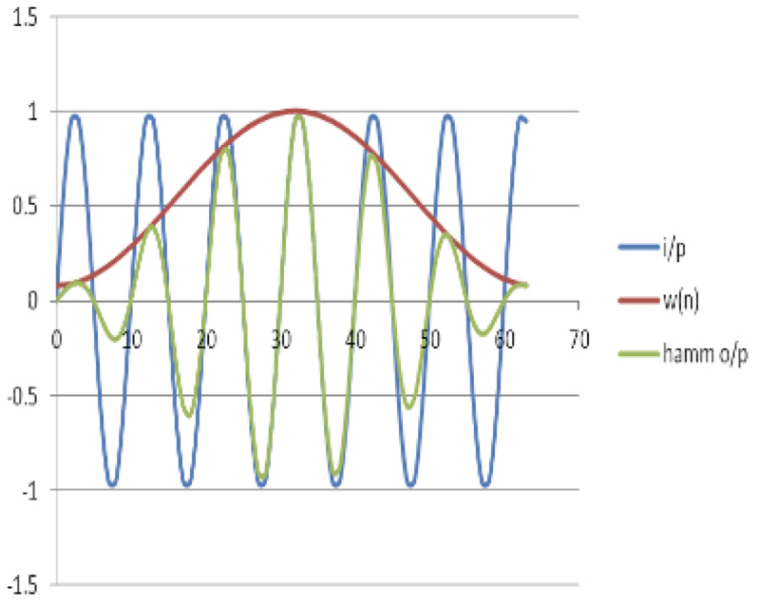

Figure 4.3 Simulated Hamming Window Function

Blackman window adds one more cosine term to cancel the 3rd and 4th side-lobes and cancels the window function to zero at the edges. The function for Blackman window can be expressed as

$$w_{Blkman}\left(n\right) \; = \; a_0 \; - \; a_1 * \cos\left(\frac{2\pi n}{N-1}\right) \; + \; a_2 * \; \cos\left(\frac{4\pi n}{N-1}\right) \tag{4.5}$$

where $a_0 = 0.42$, $a_1 = 0.5$ and $a_2 = 0.08$. The Blackman window, with the above approximation coefficients, provide excellent attenuation of sidelobes with only a modest increase

in computation over that required by the Hann and Hamming window. The Blackman-Harris window, another generalisation of the Hamming family, produced by adding more shifted sinc functions to minimize side-lobe levels, can be expressed as

$$w\,(n)\;=\;a_0\;-\;a_1\cos\;\left(\frac{2\pi n}{N-1}\right)+a_2\cos\;\left(\frac{4\pi n}{N-1}\right)-a_3\cos\;\left(\frac{6\pi n}{N-1}\right)\quad(4.6)$$

where $a_0 = 0.35875$, a_1=0.48829, a_2=0.14128 and $a_3 = 0.01168$. The Matlab simulation results of the Blackman-Harris window is shown in Fig 4.4.

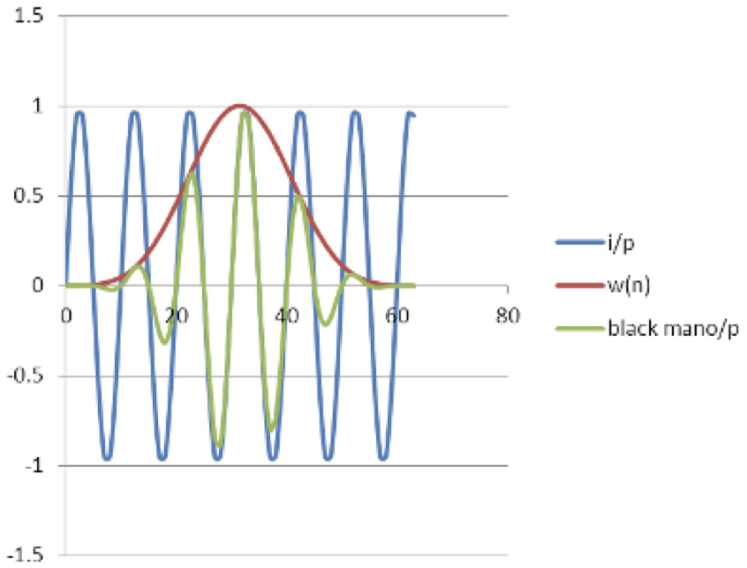

Figure 4.4 Simulated Blackman-Harris Window Function

The generalised equation for the above three window functions can be expressed as,

$$w\,(n)\;=\;a_0\;-\;a_1\cos\;\left(\frac{2\pi n}{N-1}\right)+a_2\cos\;\left(\frac{4\pi n}{N-1}\right)-a_3\cos\;\left(\frac{6\pi n}{N-1}\right)\quad(4.7)$$

where, 'n' is the discrete time index, 'N' is the window size. And $a_0, a_1, a_2 and a_3 are coefficients which$

Table 4.1 Window Type and Coefficient Values

Window Type	a_0	a_1	a_2	a_3
Hann	0.5	0.5	0	0
Hamming	0.54	0.46	0	0
Blackman-harris	0.35875	0.48829	0.14128	0.01168

In this work, windowing function (Eq.(4.7)) is implemented using FIR filter designed by employing parallel-pipelined architecture. Here, the FIR filter coefficients are derived from the mathematical model simulation. These coefficients are further optimised for the lowest possible Mean Squared Error (MSE) using the Simulated Annealing (SA) algorithm [75]. It is a probabilistic and iteration-based optimization algorithm, which approximates the global optimum of a given objective function. The optimization model was verified with the known cosine signal with 10dB additive gaussian noise. Fig.4.5 shows the FIR filter output plots with and without the optimization. It is observed from these plots that the noise ripples in the input are completely eliminated with the optimized coefficients of the filter generated using SA algorithm.

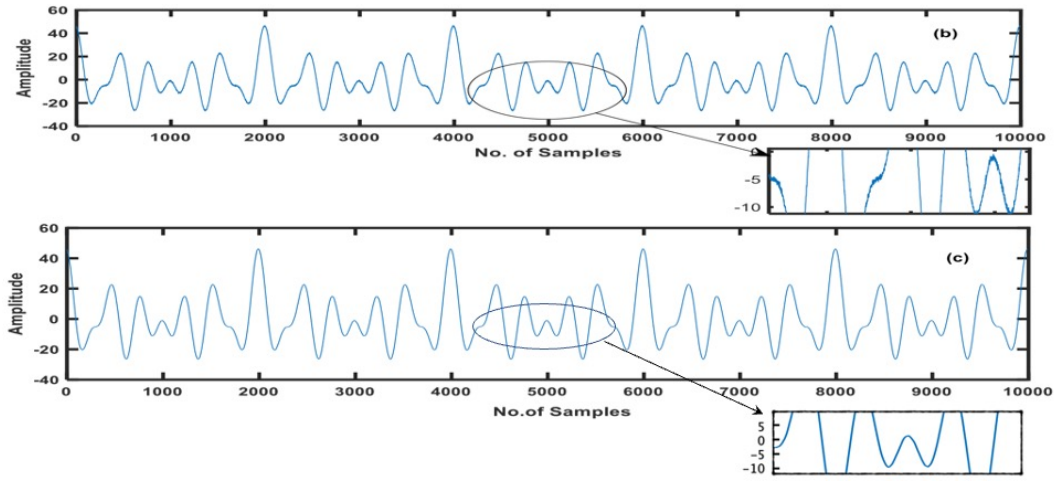

Figure 4.5 Filter Performance with SA Algorithm Optimization

The performance of the Hann, Hamming & Blackman-Harris filters are evaluated by computing MSE and SNR parameter with and without optimisation by SA Algorithm. As an example, the performance metrics computed for HANN window are presented in Table 4.2 for different noise levels. Assuming that the Standard Hann represents the Hann window-based FIR filter and SA–FIR HANN represents the FIR filter optimised with Simulated Annealing Algorithm, it may be noted that the optimised filter using SA algorithm produces better MSE and SNR than the standard filters in all the three cases of different noise levels. The optimised values are used for fine tuning the filter coefficients. Similarly, the optimised filter coefficients are derived for Hamming and Blackman-Harris windows also.

To design a FIR filter with the optimised coefficients for windowing, the time do-

Table 4.2 Performance Parameters of Standard FIR and SA-FIR filters

Noise -SNR in dB	**Filter Type**	**MSE**	**SNR (in dB)**
3	Std. Hann	12.2510	24.1775
	SA - FIR Hann	1.1101	24.8416
5	Std. Hann	12.1203	26.2487
	SA - FIR Hann	0.941835	27.0779
10	Std. Hann	11.9976	31.3157
	SA - FIR Hann	0.749809	31.9743

main filter coefficients must be restricted in number by truncation which is equivalent to multiplying by a window function of a finite width. Consider a low pass filter which is to be designed by windowing technique. Since we are multiplying in time domain, their corresponding frequency domain equivalents get convolved. We can divide this into two parts; convolution of H (ω) with main lobe and convolution with sidelobe. Convolution with the main lobe causes the broadening of the filter's transition region which is approximately the width of the main lobe. As the filter gets longer, the main lobe gets narrower and broadening becomes less noticeable. As long as filters have shorter transition regions, Convolution of sidelobes causes ripples in filter's frequency response. Large sidelobes causes higher ripples.

The FIR filter of order 8 is designed for the proposed architecture with a cut-off frequency of 2 KHz as the random vibration signal frequency range is 20 Hz -2 KHz. The Stop band frequency of 5 KHz is selected for smooth fall-off. For an attenuation factor of 50dB, passband of 2 KHz and stop band of 5 KHz, based on Fred harris' rule of thumb, the order of the filter works out to 7 and based in Kaiser's formula the order of the filter needed is 4. Hence, it is proposed to configure an FIR filter of Order 8 to meet the stringent flight vehicle avionics application requirements. The derived optimised coefficients are stored in the Block RAM of FPGA. The filter architecture is based on the equi-ripple FIR filter for optimum response both in terms of magnitude and phase. Fig.4.6 shows the 8th order FIR filter architecture proposed for window functions to be

carried out in the analysis of flight vehicle random vibration signals.

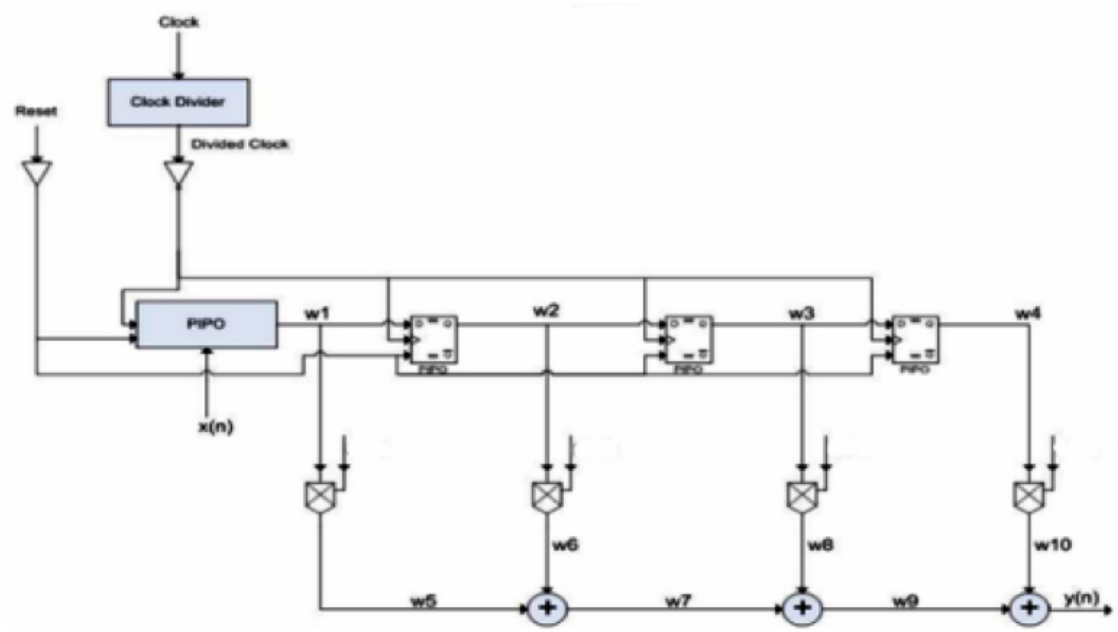

Figure 4.6 FIR Filter Architecture

The magnitude response of this filter is obtained using the MATLAB tool and the plot is presented in Fig.4.7. It may be observed from the plot that the ripples due to sidelobes beyond the stop band of 5 KHz is of the order of -55dB which is well below the required isolation of -40dB for avionics applications.

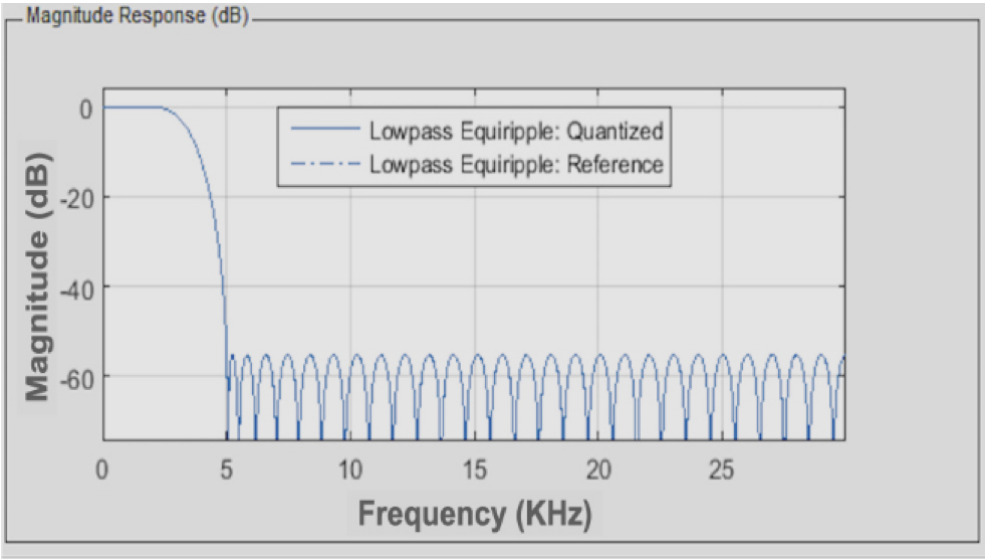

Figure 4.7 Magnitude Response of Equi-ripple Filter

4.3 Overlapping

Overlapping is used to include more original data in the PSD and to generate more Degrees of Freedom (DoF) within a period of time. With 0% overlapping, each frame of data depending on the number of data points are fully separated and there is no common data between the adjacent data frames. As the overlapping percentage increases, that much data points are reused in the next frame forming the number of data points (See Fig 4.8). This overlapping procedure can be applied at any point of the process after digitisation. Each frame of data is stored in the memory for use in the next frame to overlap. Thus the memory requirements are based on the overlapping options needed for the particular application.

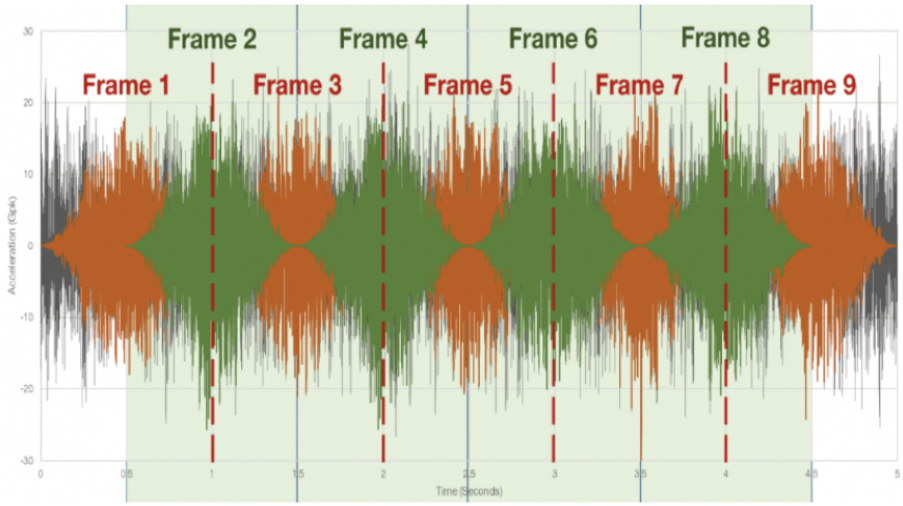

Figure 4.8 Overlapping of Data Points

The proposed scheme for implementing the overlapping function is shown in Fig 4.9. The architecture for implementing the proposed overlapping scheme is developed with an option to select the percentage of overlap of 0%, 25% and 50%. Hardware-in-Loop-Simulation test results show that 25% overlapping of the data provides better results with the proposed windowing techniques [88]. The same overlapping % is to be considered for on-board implementation based on the ground experimental results of the proposed architecture.

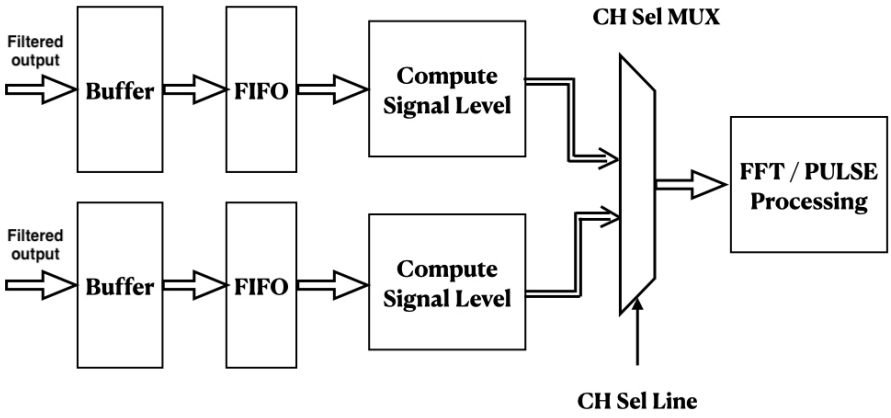

Figure 4.9 Data Flow architecture for Overlapping

4.4 Fast Fourier Transformation

Discrete Fourier transform of a sequence 'x' of N terms x(0),x(1),...,x(N -1), can be defined as

$$X(p) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{np}{N}} \tag{4.8}$$

The N terms of x(n) are N samples of a sampled analog signal which can be expressed as x(n) = x (nTs), where Ts is the sampling period and the N terms of X(p) correspond to a frequency approximation of the Fourier transform of this signal to the N points. The above equation can be written as

$$X(p) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)p} \tag{4.9}$$

$$X(p) = A_p + W_N^p * B_p \tag{4.10}$$

where Ap , W_N^p and Bp are

$$\mathrm{A}_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N2}(n)p}$$

$$\mathrm{B}_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1}\ e^{-j\frac{2\pi}{N2}(n)p}$$

$$\mathrm{W}_N^p = e^{-j\frac{2\pi}{N}p}$$

Since the DFT is considered as periodic, X(p) can be written as

$$X(p+\frac{N}{2}) \;=\; A_p \;\; - \;\; W_N^p \;*\; B_p \tag{4.11}$$

The Eq.4.8 – Eq.4.11 define the basic structure of the FFT algorithm in the butterfly form and architecture for the same is developed using Radix-2 butterflies (See Fig. 4.10). Typically, FFT operates over complex numbers and produces complex numbers. The hardware realization of the computation of FFT of complex numbers is referred as complex FFT (CFFT) [36, 76, 77]. Since many of the physical signals are real valued, the Real FFT (RFFT) is important. The frequency spectrum of the RFFT output is Hermitian symmetric as the input samples are real in time domain [78]. Thus, approximately half of the computations are redundant and this property can be exploited to reduce the hardware complexity. The various algorithms for the implementation of FFT and the VLSI architecture for the computation of the Fast Fourier Transformation with Pipelined, Parallel and Memory based architectures are presented in Chapter 2. The pipelined Radix-2 architecture allows more parallelism based on the field requirements along with necessary mathematical calculations for the identification of the dominant frequency for real-time mission critical applications like flight vehicles. The radix-2 architecture and the pipelined streaming I/O scheme for the flow of input data applied to the pipelined architecture designed using Radix-2 butterflies are shown in Fig.4.10 and 4.11.

Fig.4.12 shows the 1024-point FFT developed using memory based pipelined architecture. ADC word is considered to be 14 bit signed 2's compliment number. FFT memory width is 24 bits and 22 bits are needed to fully accommodate bit growth. The system clock rate is set at 50 MHz and the conversion speed is approximately 100 microseconds for 1024-point complex transform. The input signal amplitude is chosen as 1.5V for simulation purpose. In the real scenario, the amplitude of the signal varies from -5V to +5V i.e. 10V p-p.

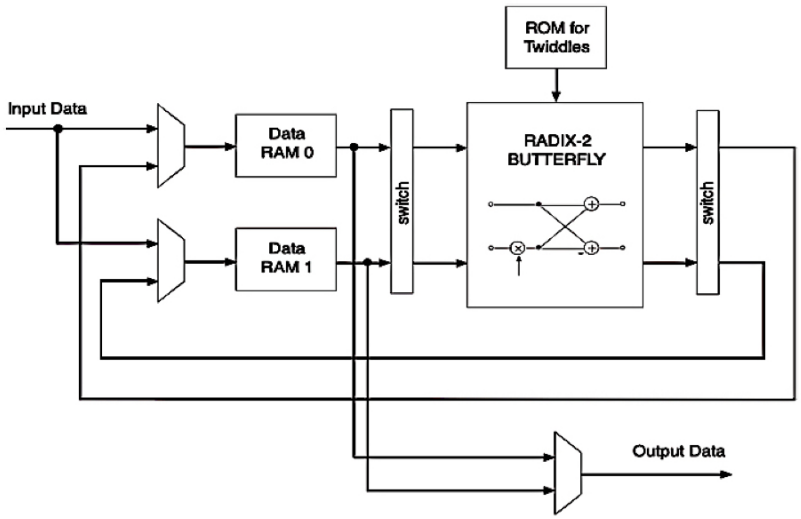

Figure 4.10 Radix-2 Architecture

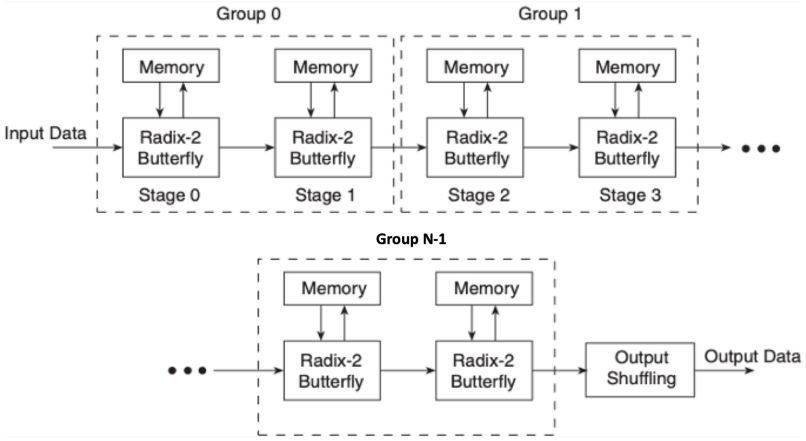

Figure 4.11 Pipelined Streaming I/O Scheme

The proposed FFT architecture is modelled using HDL, simulated and synthesized using Xilinx Design Vivado tools to verify the functionality assuming that the higher-level application logic writes the pre-processed real time random signals into the RAM modules of FFT core. The front end logic of the core handles input data and synchronises with 50 MHz clock. The synchronised data is stored into BRAM module based on the sequence. The sequencer is designed using Moore machine logic, which takes input control signal to initiate the FFT process. This sequencer generates sequence of control signals to store input data, perform butterfly operations sequentially and store the processed data into

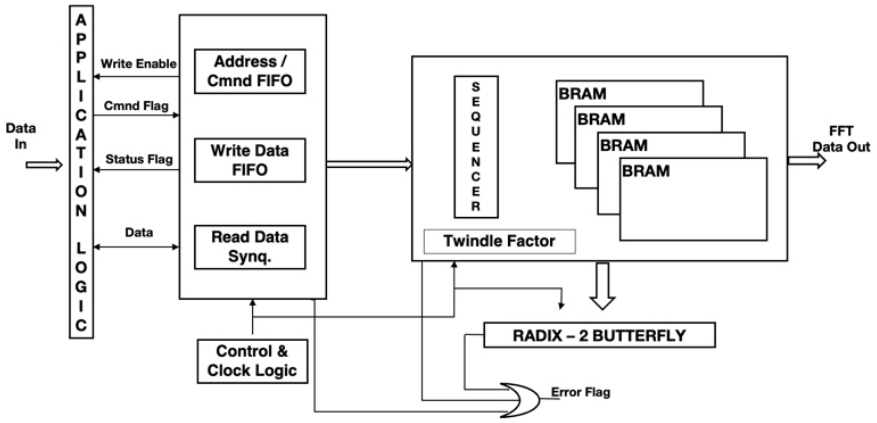

Figure 4.12 Top Level Architecture of FFT Core

another BRAM module. Finally, it generates 'data ready' signal to the other processing modules for further computations.

The functional verification of this FFT core (see Fig.4.12) with 1024-point FFT is performed with varying periodic inputs. To ensure that the signal appears periodic at the boundaries, the signal frequency is selected as 1221 Hz i.e. 20 times 5000/1024. The simulation results for one of the test cases are presented in Fig. 4.13. When multiple sinusoid input with varying amplitude levels are applied to the core as stimuli, the response of FFT to these inputs are stored in RAM modules. The functionality of the FFT core is verified by plotting the results stored in the RAM using Matlab (See Fig.4.14). It is concluded that the FFT core is providing expected results for the given input sequence of multiple sinusoids.

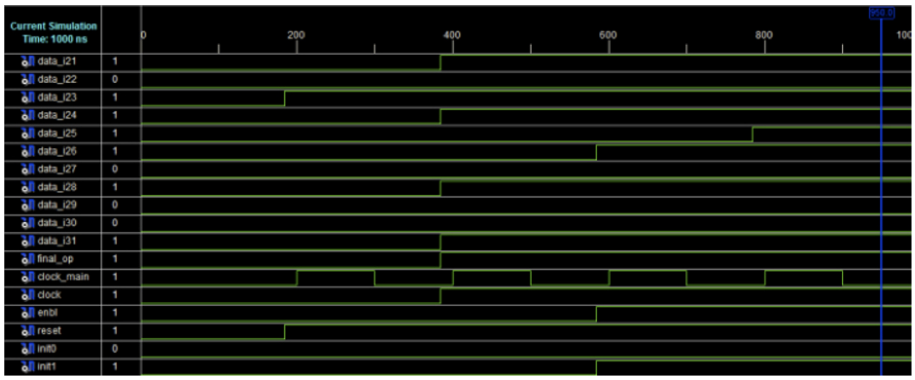

Figure 4.13 Test Bench Periodic Input

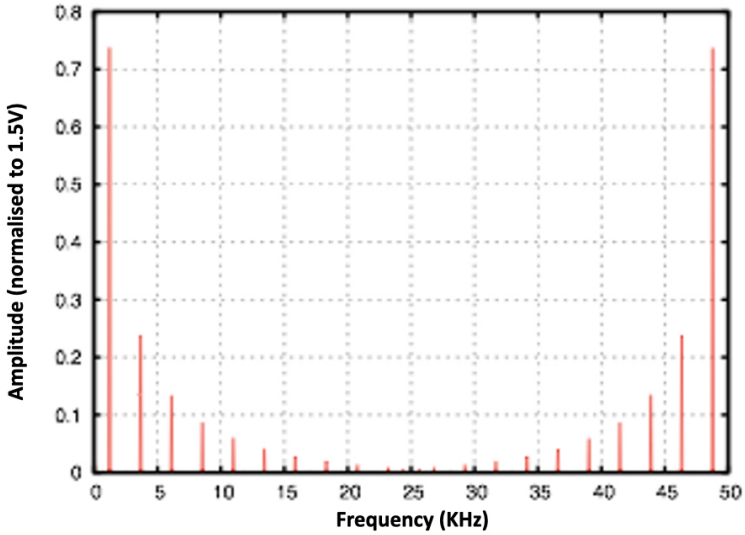

Figure 4.14 Test Bench Output Waveform

4.5 Power Spectral Density

The frequency in the x-axis and the amplitude in the y-axis are normalised by the spectral resolution of the measurement sensitivity and the digitisation resolution. The time history of the random vibration signal measured by an accelerometer and the frequency distribution is illustrated in Fig.4.15

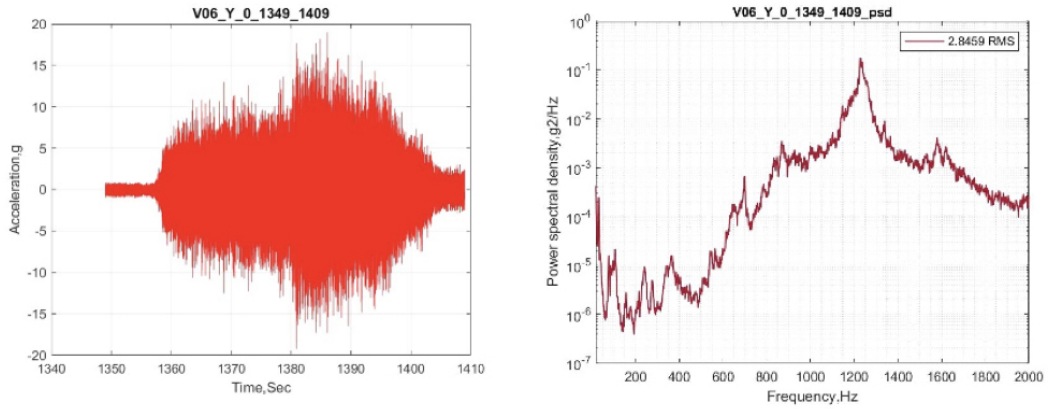

Figure 4.15 (a) Time History (b) Power Spectral Density

In vibration mechanics, the following three methods can mainly be used to calculate the Power Spectral Density for an excitation between -T/2 and +T/2.

- from the autocorrelation function
- by signal filtering with Δf wide filters and
- calculation of the rms value of the filtered signals with the help of Fourier transforms

The most widely used method is using the rms value [73], as the mathematical functions are simpler and data processing in digital mode is easier after domain conversion. The PSD is defined from the Fourier transform of the random vibration $\ddot{x}$(t) sample with T duration as

$$G(f) \;=\; \lim T \to \infty \frac{2}{T} \left| \ddot{X}(f) \right|^2 \tag{4.12}$$

In practice, when the sample duration is finite then,

$$G(f) \;=\; \frac{2}{T} \left| \ddot{X}(f) \right|^2 \tag{4.13}$$

The Fourier transform gives the highest peak amplitude of the output signal from a filter (f, Δf) during the period T. Hence, it is desirable to carry out averaging of several Fourier transforms to improve precision. An averaging module using Goldschmidt algorithm is introduced in the PSD core for storing the calculated PSD data into the block RAM and averaging based on the throughput rate and array search method for finding the Instant Maxima of the PSD data.

4.6 Proposed System Architecture

The system architecture proposed in this work for the frequency domain analysis of flight vehicle random vibration signals employing the modules windowing, overlapping, FFT and PSD Computation presented in the previous sections (4.2-4.5) is shown in Fig 4.16. This architecture is designed by employing pipelined memory based architecture [58, 79]. The pre-processed data from the Analog-to-Digital convertor is fed to the FIR filter for windowing the signals. The stored values in the BRAM are taken as input to the MAC for the frequency domain conversion using FFT, after necessary overlapping of data. Bit rotation is implemented to reverse the output order of the FFT and the output

of the FFT core is used for generating the PSD data and identifying the peak value of Y along with the corresponding X value. The components of the proposed architecture are explained in the following subsections.

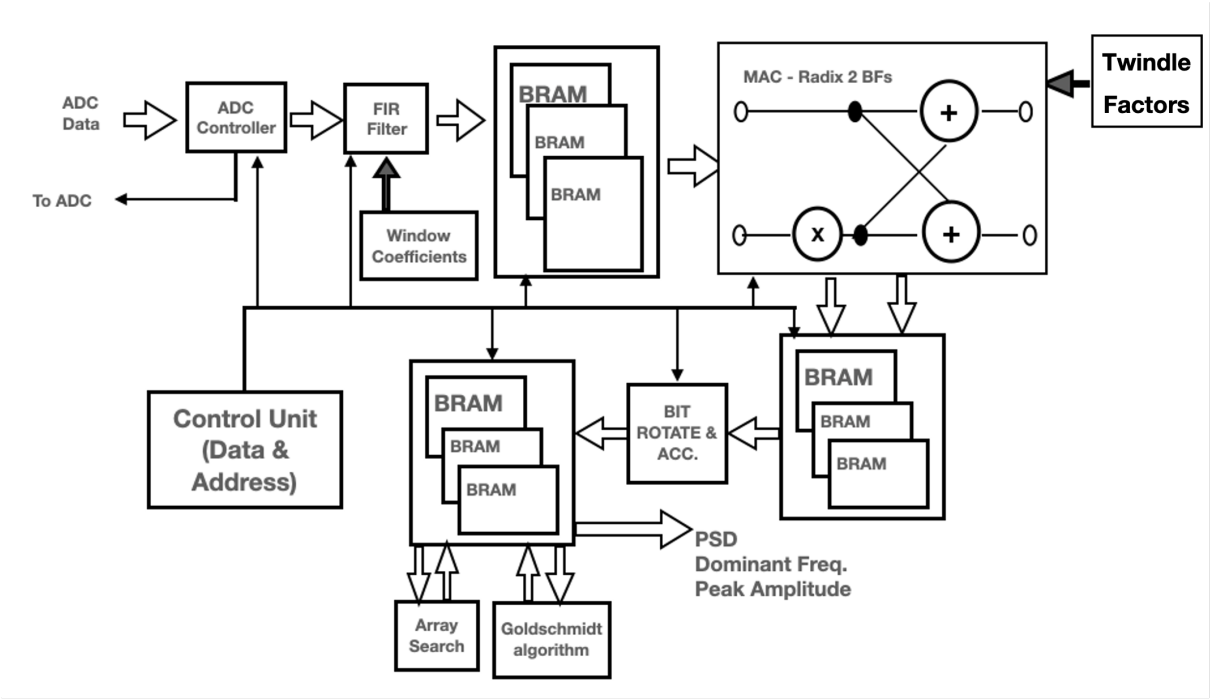

Figure 4.16 Proposed Architecture

ADC Controller: The ADC controller takes ADC data for each channel of data size of 14 bits. It generates control signals for ADC through Serial Peripheral Interface (SPI) and then registers the digital data. For 12 channel data acquisition, 12 ADCs and 12 ADC controllers are considered for the proposed system. The registered ADC data is transferred to the FIR filter.

FIR Filter: The 8th order FIR filter is designed and the coefficients are loaded into the memory. The FIR filter takes the registered ADC data and performs the Windowing function for smoothening of the input digital signals. The processed FIR data are stored in the BRAM and accumulated for further FFT processing. 12 BRAMs are selected with each BRAM of 1Kx16 bits. Based on N_{FFT}, the BRAM size is adjusted for storing the information. The stored BRAM data is further fetched by the FFT processing unit.

Radix-2 BF: This forms the basis of FFT processing unit. One FFT processor is configured to perform the FFT of the input from 12 channels sequentially. The processed data is stored in BRAM for next level of signal processing. The FFT module fetches the data from input BRAM and stores into the output BRAM to optimize the timing issues. The data from the ouput BRAM is processed through Bit Rotate and Accumalator unit for alignment of the output digital data bits. These output bits are stored in another BRAM for further carrying out the mathematical operations of Averaging and Instant Maxima.

Control Unit for Data & Address: The control unit is a finite state sequential machine which controls all the functional modules by issuing appropriate control signals in synchronisation with the clock signal and maintains the timing synchronization between the sub-modules.

Averaging Module: This module is used to compute the average of the PSD array data of FFT output which will be stored in BRAM. It is designed by employing the Gold-Schmidt algorithm which uses an iterative process of repeatedly multiplying both the dividend and divisor by a common factor Fi chosen such that the divisor converges to 1.

Array Search Module: The array search module is to find out the value of X vector corresponding to maxima of Y vector of averaged PSD data.

4.7 Results and Analysis

The proposed architecture developed for the onboard frequency domain analysis of the flight random vibration signals is modelled using Verilog HDL and synthesised with Vivado tools targeting to implement using Kintex-7 XC7K410T FPGA device. The screen shot of the device utilisation is presented in Fig 4.17.

The proposed architecture is simulated with a synthetic signal of 300 Hz frequency

Resource	Estimation	Available	Utilization %
LUT	8653	254200	3.40
LUTRAM	4986	90600	5.50
FF	10761	508400	2.12
BRAM	496	795	62.39
DSP	170	1540	11.04
IO	21	400	5.25
BUFG	6	32	18.75
MMCM	1	10	10.00

Figure 4.17 Device utilisation Screenshot

padded with Gaussian white noise. The number of FFT points is set at 1024 and the overlapping percentage is selected as 25%. The test input sinusoidal signal applied as a stimuli is shown in the screenshot of behavioural simulation window presented in Fig. 4.18. The screenshot of input and output array of a single sinusoid reported after post-synthesis is presented in Fig.4.19. The level of input sinusoidal signal is varied in range from -10 dBm up to -40 dBm with a step of -5dBm for verification of the sensitivity.

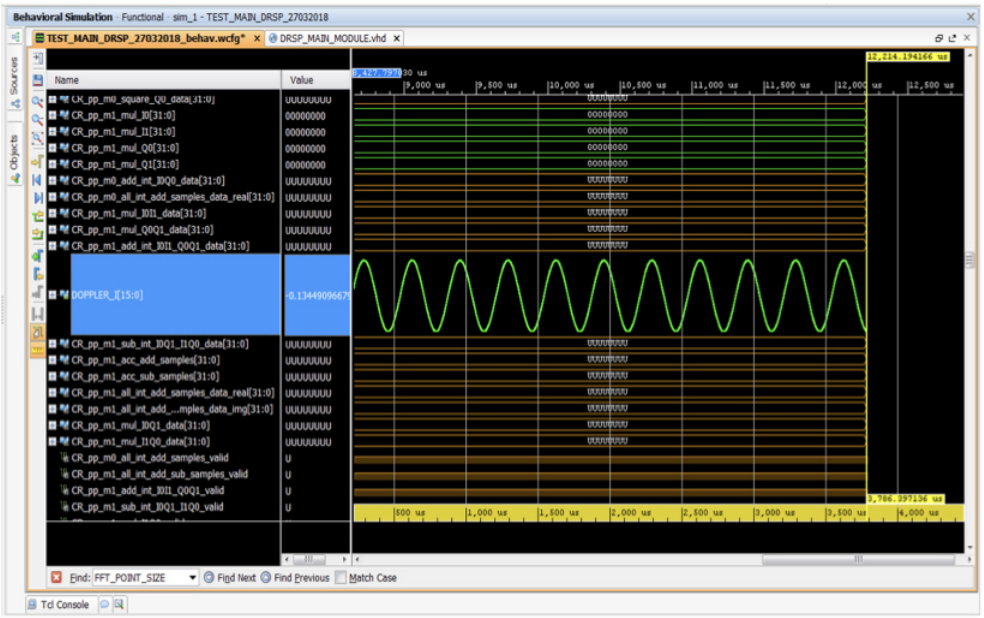

Figure 4.18 Sinusoidal Signal Test Input

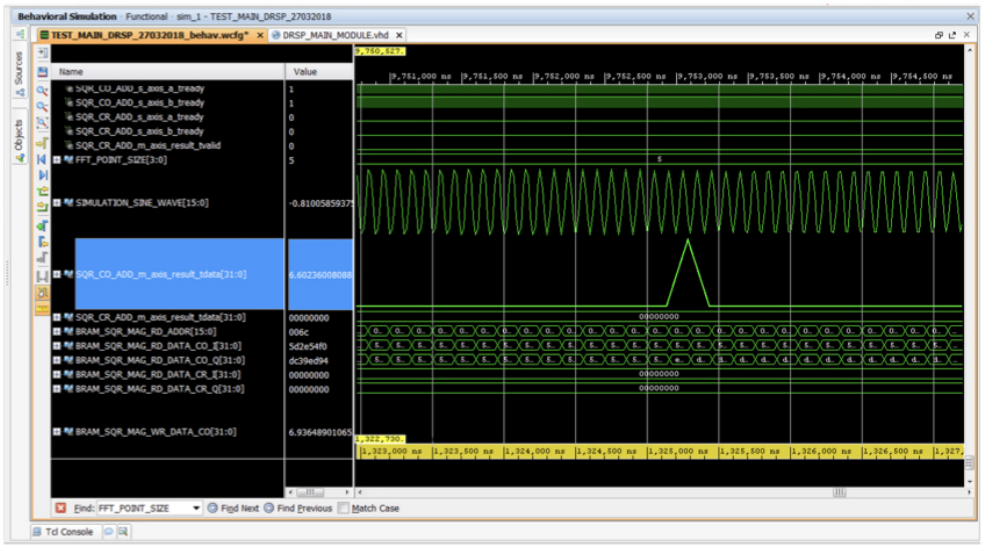

Figure 4.19 Input and Output Array for Summated Sinusoid

The screenshot of input and output array of the summative sinusoid with White Gaussian noise observed after behavioural simulation is presented in Fig. 4.20.

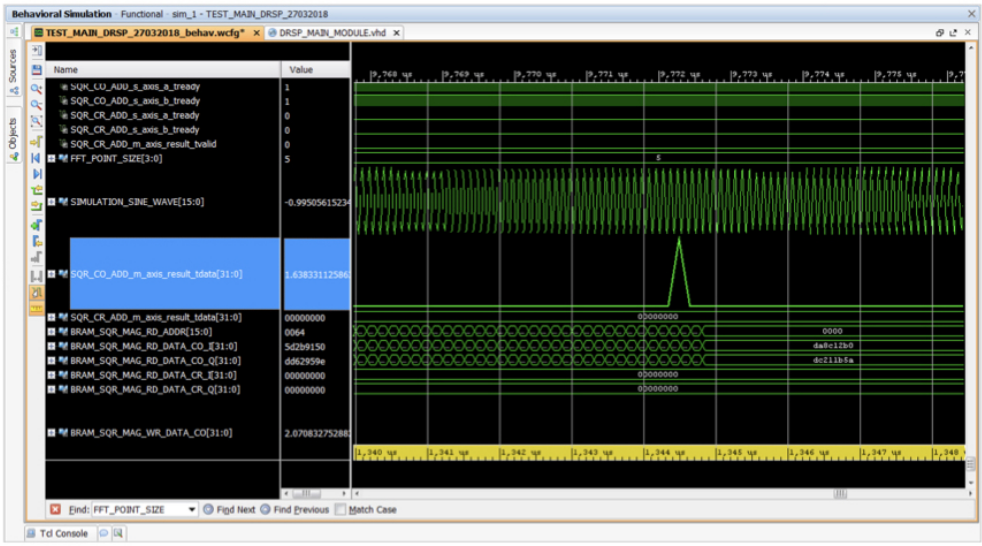

Figure 4.20 Input and Output Array for Summated Sinusoid with Noise

The performance of the proposed architecture is verified based on the post-synthesis functional simulation results. It is evident from the simulation results shown in Fig. 4.19 & 4.20 that the dominant frequency of the input noisy signal is clearly identified by the proposed architecture at 300 Hz. Further, the proposed architecture is validated with the

real time signal from the field sensor input added with predefined noise levels. The field level test setup is prepared to feed the doppler signal of 300 Hz frequency with varying gaussian noise signals.

FFT processed data pumping out is of the form

$$y\ (n) = |\ real\ \ (FFT\,(x_{m,n}))|^2\ +\ |\ imag\ \ (FFT\,(x_{m,n}))|^2 \tag{4.14}$$

where $x_{m,n}$ is the time series vector of i_{th} range bin and j_{th} FFT frame. In order to translate the power level of the signal at the ADC input to the digital signal peak value, a scaling factor has to be included in the power computation as,

$$Power\ \ (dBm) =\ 20\log_{10}(\alpha\ \frac{\sqrt{y\,(n)}}{N_{FFT}\ 2^{16}}) +\ c \tag{4.15}$$

In order to compute a general scaling factor, a constant power level of -40dBm is fed to the INref1 and INref2 channels for different N_{FFT} values. The values chosen for the verification methodology are 128 and 256. Peak values from 3 FFT frames are averaged to arrive at an estimate of the scaling factor. The scaling factors noted from the test runs are presented in Table 4.3.

Table 4.3 Scaling Factors for Calibration

N_{FFT} = 128, Input = INref1	Scaling Factor=2.38e-11
N_{FFT} = 256, Input = INref1	Scaling Factor=2.15e-11
N_{FFT} = 256, Input = INref2	Scaling Factor=2.28e-11

The average of these values is computed to estimate the scaling factor which is 2.275e-1. The proposed architecture for processing the input signal with simulated peripheral components is tested with the real-time doppler signal of 300 Hz known frequency with varying input signal level in dynamic range test. The spectrum of the FFT frame from the Power Analyser for 300 Hz input signal with -40dBm noise level is shown in Fig. 4.21.

The proposed architecture is also validated by performing multiple simulation runs with Hardware-in-loop, while varying the no. of FFT points between 256 and 2048 with

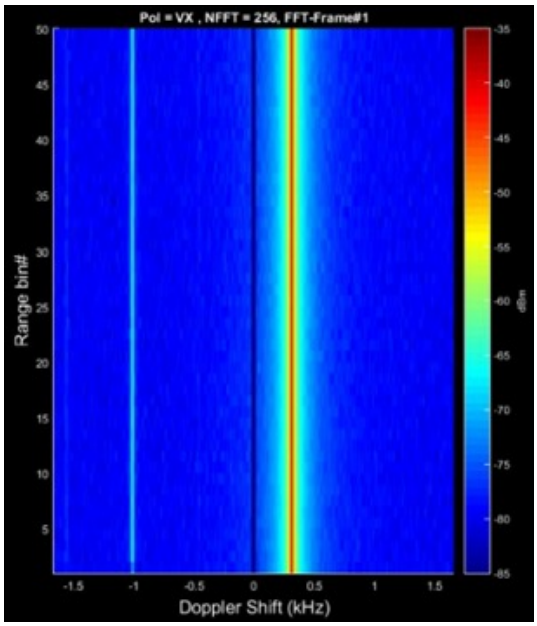

Figure 4.21 Power Analyser Spectrum

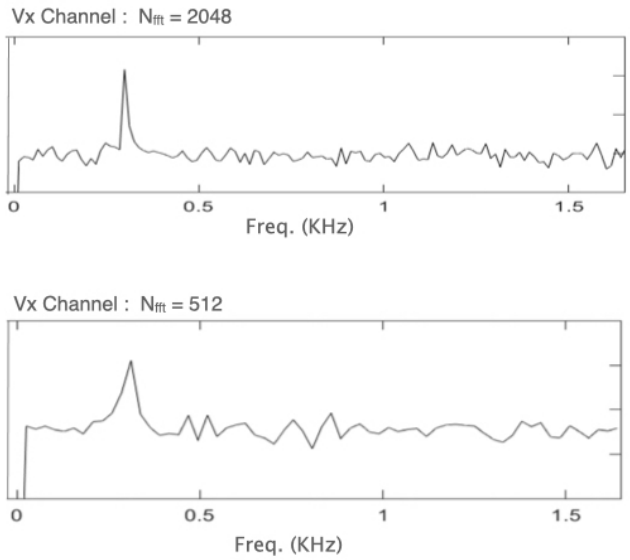

Figure 4.22 Vx Channel Spectrum for N_{FFT}=512 and N_{FFT}=2048

same input signals. The output channel spectrum with number of FFT points N_{FFT}=512 and N_{FFT}=2048 are shown in Fig. 4.22. This architecture can provide better isolation upto -70dBm of the input signal beyond which the signal is merging with the noise floor. The output of the unit under test designed with 1024 point FFT for varying sinusoidal input signal levels is presented in Fig.4.23. It can be seen from the field test output plots that the spot of frequency of 300 Hz is clearly identified. The signal level upto -78.84 dBm is clearly identified (See Fig.4.23-pink colour plot), beyond which the signal is merging with the noise floor, even though the requirement is for isolation upto -40dBm. It is also observed from the field test conducted by varying the number of FFT points that the noise floor levels are improving approximately by -3dBm as the number of FFT points are doubled.

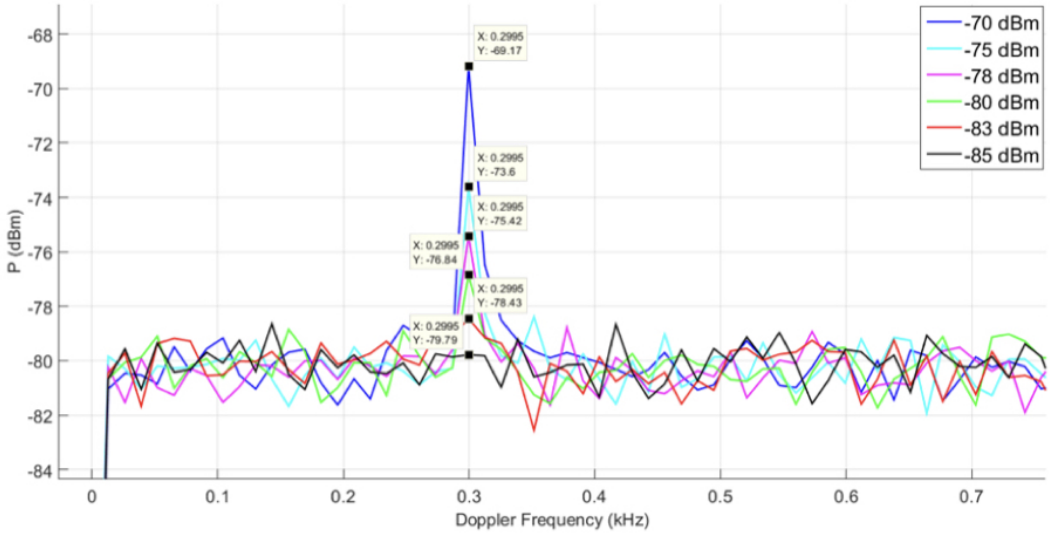

Figure 4.23 Field Test Signal with 1024 Point FFT and 25% Overlapping

4.8 Conclusion

An Architecture is proposed for the frequency domain analysis of the random vibration signal. The proposed system is modelled using HDL and simulated to verify the functionality. The RTL netlist is synthesized using Xilinx Design Vivado tools and the resource utilisation is tabulated. The performance of the proposed architecture is evaluated with a known signal added with Gaussian white noise. The General System Specification requirement for the handling of random vibration signals during flight is limited to

-40dBm isolation of the signal noises. It can be seen from the field test output plots that the spot of frequency of 300 Hz is clearly identified. Also, the isolation is provided up to -70dBm in the worst case. The update rate of the output PSD data array is at about 1.77 msec with 25% overlapping of the input data. Further, the proposed architecture is tested with random signals of different and multiple frequency peaks to prove the consistency. The same can be deployed in on-board applications for the identification of the dominant frequency of the random vibration signals to decide filter co-efficients. A hardware with Xilinx Kintex-7 FPGA is designed with necessary pre and post processing electronics. The architecture discussed here is ported onto a hardware to verify the functionality with real-time accelerometer inputs. The design of the hardware, configuration in flight vehicle avionics systems and the field test results are presented in the next chapter.

Chapter 5

System Design and Field Verification

The architecture proposed for the frequency domain analysis of flight random vibration signal and implementation on Kintex-7 FPGA board are presented in Chapter 4. As the results are satisfactory, a dedicated hardware unit is designed to suit the existing flight vehicle avionics configuration and connectivity with necessary pre-processing and post processing electronics for signal analysis. The basics of the avionics connectivity and the interface requirements needed for the proposed hardware are presented with the field test results.

5.1 Flight Vehicle Configuration

The configuration and design of the Launch vehicle depends on the application, purpose and many other parameters. Major factors contributing to the design and selection of the launch vehicles are

- Range
- Type of Payload
- Launch platform
- User limitations

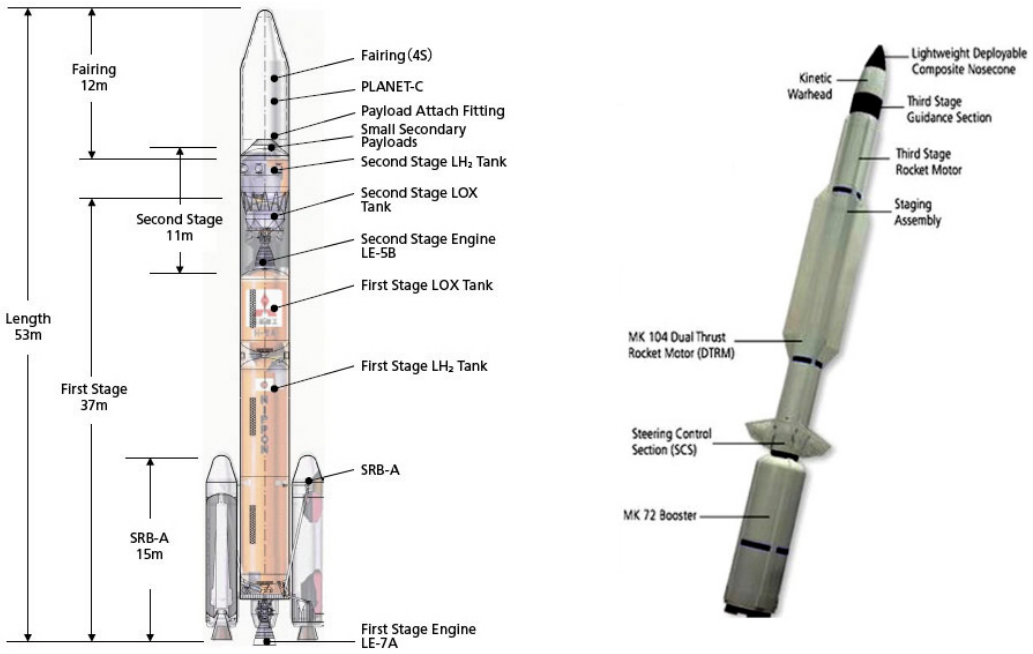

Figure 5.1 Typical Flight Vehicle

Based on the above factors, the flight vehicle can be single staged and multi-staged vehicle (See Fig.5.1) and the avionics sub-systems integrated onto the launch vehicle will vary depending on the application. A general configuration of a flight vehicle avionic sub-systems with the proposed Vibration Signal Processing Unit (VSPU) and the interface are shown in Fig 5.2. There are two nodes in On-Board-Computer (OBC), namely, one for Avionic sub-system interface in the vehicle and the other one for Ground Checkout Console for functionality checks. The sub-systems are interfaced with OBC through MIL STD avionics 1553B bus. Telemetry System is used for transmitting the flight sub-system data to Ground for Real-time tracking and post flight data analysis.

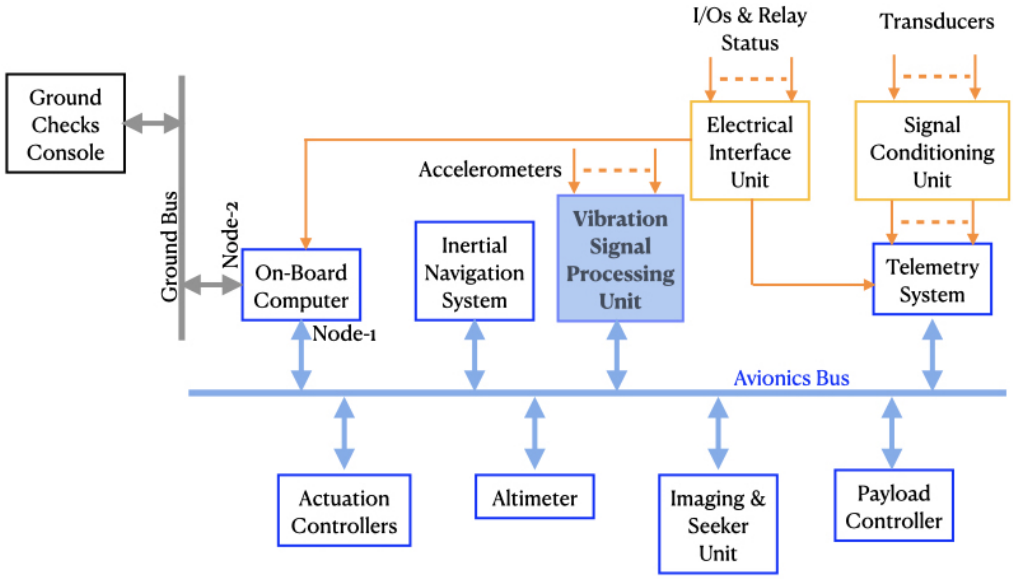

Figure 5.2 Proposed Flight Vehicle Avionics Configuration

5.2 Realisation of Vibration Signal Processing Unit

Vibration Signal Processing Unit (VSPU) is to monitor the on-board vibration during the flight test. These vibration signals are of different bandwidth based on the mounting location and measurement requirements. Both the Mode-shape and Spectral vibration signals are to be acquired through suitable vibration sensors on-board, processed and analysed in frequency domain. The Power spectral Density data are to be telemetered to ground for further analysis and use. Also, the unit need to have the interface provision to receive data through avionics bus in digital format and process the same. The system will have improved capabilities to process more number of channels in a single card. Processing the vibration data on-board gives reduced data rate without compromising on the measurement accuracy and rate. Reducing the power requirement and the weight of the stand-alone unit is the other objective during the design of the system to cater to the flight vehicle limitation on space and power optimisation.

5.2.1 System Specifications

The Vibration Signal Processing Unit will receive the input from the Integrated Circuit Piezoelectric (ICP) accelerometers and these random vibration signals are to be pre-processed before converting into the frequency domain. Input section consists of the anti-aliasing filter followed by analog to digital converter. The anti-aliasing filter is tuned for respective channels viz. Mode shape and Spectral. The frequency range of the mode-shape sensor is 0.1- 200 Hz and the spectral sensor is 20 - 2000 Hz. The sampling frequency for A-D conversion is 2K for Mode shape and 8K for Spectral channels. FPGA will process the vibration channels data in real time and transfers data to telemetry system. FFT is performed on a block of samples for computation of PSD and identification of dominant frequency. FFT output is followed by calculation of power spectral density (PSD). The total process consists of

1. Windowing with Hann/ Hamming / Blackman-Harris window.
2. Overlapping with 0 / 25 / 50 % of data and averaging.
3. Fast Fourier Transformation with Cooley-Turkey algorithm
4. Computation of PSD and averaging
5. Identification of dominant frequency

Antialiasing filters are implemented using 8th order switched capacitor filter with cutoff frequency programmable with EPLD. To increase the system reliability, hardware watch dog timers are incorporated. An EEPROM is provided for storing the programmable parameters and necessary write-protect circuitry is designed to avoid inadvertent writes. Separate memory module is incorporated to store the power spectral density for each cycle. The cycle duration is about 10 msec and memory is sufficient to store all the designed channels for a max duration of 1000 secs. This separate memory module and interface are required only during the ground functionality and validation tests. The specifications for the proposed hardware to design Vibration Signal Processor Unit (see Fig.5.2) are

Vibration sensors	: 12 channels.
Sensor type	: ICP Accelerometers
Pre-Processing	: Signal Conditioning using Discrete components
Analysis Frequency Range	: 2 KHz
Dynamic Range	: 40 db
Output data Rate	: 10 spectrum/ Channel/ sec
Processing	: Computation of N point FFT and PSD
Output Interface	: (i) 1553B / AFDX for Data.
	: (ii) RS422 for configuration / programming.
Programmable Parameters	: (i) Sampling rate
	: (ii) No of FFT points
	: (iii) Overlapping percentage
	: ((iv) Window type

Fig.5.3 shows the block diagram of VSPU and Table5.1 presents the list of different resources required for VSPU. The output of the proposed hardware are Channel no., Power spectral density and Dominant frequency value represented using 16 bits.

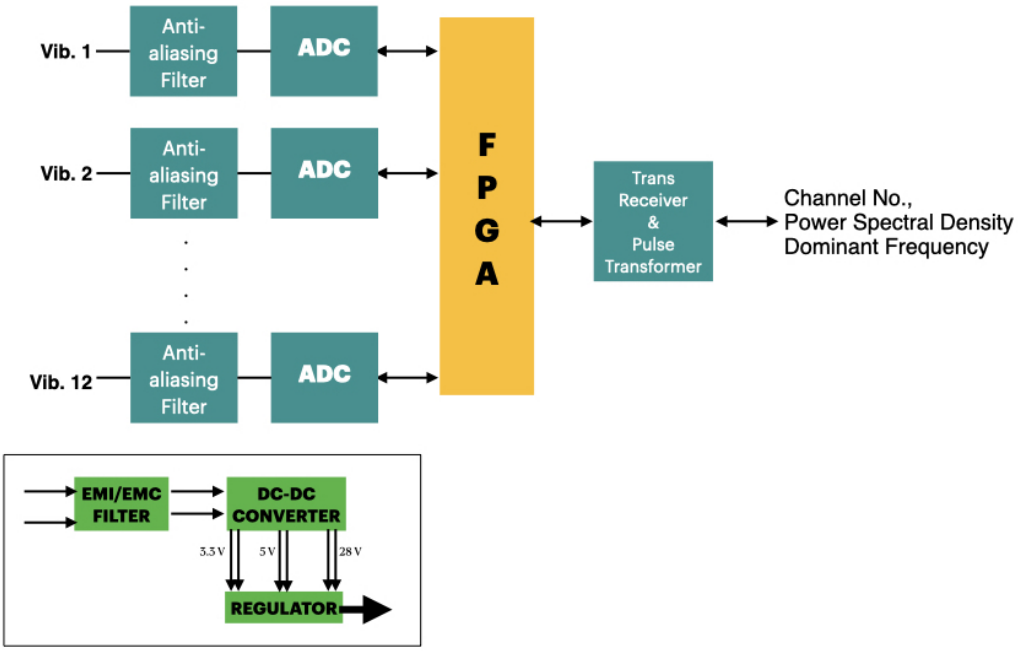

Figure 5.3 Proposed Hardware Block Diagram

Table 5.1 List of Resources for Proposed VSPU

Sl. No.	Components	Requirements
1	No. of ADCs	12 nos.
2	ADC Resolution	14 bit
3	FPGA Kintex7	1 no.
4	On board memory	2 GB DDR2
5	Debug Port	RS232
6	Programming	JTAG for FPGA

5.2.2 Implementation of Proposed VSPU

In this chapter, the architecture for random vibration signal processing validated with prototype board in Chapter 4 is ported on to the dedicated hardware with Xylinx Kintex-7 FPGA device for real time processing of the vibration signals. The hardware for VSPU is shown in Fig.5.5 is designed with four PCBs to cater the functions namely, Power Supply, Signal Conditioning, Vibration Signal Processing and PCM Encoding. The encoding process provides outputs in Bi-phasal for Ground Telemetry receiving station. Both raw accelerometer data and the processed data with PSD and instantaneous dominant frequencies are available at the output for further processing. Fig.5.4 shows the image of the Signal Conditioner & Signal Processor PCBs and the fabricated Vibration Signal Processing Unit (VSPU with 4 PCBs) is shown in Fig.5.5.

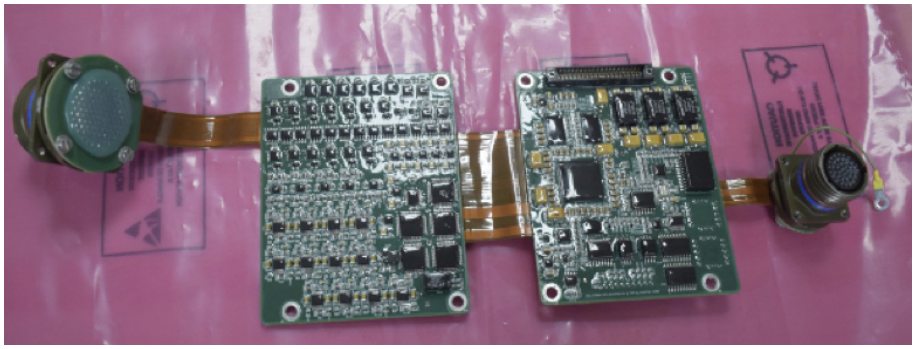

Figure 5.4 Signal Conditioner and Processor

Figure 5.5 Vibration Signal Processing Unit

5.3 Vibration Test Setup

Fig. 5.6 shows the setup used for testing the fabricated VSPU (See Fig.5.5) in the field condition with real-time input and the output of the unit is verified with the reference controller. The reference controller is the same as used in the Hardware-in-Loop simulation of the mathematical model as presented in Chapter 3. The specifications of the reference controller Dewetron Data Analyser (DEWE-4010) is

Throughput	: 45 Mbps
Max Sampling Freq.	: 10 KHz
No. of Points	: 1024
Averating	: Exponential type of 12 lengths
Output Accuracy	: 6-sigma
Windowing	: Selected Hann Window
Overlapping	: 0%

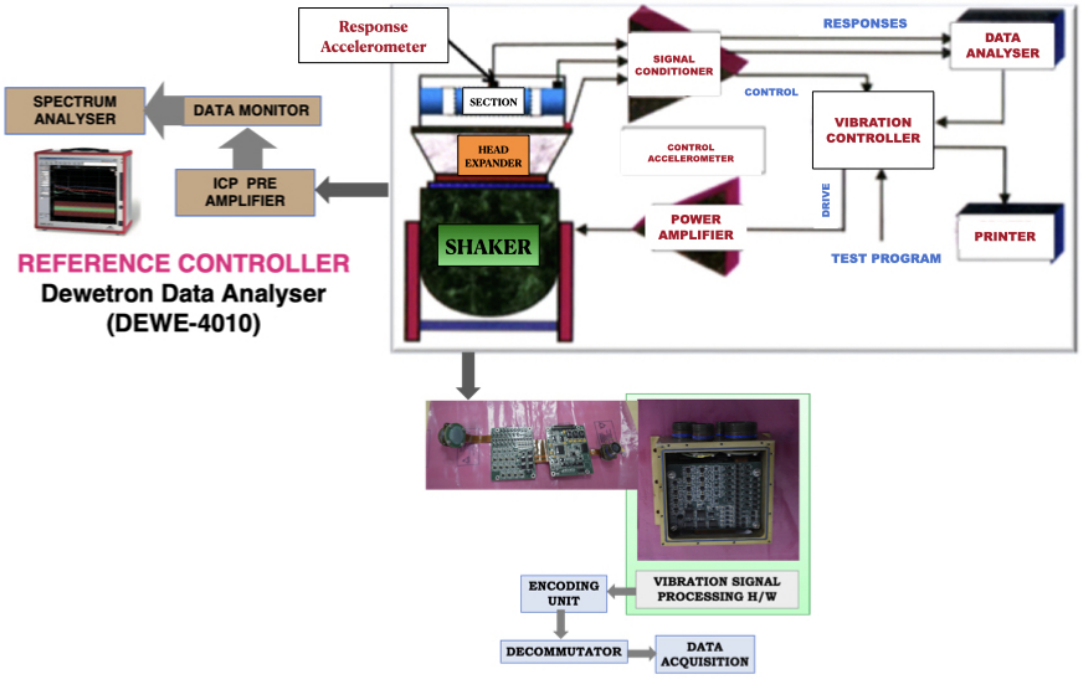

Figure 5.6 Vibration Test Setup

The objective of the test is to condition the input accelerometer signals and extract the Power Spectral Density and the instantaneous dominant frequencies from the spectrum with the Vibration Signal Processing Unit designed for the purpose. The encoder will post the output in Bi-phasal and the data will be received at the Ground Telemetry station. The received spectrum through telemetry is compared with the reference controller output for a definite duration. The spectrum signature and the dominant frequency values are compared and validated for performance. Two different test objects are randomly selected based on the availability of the vibration table and the ongoing dynamic tests of flight sections.

5.4 Results and Analysis

For the field verification of the implemented VSPU (See Fig.5.5), two flight sections are selected during the ongoing tests. The monitoring sensor outputs as per the test plan are interfaced to the hardware and the reference controller. The analysed data from the reference controller and the designed hardware are compared and the observations are as

presented in this section.

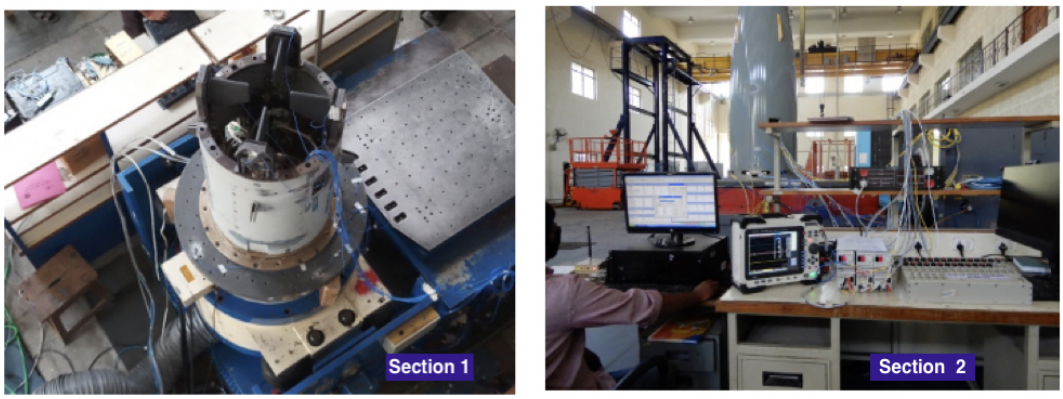

Figure 5.7 Test Objects on Vibration Table

The flight sections integrated with flight components undergo Random Vibration Test for specific level and duration depending on the application before they are flown in actual flight. Vibration sensors are mounted at different locations of the flight section based on the geometry of the section and the sub-systems integrated. The responses from these vibration sensors are monitored to verify the integrity of the flight components. A few monitoring sensors mounted on two different test sections (see Fig.5.7) are selected across the geometry of the test object at different locations. The power spectral density plot from the reference Dewetron controller is compared with the output of the proposed hardware with 2048-point FFT and 512-point FFT for verification of performance of the hardware.

Section 1 Sensor R1

The PSD Plot of Sensor R1 in Section 1 (Fig5.7) is shown in Fig.5.8 and the part of this plot is zoomed in to show the dominant frequency (Fig.5.9). PSD Spectrum recorded from the VSPU is matching with the Reference Controller for both the cases 2048-point FFT and 512-point FFT. However variations in the value of identified dominant frequency is observed between these two cases of N_{FFT} as can be observed from Table 5.2.

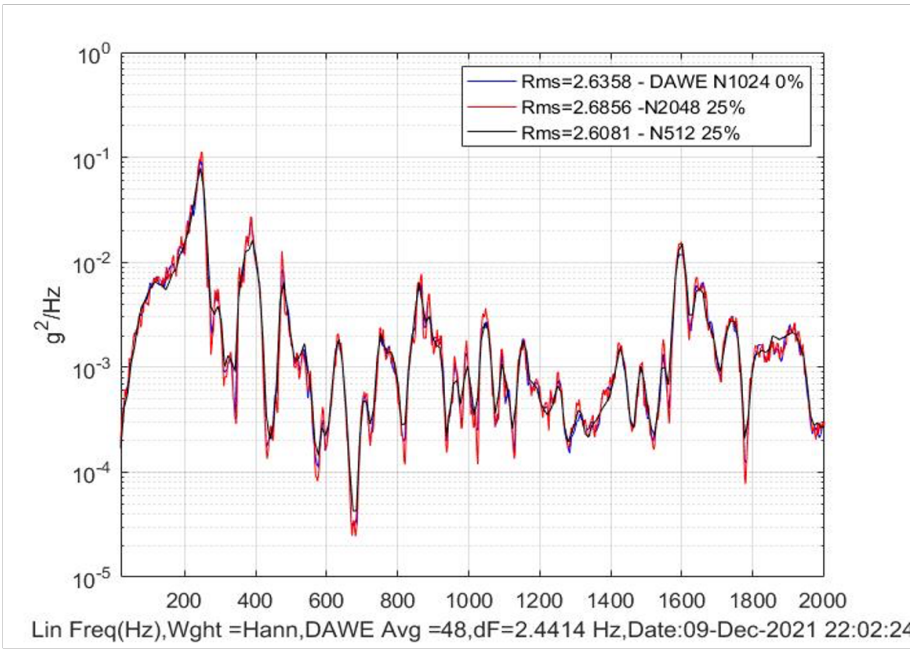

Figure 5.8 PSD Spectrum of Section 1 Sensor R1

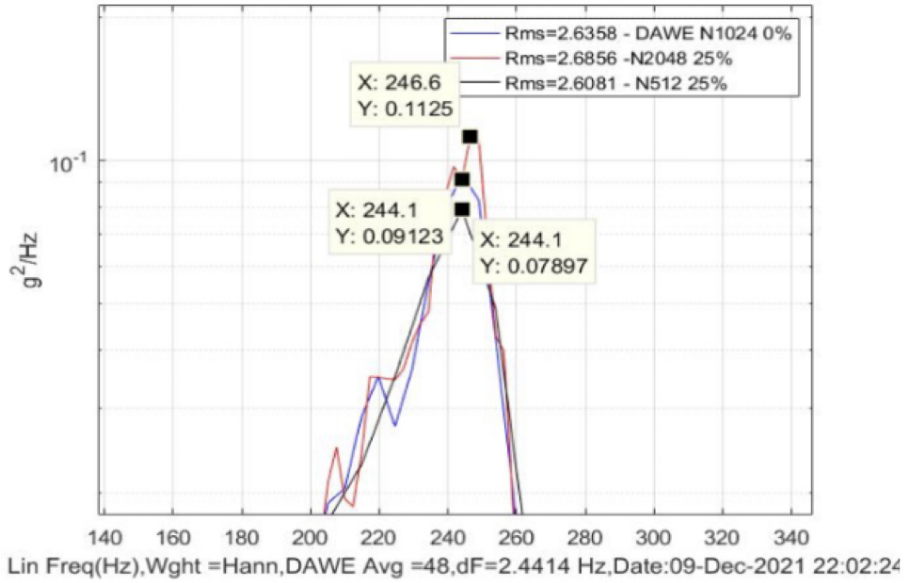

Figure 5.9 PSD Spectrum of Section 1 Sensor R1 Zoomed-In

Table 5.2 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor R1

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	244.1	0.09123
N512,25	244.1	0.7897
N2048, 25	246.6	0.1125

Section 1 Sensor R2

The PSD Plot of Sensor R2 in Section 1 is shown in Fig. 5.10 and the part of this plot to show the dominant frequency is zoomed in (Fig.5.11). It may be analysed from the PSD plot obtained from this that overall g_{rms} computed for 2048-point FFT for the proposed hardware is higher that that of the reference controller. It may also be observed from Table 5.3 that the dominant frequency value reported by the proposed hardware is higher than those of the reference controller.

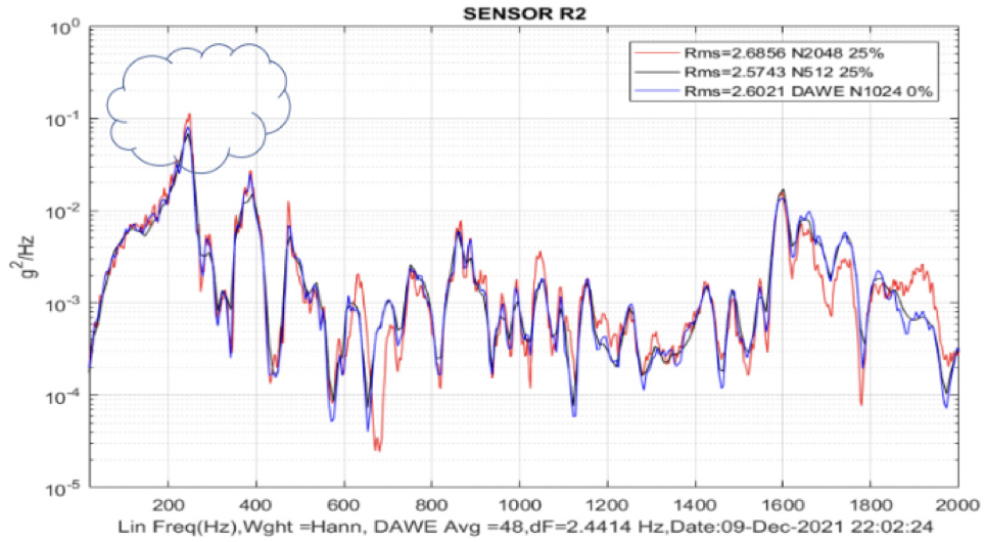

Figure 5.10 PSD Spectrum of Section 1 Sensor R2

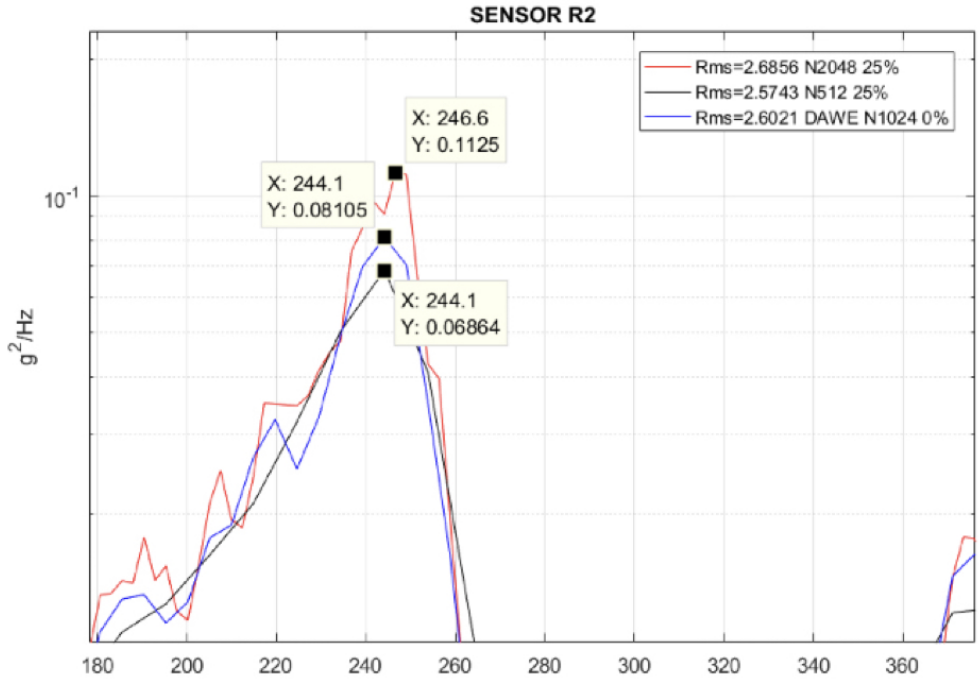

Figure 5.11 PSD Spectrum of Section 1 Sensor R2 Zoomed-In

Table 5.3 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor R2

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	244.1	0.08105
N512,25%	244.1	0.06864
N2048, 25%	246.6	0.1125

Section 1 Sensor R3

The PSD Plot obtained for Sensor R3 in Section 1 is shown in Fig. 5.12 and the part of this plot to show the dominant frequency is zoomed in (Fig.5.13). The signature of spectrum in all the three cases are similar and the identified peaks are matching very closely, which are presented in Table 5.4.

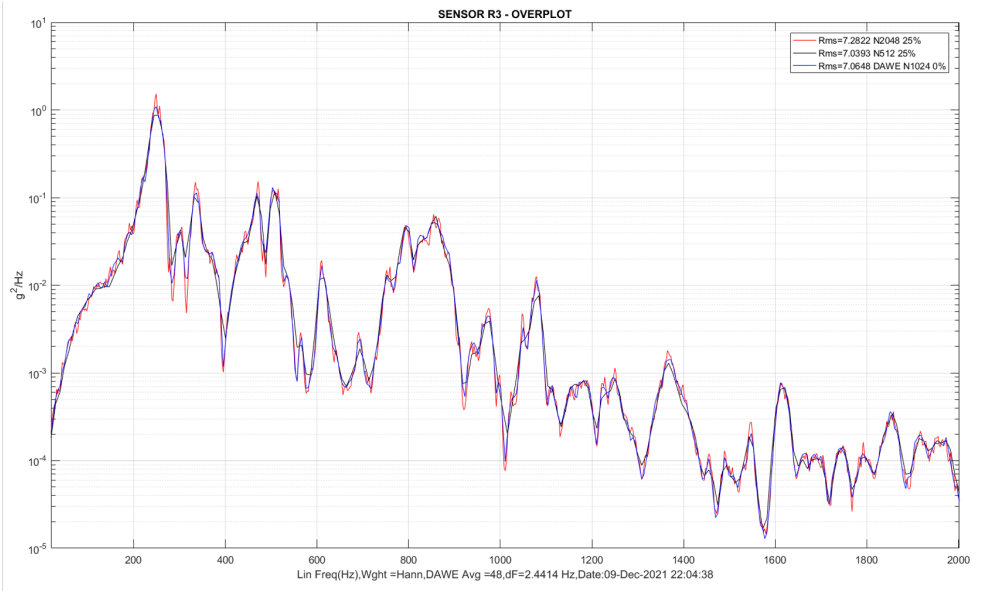

Figure 5.12 PSD Spectrum of Section 1 Sensor R3

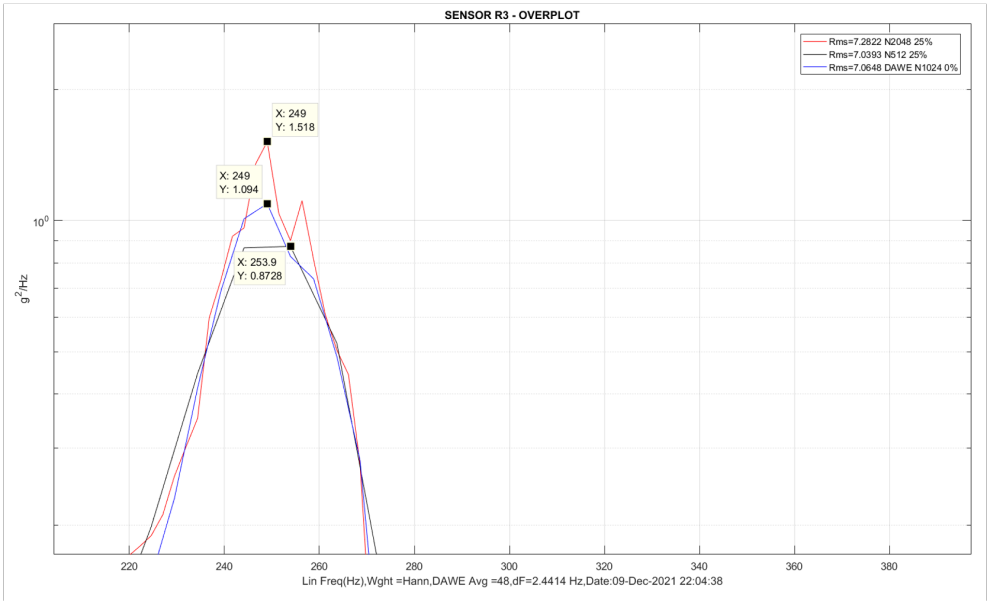

Figure 5.13 PSD Spectrum of Section 1 Sensor R3 Zoomed-In

Table 5.4 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor R3

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	249	1.094
N512,25%	253.9	0.8728
N2048, 25%	249	1.518

Section 1 Sensor S4

The PSD Plot of Sensor S4 in Section 1 is shown in Fig. 5.14. The area of interest is zoomed in plotted in Fig.5.15. More Variations in the PSD Spectrum is observed, though the spectrum signature is similar. The comparison of proposed hardware and reference controller with respect to the observed dominant frequency and the amplitude is presented in Table 5.5.

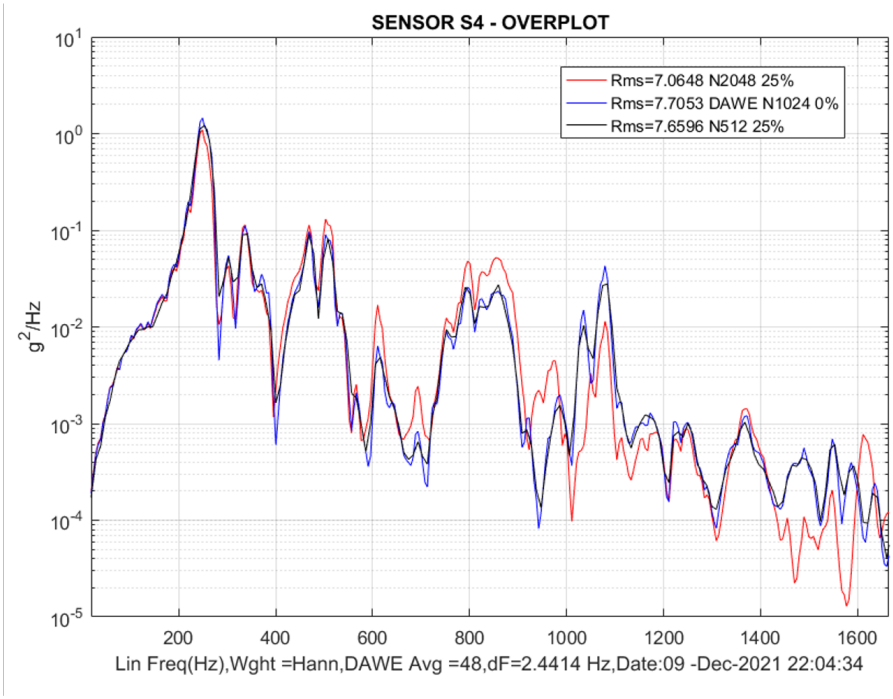

Figure 5.14 PSD Spectrum of Section 1 Sensor R4

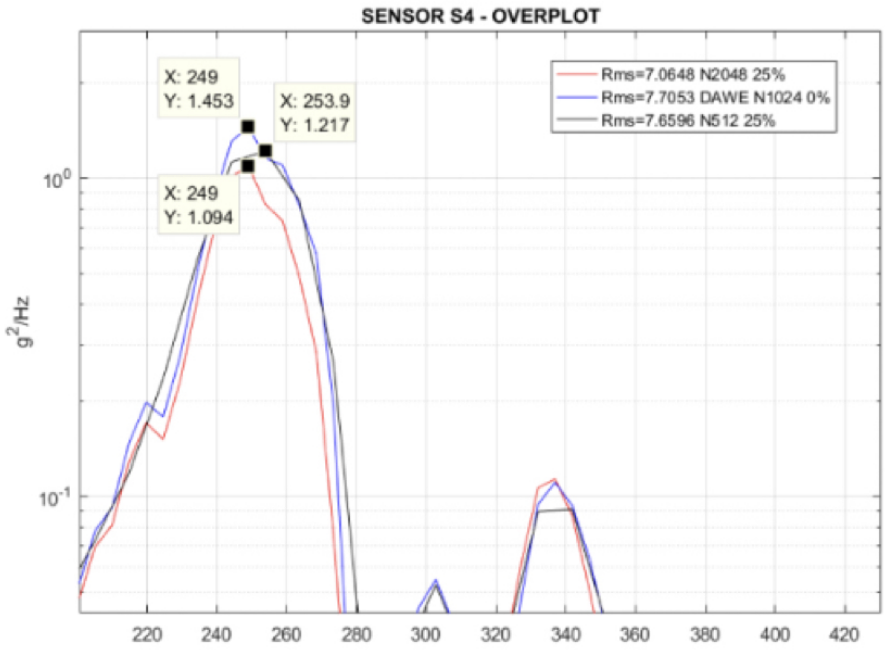

Figure 5.15 PSD Spectrum of Section 1 Sensor R4 Zoomed-In

Table 5.5 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor S4

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	253.9	1.217
N512,25%	249	1.094
N2048, 25%	249	1.453

Section 1 Sensor R6

The PSD Plot obtained for Sensor R6 in Section 1 is shown in Fig. 5.16. The PSD Spectrum signatures from the proposed hardware are very similar to the spectrum obtained from the reference controller. The comparison of proposed hardware and reference controller with respect to the observed dominant frequency and the amplitude is presented in Table 5.6.

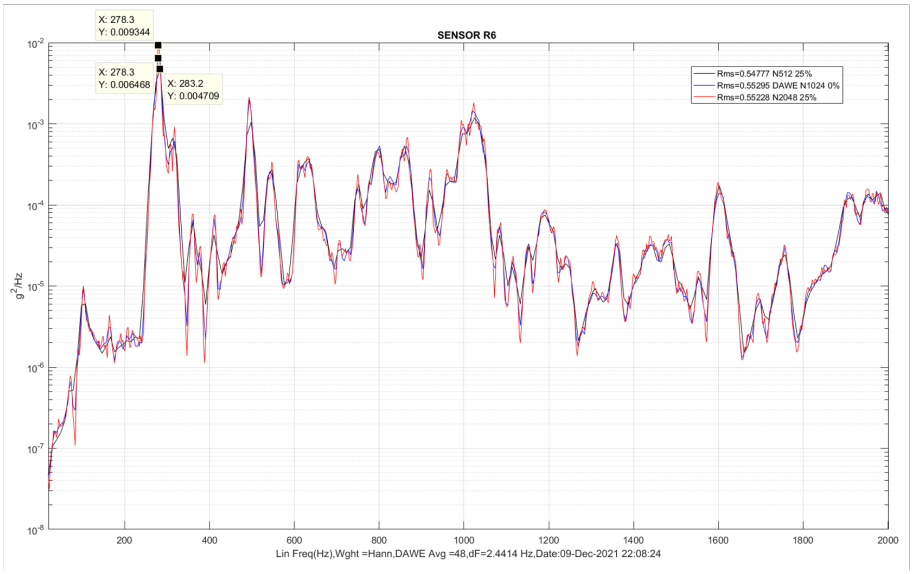

Figure 5.16 PSD Spectrum of Section 1 Sensor R6

Table 5.6 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor R6

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	278.3	0.006468
N512,25%	283.2	0.004709
N2048, 25%	278.3	0.009344

Section 1 Sensor R7

The PSD Plot obtained for Sensor R7 in Section 1 is shown in Fig. 5.17. It is observed form the PSD plots shown in Fig.5.17 obtained for Sensor R7 that the spectrum signatures and the identified dominant frequency values are matching very closely for the proposed hardware and the reference controller. The comparison of proposed hardware and reference controller with respect to the observed dominant frequency and the amplitude is presented in Table 5.7.

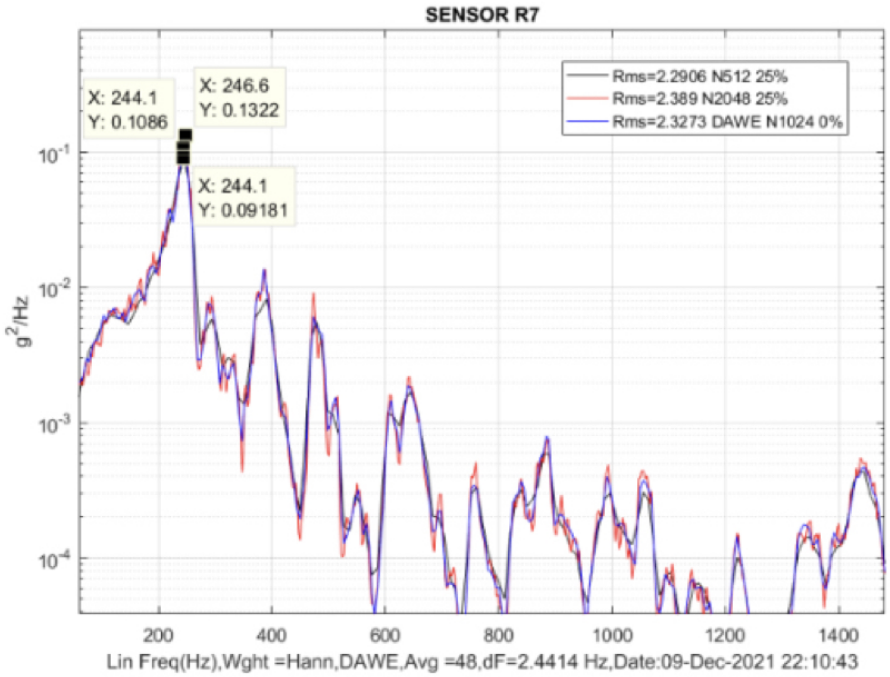

Figure 5.17 PSD Spectrum of Section 1 Sensor R7

Table 5.7 Performance Comparison of Proposed Hardware and Reference Controller - Section 1 Sensor R7

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	244.1	0.1.86
N512,25%	244.1	0.09181
N2048, 25%	246.6	0.1322

Section 1 Sensor R9

In the testing process of proposed hardware, Sensor R9 is considered to demonstrate a specific case of sensor failure. In spite of extremely low output signal values of this sensor as the sensor was made to fail intentionally, the proposed hardware could identify the dominant frequency with the available input. This classic case of sensor failure is to be handled in real time flight environment by adding necessary logic in the Mission software to reject very low value dominant frequency peaks. The dominant frequency and amplitude values observed from the reference controller and the proposed hardware for

the case of sensor failure are presented in Table 5.8. It may be noted that the amplitude values are too low to be considered for any further processing.

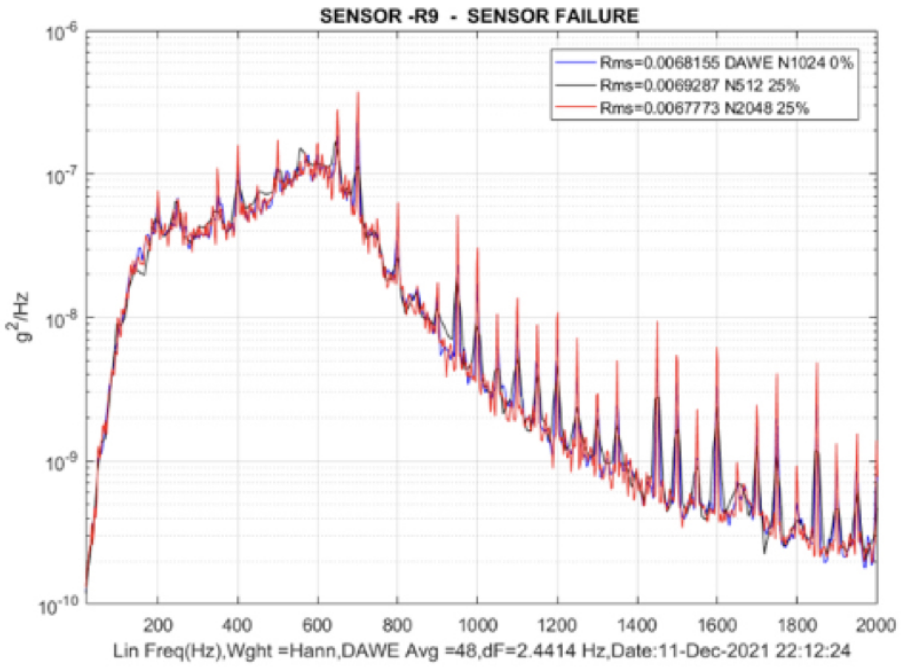

Figure 5.18 PSD Spectrum of Section 1 Sensor R9

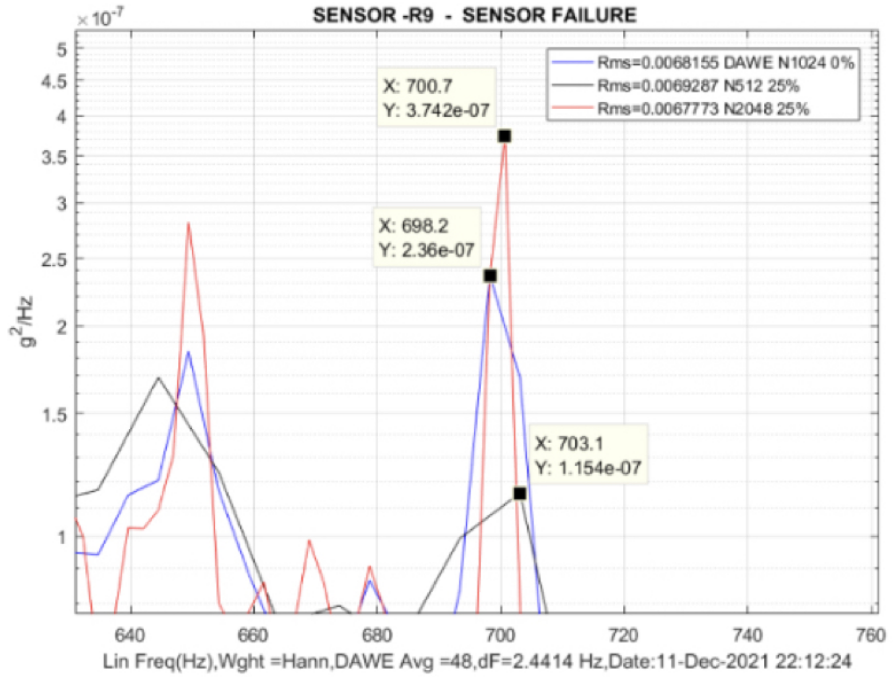

Figure 5.19 PSD Spectrum of Section 1 Sensor R9 Zoomed-In

Table 5.8 Identified Parameters - Section 1 Sensor R9

NaN - Not a Number as the amplitude values are too low to consider.

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	698.2	NaN
N512,25%	703.1	NaN
N2048, 25%	700.7	NaN

Section 2 Sensor R1

The validation process of proposed hardware is carried out with another test section having different shell mode frequency. The PSD spectrums obtained from the proposed hardware is compared with the reference controller. It is observed that the spectrum signature and dominant frequency values obtained from the proposed hardware with FFT configured for N=2048 and N=512 are closely matching with those obtained from the reference controller. Fig.5.20 shows the PSD Spectrums obtained for Section2-Sensor R1 and the dominant frequency values along with the amplitude are presented in Table 5.9.

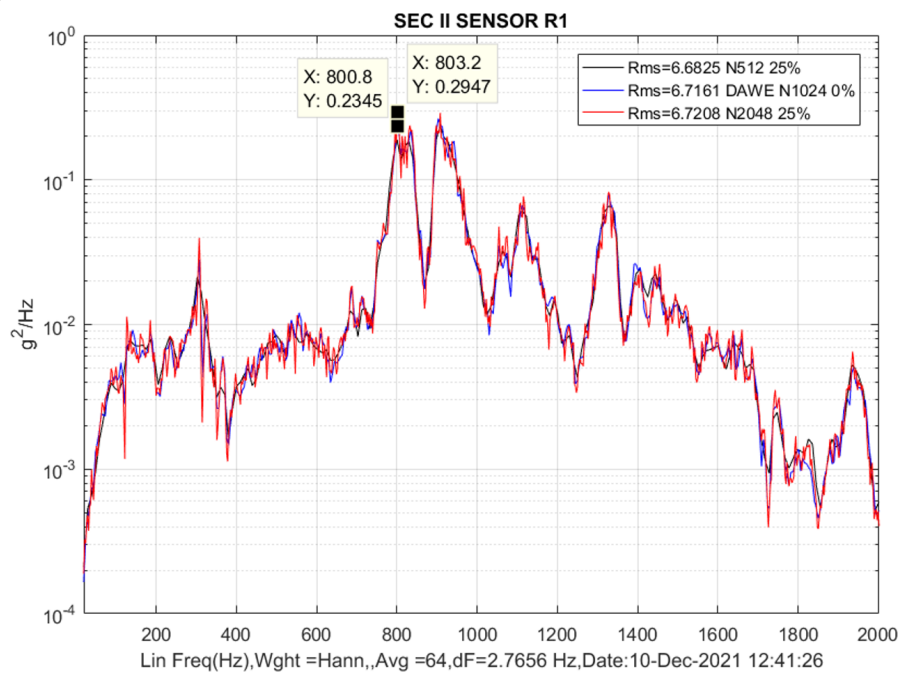

Figure 5.20 PSD Spectrum of Section 2 Sensor R1

Table 5.9 Performance Comparison of Proposed Hardware and Reference Controller - Section 2 Sensor R1

DAQ	X-Freq.(Hz)	Y-g^2/Hz
DEWE	800.8	0.2345
N512,25%	803.2	0.1983
N2048, 25%	803.2	0.2947

Section 2 Sensor R4

Fig.5.21 shows the PSD plot obtained for Section2-Sensor R4 and Fig.5.2 shows zoomed in part of this plot to observe the dominant frequency zone. The performance comparison of the proposed hardware and reference controller with respect to the dominant frequency and the associated amplitude is presented in Table 5.10. The spectrums in all the three cases are similar and the identified frequencies are closely matching.

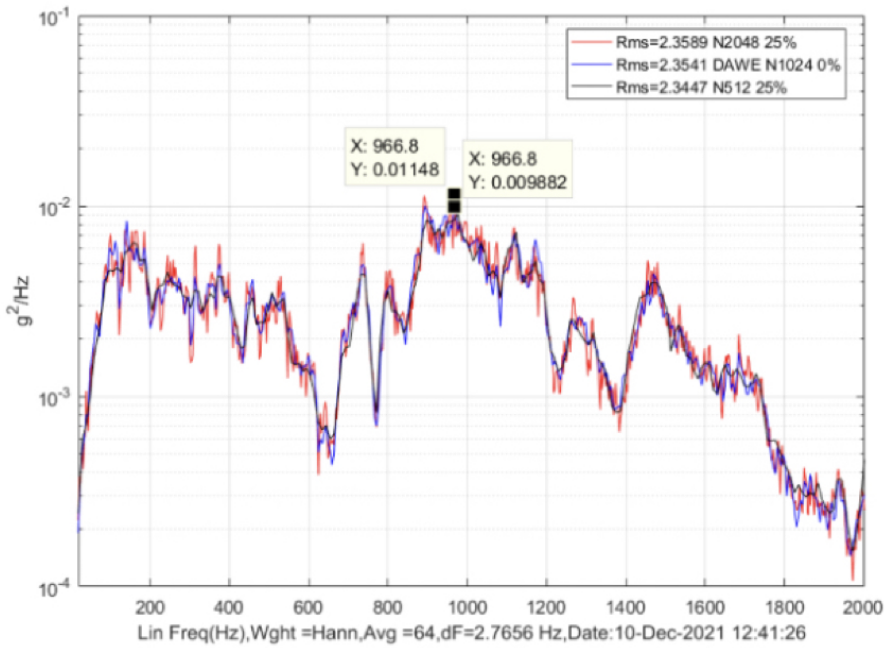

Figure 5.21 PSD Spectrum of Section 2 Sensor R4

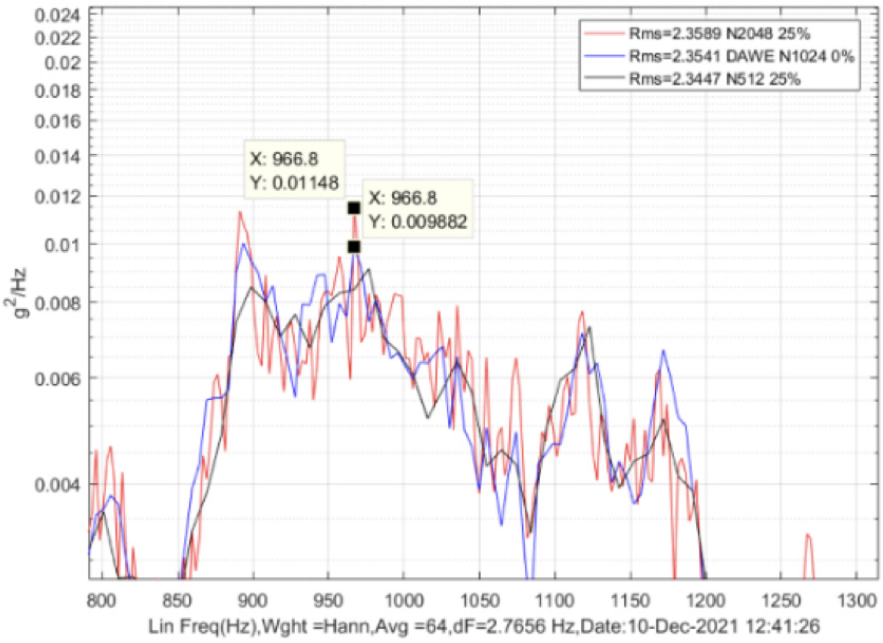

Figure 5.22 PSD Spectrum of Section 2 Sensor R4 Zoomed-In

Table 5.10 Performance Comparison of Proposed Hardware and Reference Controller - Section 2 Sensor R4

DAQ	**X-Freq.(Hz)**	**Y-g^2/Hz**
DEWE	966.8	0.009882
N512,25%	969.2	0.00875
N2048, 25%	966.8	0.01148

For performance evaluation of the proposed hardware, a total of eight real-time sensor outputs are analysed using the proposed hardware and the reference controller. It is observed that the PSD spectrum signatures are matching in all the test cases except minor variations in the dominant frequency values. This variation in frequency values is limited to one Δf, which is the resolution of the FFT. With 2048-point FFT, the identified peaks are more precise and the amplitude levels are a bit lower because of the distribution of the energy content in the nearby frequency components. Table 5.11 presents the summary of the dominant frequency values reported by the proposed hardware and the reference controller for all the test cases and the error observed between the values reported.

Table 5.11 Identified Dominant Frequencies and the Corresponding Errors

DAQ	**S1R1**	**S1R2**	**S1R3**	**S1S4**	**S1R6**	**S1R7**	**S2R1**	**S2R4**
DEWE	244.1	244.1	249.0	253.9	278.3	244.1	903.3	966.8
N512,25%	244.1	244.1	253.9	249	283.2	244.1	903.3	969.2
Error	0	0	4.9	-4.9	4.9	0	0	2.4
N2048,25%	246.6	246.6	249	249	278.3	246.6	908.2	966.8
Error	2.5	2.5	0	-4.9	0	2.5	4.9	0
DEWE - Dewetron Reference Controller and DAQ								
N512, 25% - 512 Point FFT, Hann Window and 25% Overlapping of data in proposed hardware								
N2048, 25% - 2048 Point FFT, Hann Window and 25% Overlapping of data in proposed hardware								

5.5 Conclusion

A dedicated hardware is designed and realised by employing the proposed architecture along with the required pre- and post-processing electronics. This hardware is tested for functionality and performance with the inputs from the accelerometers mounted in the flight sections. These sections undergo dynamic testing by mounting them on the vibration table and tested for the random vibration over the frequency range of 20-2000 Hz.

It is observed from the field test results that the spectrum signatures are closely matching, except a small variation in the amplitude to the tune of 0.00102 g2/Hz even when two dominant frequencies are at close proximity. The maximum difference between the reference and estimated dominant frequency is 4.9 Hz amounting to 1.9% of reference value and 0.24% in full scale range of 2000 Hz. For all other test cases, the accuracy of estimation over the full-scale range is less than 0.2% and over the reference value is less than 2%, whereas the resolution (Δf) of 1024-point FFT over the range is 1.93

Hz. These values are well within the acceptable limit of the mission design requirement. The comparison of analysis results with standard reference processor are presented. The results are satisfactory in comparison with the reference values. With the identification of instantaneous dominant frequency, the same to be posted to the OBC for utilisation in auto-pilot algorithm to dynamically tune the notch filter. The conventional method of auto-pilot algorithm with Notch filter to avoid control-structure interaction and the proposed methodology to dynamically tune the same are in next chapter.

Chapter 6

Auto-Pilot and Dynamic Tuning of Notch Filter

The flight vehicle has to reach the intended target with minimum Center Error of Probability (CEP). Auto-pilot is the mechanism which drives the vehicle and keeps it in the predesigned trajectory using the control elements in the vehicle viz, the Fins, Nozzle and Jet Vanes. The autopilot algorithm coded in the Ob-Board Computer computes the corrections needed in the attitude of the vehicle, based on the present state during the trajectory. Identification of the dominant frequency using the proposed architecture in a dedicated hardware are detailed in Chapter 4. In this chapter, a scheme is proposed to use the identified instantaneous dominant frequency to dynamically tune the autopilot notch filter coefficients to have better control margins and stability of the flight vehicle during the path of the trajectory.

6.1 Auto-Pilot and Attitude

Autopilot is a closed loop control system, to stabilise the missile motion about its CG and manoeuvres/orients the missile in the desired direction and manner as required by the guidance while effectively rejecting the disturbances. The part of Missile Flight control system (FCS), configured for modifying missile motion by giving deflection commands to actuators using accelerometers and /or rate gyros feed backs, is usually called Autopilot.

Thus, the Process of determining the Control laws, which ensure stability, tracking and rejecting disturbances and noise, in simple terms, the Process of determining the gains / characteristics of the Compensator ensuring the above criteria is the primary

functions of the auto-pilot design. Considering the flight vehicle as a plant, then the attitude parameters are defined as in Fig.6.1

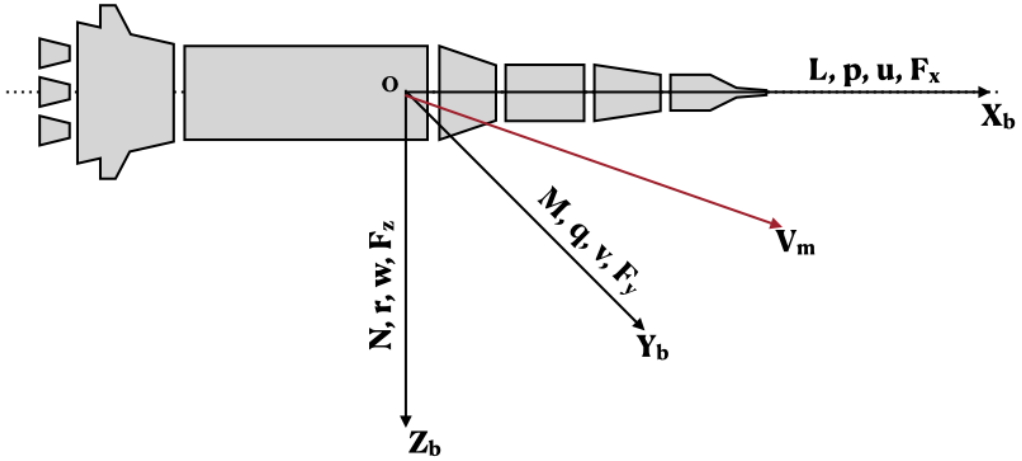

Figure 6.1 Flight Vehicle Attitude Parameters

where,

O	- Origin of Axis System, located at instantaneous CG of the vehicle
Xb	- Roll Axis, forward along the axis of symmetry
Yb	- Pitch Axis, outward and to the right if viewing the missile from the nozzle side
Zb	- Yaw Axis, Downward in the plane of symmetry
L,M,N	- Moments about Xb,Yb and Zb axes
Fx,Fy,Fz	- Forces along Xb,Yb and Zb axes
p,q,r	- Angular rates about Xb,Yb and Zb axes
u,v,w	- Velocity along Xb,Yb and Zb axes
Vm	- Total Velocity
Ixx, Iyy, Izz	- Moments of Inertia along axes system

The three translational equations of motion are defined by,

$$\frac{du}{dt} = \frac{Fx}{Mass} + (r * v \ - \ q * w) \tag{6.1}$$

$$\frac{dv}{dt} = \frac{Fy}{Mass} + (p * w \ - \ r * u) \tag{6.2}$$

$$\frac{dw}{dt} = \frac{Fz}{Mass} + (q * u \ - \ p * v) \tag{6.3}$$

and the rotational equations of motions are defined by,

$$\frac{dp}{dt} = \frac{L}{Ixx} + \frac{(Iyy - Izz) * q * r}{Ixx} \tag{6.4}$$

$$\frac{dq}{dt} = \frac{M}{Iyy} + \frac{(Izz - Ixx) * r * p}{Iyy} \tag{6.5}$$

$$\frac{dr}{dt} = \frac{N}{Izz} + \frac{(Ixx - Iyy) * p * q}{Izz} \tag{6.6}$$

6.2 Control and Margins

The vehicle performance is analysed using detailed simulation model where all non-linearities and disturbances including atmospheric disturbances are studied. Such a study is called a 'Long period' analysis and it includes the effect of autopilot performance parameters on the overall trajectory. Based on the analysis results, the autopilot specifications and parameters are modified to meet the mission objectives. The major objectives of the control system for a flight vehicle are [1],

1. Good stability margins
2. Good speed of response to guidance commands leading to good tracking accuracy
3. Less sensitivity to disturbances

Specifications on the stability margins need to take care of the following.

- Approximations in the analytical models used for vehicle and other control elements inside the control loop such as actuators, sensors etc.
- Uncertainties of the parameter values in the above models
- The autopilot performance parameters such as damping ratio, overshoot etc.

It is observed from the practical experience that following stability margins for nominal case have given good performance during flight trials [1].

1. Gain margin > 6db
2. Negative gain margin $<$ -6db
3. Phase margin $> 30^o$
4. Stability margin R or —1 + GH—min > 0.5 , where GH is the open loop gain of the system
5. Attenuation for higher modes >10db

The vehicle dynamics continuously changes during flight due to change in mass, moment of inertia, centre of gravity and the trajectory parameters such as dynamic pressure, Mach number, etc. Time-slice approach is used as a standard practice in which the vehicle parameters are assumed to remain constant for a short period of time. The design is carried out for regular interval of flight time and the schedule of gains is arrived. The gain adaptation based on the type of flight vehicle for which the trajectory is designed. For launch vehicles, the trajectory is very well- defined long before the actual flight. The gains are generally made a function of time. It is subsequently tested for adequacy for expected variations in thrust, centre of gravity, and aerodynamic perturbations, giving maximum possible shift in centre of pressure. The trajectory is not so well-defined in advance for missiles and the gains are the function of missile velocity or the dynamic pressure. The design needs to be validated for the entire flight zone and perturbations in inertial parameters, propulsion and aerodynamic data. For aircraft, one needs to cater for take-off, landing and the remaining flight conditions, depending on whether it is a transport aircraft or a fighter. It is observed that gains are stored as huge, two- dimensional arrays with altitude and Mach numbers as dimensions and is likely to involve a tedious and time-consuming design cycle during the development time. An adaptive design scheme has been used to overcome the above disadvantages. The scheme consists of the following features.

- Determination of vehicle transfer function
- An algorithm for computing gains
- Specification of dominant poles or control bandwidth.

It requires to determine the vehicle transfer function to make the control adaptive, which is achieved in the following manner.

- Mass, Centre of gravity, and Iyy are made as a function of time (due to propellant burning) from a look up table or a polynomial on time.
- Aerodynamic parameters are obtained as a function of the Mach number
- The dynamic pressure and Mach number are obtained by obtaining air density and sound velocity as the polynomials in altitude from standard atmosphere and instantaneous missile velocity.

The design also incorporates an additional compensator for achieving the desired gain and phase margins and filters to handle structural flexibility. The design is tested extensively by simulation for a large number of trajectories and a number of flight tests. The auto-pilot algorithm thus arrived is defined as the final one for the specific flight article.

This tedious process of extensive simulation runs and repeated ground tests after each flight test can be avoided by employing the dynamically tuned Notch Filter proposed in this work. The filter coefficients are to be tuned dynamically based on the vehicle attitude and structural behaviour. This proposed technique can ensure the requirements of stability margins for flight vehicles with high manoeuvrability by using real-time modal frequency updates to dynamically tune the notch filter coefficients.

6.3 Notch Filter for Modal Frequencies

Notch filters are employed to filter out the flex frequencies of the vehicle, so that only the rigid body variations are be considered for the control loop. The simplified block diagram of the sensor loop in autopilot algorithm is shown in Fig. 6.2

The sensor inputs measuring the vehicle body rates and the accelerations are filtered through a notch filter, so as to allow only the rigid body variations to be addressed by the autopilot algorithm. The unstable pole of the flight vehicle is defined by

$$\mu_\alpha = \sqrt{\frac{CN_\alpha * q * s * h}{I_{yy}}} \tag{6.7}$$

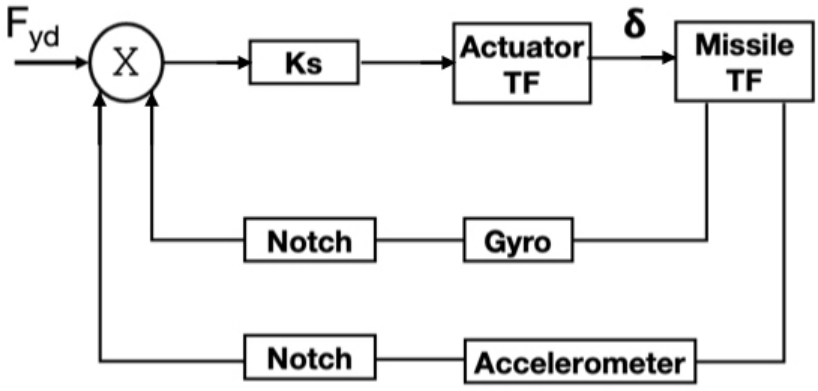

Figure 6.2 Block Diagram of Sensor Loop

where, h = (Centre of Pressure – Centre of Gravity), S = Surface area, q = Dynamic pressure and Iyy = Moment of Inertia. The stability of the vehicle is ensured either by Gain Stabilisation with adequate attenuation or by Phase Stabilisation with higher closed loop damping of the modal oscillations than the pure structural damping. The transfer function for the Notch filter is

$$F_{notch} = \left[\frac{s^2 + 2\zeta_1\omega_1 s + \omega_1^2}{s^2 + 2\zeta_2\omega_2 s + \omega_2^2}\right]\frac{\omega_2^2}{\omega_1^2} \tag{6.8}$$

where, ω_1 and ω_2 are the flex frequencies and σ is the damping factor. For symmetric notch filter, $\omega_1 = \omega_2$. In conventional approach, based on the ground experimental results, a single frequency notch filter or a wideband cascaded notch filters are designed to attain the necessary control margins.

6.4 Proposed Dynamically Tuneable Notch Filter

The conventional auto-pilot algorithm uses the notch filter of fixed frequency with wide band or a cascaded filters for a band of frequencies to filter out the unwanted flex disturbances to the extent possible and at the same time maintaining the system lag within the acceptable limits. The proposed scheme is to utilise the modal frequency identified in real-time during the course of trajectory to tune the notch filter co-efficients. The block diagram of the proposed scheme is in Fig. 6.3. The functionality of different building blocks of the proposed scheme is explained in the following subsections.

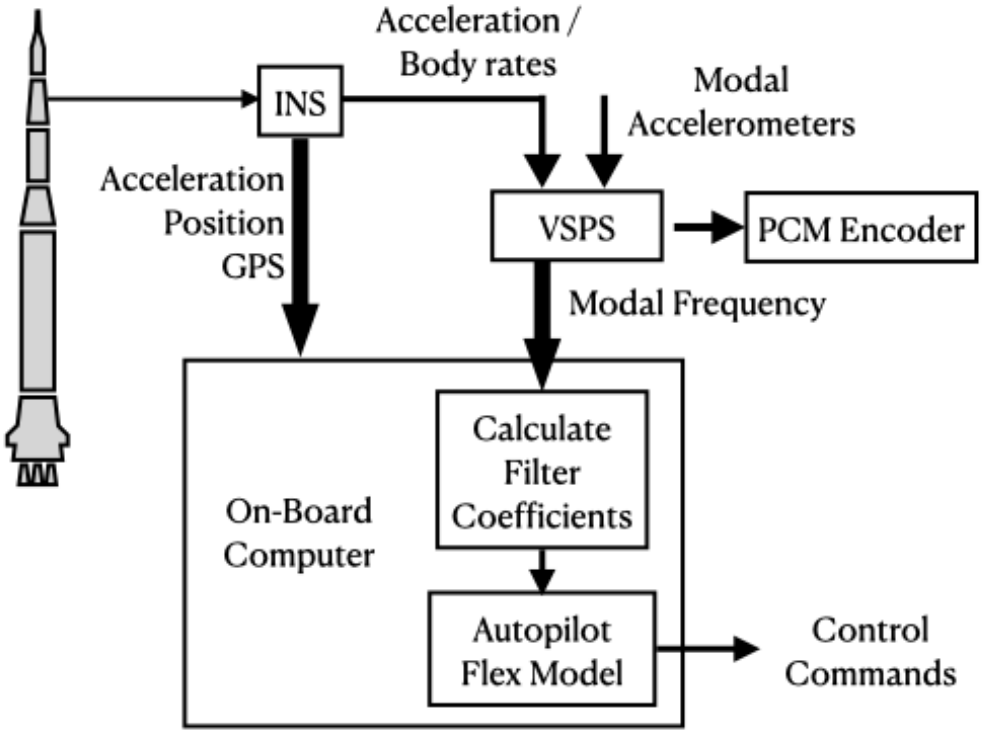

Figure 6.3 Block Diagram of Proposed Scheme

INS - Inertial Navigation System is an important avionics system that provides the position and acceleration of the flight vehicle. The sensors provide the acceleration of the vehicle in three directions and the body rates of the article in three axes.

Modal Accelerometers - senses the flex mode vibrations of the structure at the critical locations.

VSPS - Vibration Signal Processing System processes the sensor inputs from the Inertial sensors and the accelerometers. The signals are converted and analysed in the frequency domain to identify the dominant modal frequency of the flight structure and the spectral behaviour.

On Board Computer (OBC) - This is the central controller of the flight vehicle. It communicates with other avionics sub-system through the avionics bus to collect data and issue the necessary control commands. Also, it ensures the intended trajectory of the vehicle during the flight with the autopilot algorithm. To dynamically tune the autopilot notch filter to avoid control-structure interaction, it receives the dominant frequency

data from the VSPS and based on the identified dominant modal frequency, the filter co-efficients are calculated in real-time. These co-efficients thus calculated are used to finetune the notch filter at that specific frequency in the autopilot flexibility modal. Control commands are issued after effectively filtering out the disturbances from the modal behaviour of the structure thereby improving the stability margins.

6.5 Performance Comparison

Based on the Structural Analysis of the flight vehicle and Ground Resonance Test of the Flight Simulated article, it is estimated that the first mode frequencies of the test article vary from 19 Hz to 31 Hz during the course of its trajectory. With conventional method, notch filter is designed to cater this band with three filters in cascade with the central frequencies of 19 Hz, 24 Hz and 29 Hz. The inflight measurement of the body rates and accelerations are received in the ground station through telemetry link. The channel Az i.e., acceleration in Z- axis received by the Ground Telemetry Station in one of the flight trial is selected for analysis and the time-data plot is shown in Fig. 6.4.

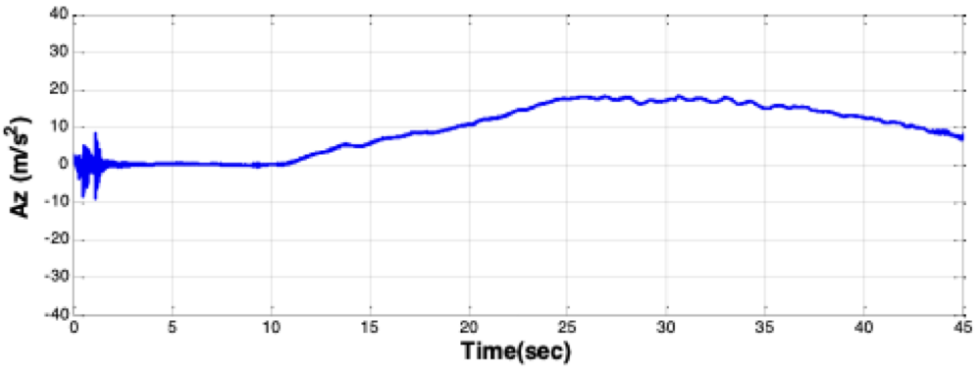

Figure 6.4 Measured Body Rates from Telemetry Data

Spectral analysis is carried out with the telemetry data of Az measured by the Inertial sensors using the ground Spectrum Analyser and the time-frequency plot is shown in Fig. 6.5. It is observed from Fig.6.4 and Fig.6.5 that high vibrations are encountered by the article during the initial phase and the frequency is varying continuously upto around 50 secs. For the subsequent duration of the flight, major variation in frequency is

not observed. Analysis is carried out on the same Az signal with the hardware developed

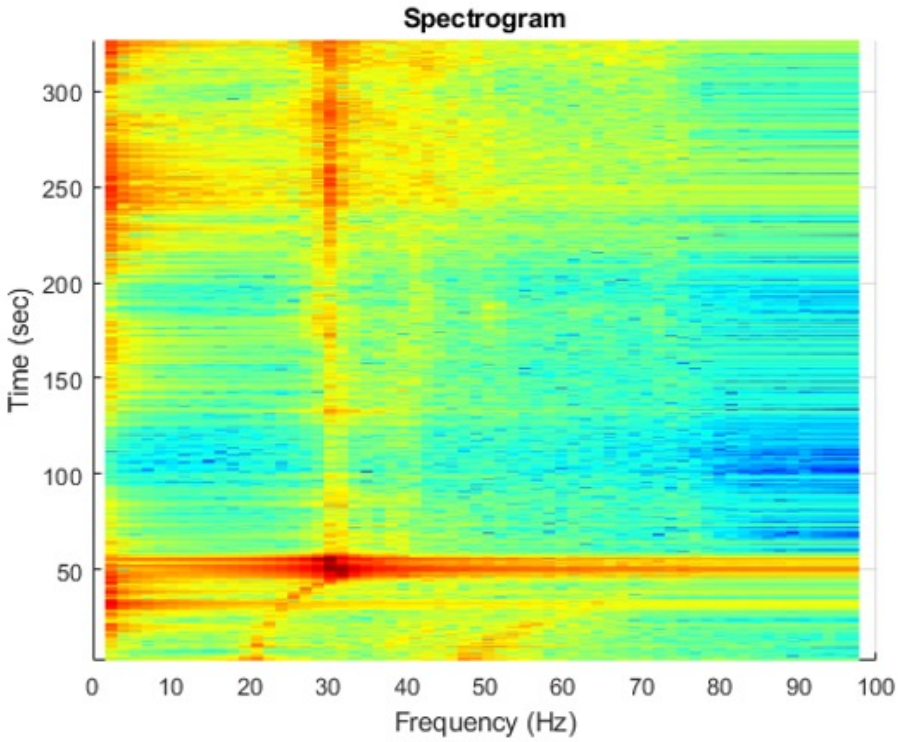

Figure 6.5 Spectrogram of Az

and described in earlier section in HILS test bed by feeding the ground received data. The output of the analysis is plotted as Waterfall diagram in Fig. 6.6 for the initial 45 secs. of the flight duration and few of the first modal frequency points with time and amplitude are shown in Table 6.1.

Table 6.1 Variation of First Mode frequency with time

Y-Time(sec)	3.3299	21.3628	34.2435	44.548
X-Freq(Hz)	19.4089	22.1262	24.0671	30.278
Z-Acc.(g^2/Hz)	0.23299	0.040785	0.033598	0.2135

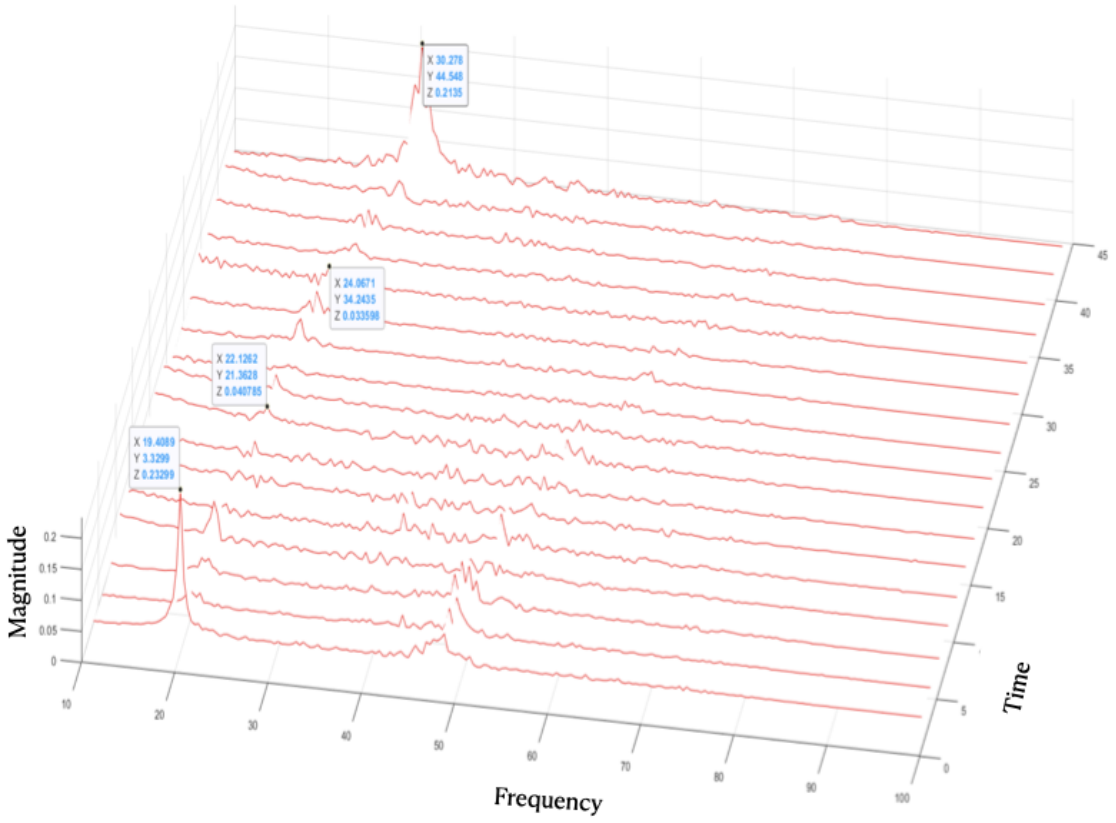

Figure 6.6 Waterfall Diagram of Az

It is observed from Fig.6.6 that the first mode bending frequency of the flight vehicle is varying between 19.4 Hz to 30.2 Hz, which is very close to the design prediction. As the mass and CG of the flight vehicle is varying during the initial time of the flight, the frequency is also varying. Beyond the burnout period, the frequency is stabilised at around 30 Hz, as there is no variation in the Mass properties, except those induced by the temperature, which is benign. The auto-pilot notch filter is to be designed with the predicted modal frequency value that the filter attenuates the signals in the frequency range of 19 to 31 Hz. In conventional method, the same is achieved using cascaded filters to cover the predicted frequency range with effective attenuation and acceptable phase lag. But the performance of the proposed auto pilot notch filter is highly improved with the dynamic tuning of the notch filter coefficients. In the proposed scheme, the body rates are measured by the sensors and the signal is processed to identify the instantaneous first mode frequency. The filter co-efficients are tuned to that particular instantaneous modal frequency and the co-efficients are modified based on the real-time values as the time

progresses. The improvement in the filter performance is as shown in Fig. 6.7 and Fig. 6.8

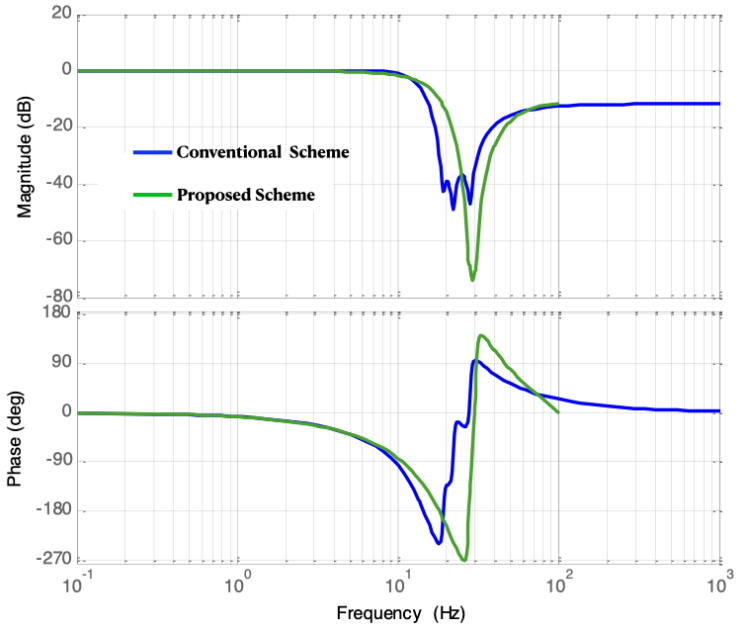

Figure 6.7 Filter Performance with same Phase Lag

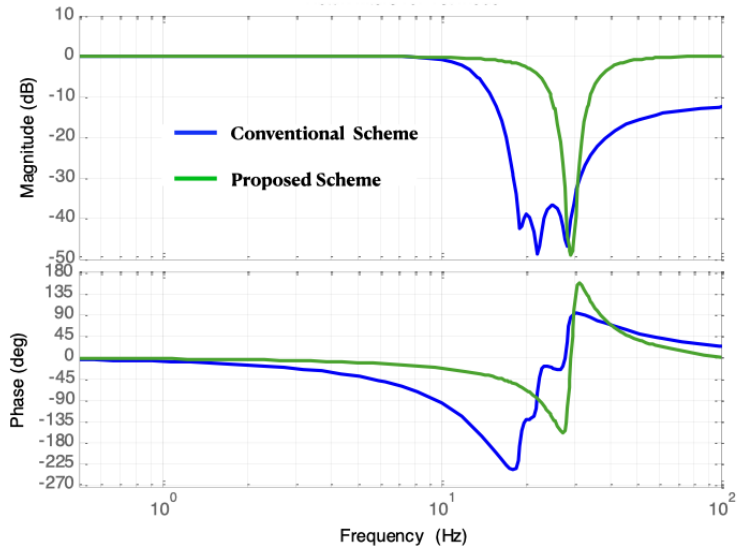

Figure 6.8 Filter Performance with same Attenuation

As can be clearly seen from the plots in Fig 6.6 and 6.7, with ground simulation and prediction, broad notch filters to cater the wide range of frequency is needed, but

with real-time measurement during the flight, narrow band notch filters can be effectively implemented. Comparing with the conventional method, the proposed dynamic tuning scheme can achieve better attenuation of the order of 30dB more, while achieving the same phase lag. It may also be noted that the proposed scheme can reduce phase lag by 5^0 to 10^0, while maintaining the same level of attenuation as that of conventional method. Based on the control system requirements, a trade-off can be made in the filter characteristics, while dynamically tuning with the real-time frequency values.

The dynamic tuning of the autopilot notch filter with real-time data update of the structural flexibility improves the overall margins in the flight vehicle control system. The Frequency Response Analysis of the guidance algorithm is plotted in Fig. 6.9. It is observed that there is a considerable improvement in the control margins. Gain Margin is increased by around 5dB and the Phase Margin is increased by almost 7^o by dynamic tuning as compared with the conventional method.

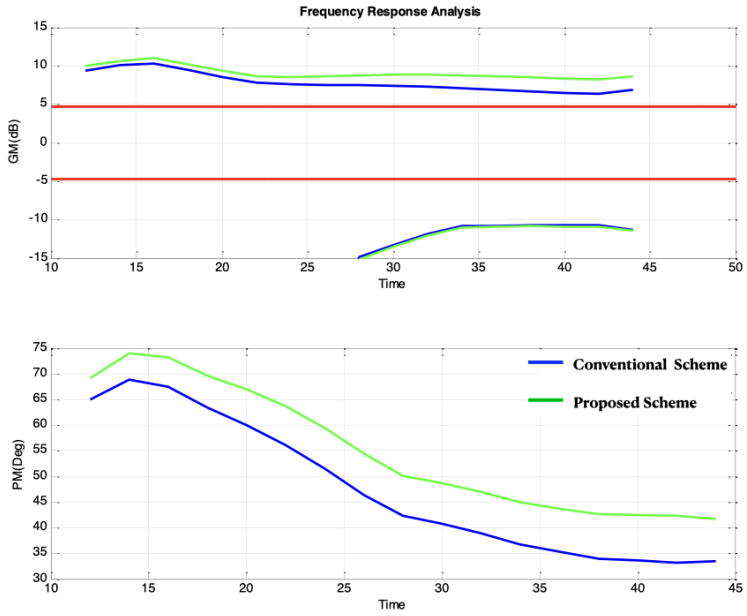

Figure 6.9 Frequency Response Analysis Results

6.6 Conclusion

A scheme for dynamic tuning of the Notch filter in the flexibility loop of the autopilot control system is proposed in this chapter. In conventional method, the notch filter is

fixed for a wide band frequency based on the flight vehicle configuration and the results of the ground resonance test. Assumptions and boundary conditions imposed during the simulation studies and ground tests induce errors in the prediction of modal frequency. These errors in prediction reduces the control margins of the autopilot control system. The real-time estimation of the modal parameters, especially the modal frequency, enables the precise control of the flight vehicle during its trajectory, by tuning the control coefficients on-board the flight vehicle. The acceleration signals sensed by the Navigation sensors are processed in frequency domain and the values are used in real-time tuning of the notch filter in autopilot algorithm. The proposed dynamic tuning of the notch filter for autopilot based on the real-time identification of the modal frequency provides effective attenuation and the acceptable phase lag, thereby improving both Phase and Gain margins. The associated hardware for the processing of the accelerometer and the gyro signals plays a crucial part in the successful implementation of this novel method of auto-pilot algorithm. Thus, tuning of the notch filter with instantaneous estimates provides a better solution for the system designer of the flight vehicle with increased Gain margin of 5dB and phase margin of 7^0 with reference to the conventional methods presently used.

Chapter 7

Conclusions and Future Scope

This chapter concludes by underlining the main contributions. It also presents the possible directions of future work.

7.1 Conclusions

In this research work, the effect of structural flexibility on the control margins and possible methods to improve the margins by dynamically tuning the autopilot notch filter are analysed. The conventional method of auto-pilot algorithm has a fixed wide band filter to cater the complete trajectory variations. This reduces the available control margins for the designer of autopilot as uncertainties and assumption errors are more. The mitigation methodology to overcome this limitation is addressed. The development and validation of mathematical models, architecture for processing the signals in real-time during the flight, validation of the architecture at field conditions and the dynamic tuning of the autopilot notch filter are addressed and the results are validated in Hardware-In-Loop Simulation test setup.

The mathematical models for time domain analysis, frequency domain analysis and time-frequency analysis are developed and tested with known sinusoids. These models are again validated in Hardware-In-Loop Simulation setup with both known sinusoids and random vibration signals. Based on the simulation and test results, frequency domain analysis of the random vibration signals using Welch method is finalized. Three different windowing techniques viz. Hann, Hamming and Blackman-Harris Windows are chosen

for filtering the input signals. Input data overlapping of 0%, 25% and 50% are modelled and tested with real-time inputs for verifying the accuracy of the result with respect to the reference controller data. This finalized method is cross verified with the flight telemetered data and the predictions.

A suitable architecture is designed to implement the mathematical model. The architecture is designed and modelled in HDL. A memory-based architecture is proposed for the hardware realization of the proposed mathematical model. This architecture is designed to realize the Windowing Function, Frequency Domain Conversion, Computation of instantaneous Power Spectral Density spectrum or averaged-out spectrum based on overlapping criteria and identification of Frequency Maxima in hardware. All the modules of the proposed architecture are modelled using Verilog HDL and functional simulations are performed using Xilinx Design Vivado tools. The Verilog netlist is synthesized targeting Kintex-7 series FPGA device and ported on the proto-board of Kintex-7 to verify the performance of the proposed architecture. The functionality of implemented proposed architecture is verified by applying Real-time doppler input signal for identification of the dominant frequency. The sensitivity of architecture is also tested by varying the floor level of Input signal noise.

A dedicated hardware is designed and realised with the proposed architecture with necessary pre and post processing electronics. The dedicated Vibration Signal Processing System suits the existing avionics configuration and connectivity with MIL STD 1553B serial bus. The system is cleared in the bench level and tested with real-time input from Flight section-level vibration test setup. The output of the architecture ported on the dedicated hardware is transmitted to the ground telemetry data acquisition system through telemetry Bi-phasal port of the unit. It is observed from the field test results that the spectrum signatures from the VSPU are closely matching with the reference controller, except a small variation in the amplitude to the tune of 0.00102 g^2/Hz even when two dominant frequencies are at close proximity. The maximum difference between the reference and estimated dominant frequency is 4.9 Hz in one measurement amounting to 1.9% of reference value and 0.24% in full scale range of 2000 Hz. For all other test cases, the accuracy of estimation over the full-scale range is less than 0.2% and over the reference value is less than 2%. These values are well within the acceptable limit of the

mission design requirement.

The conventional auto-pilot algorithm uses the notch filter of single frequency with wide band or a cascaded filters for a band of frequencies. The notch filter is so designed that the lag of the system is maintained within the acceptable limits and at the same time the unwanted flex disturbances are filtered out to the extent possible. A scheme is proposed to utilise the modal frequency identified in real-time during the course of trajectory to tune the notch filter co-efficients. The dominant modal frequency identified by the flight hardware in real-time can be utilised in the autopilot algorithm to derive the filter coefficients. Thus the dynamically tuned notch filter will effectively filter out the disturbances initiated by the control-structure interaction and provide better control margins as compared to the conventional method. Comparing with the conventional method, for achieving the same phase lag, better attenuation of the order of 30dB more can be achieved with the proposed dynamic tuning scheme. Or if the attenuation level is maintained as that of the conventional method, then the phase lag can be reduced by 5^0 to 10^0 with the proposed scheme. From the frequency response analysis of the control loop, the Gain Margin is increased by around 5dB and the Phase Margin is increased by almost 7^0 , when the narrowband notch filters are used based on the proposed scheme as compared to the wide band filter with conventional simulation methods.

7.2 Future Scope

The work proposed in this thesis can be extended for future research. Some of the possible directions in which the problems can be further pursued are:

- The proposed architecture is designed to determine the dominant modal frequency of the flight vehicle. The same can be extended for identification of Multiple dominant frequency points for complex and unsymmetric airborne structures based on multi-modal analysis suitable for real-time applications.
- The proposed parameter estimation methods and analysis of vibration signals in real-time may be extended for Active Vibration control which is still in nascent stage.

- The proposed hardware architecture may be further optimized and the optimized architecture may be glued to the On-Board Computer Architecture to reduce the electrical cable harness and better rate of data transfer

Publications

List of International Journals:

1. R Srinivasan, TessyThomas, B Lakshmi. Power Spectral Density Computation and Dominant Frequencies Identification from the Vibration Sensor Output under Random Vibration Environment. Defense Science Journal .2020; 70(06), 692-700. https://doi.org/10.14429/dsj.70.15535

2. R Srinivasan, B Lakshmi, Tessy Thomas. "An Architecture for On-board Frequency Domain Analysis of Launch Vehicle Vibration Signals", Defense Science Journal. Jan 2023;73(01), pp 3-10, DOI : 10.14429/dsj.73.17690

3. R Srinivasan, B Lakshmi, Tessy Thomas. FFT Architectures: A Review. Reveiwers' comments under updation

4. R Srinivasan, B Lakshmi, Tessy Thomas. Dynamic Tuning of Autopilot Notch filter in Flight Vehicles, Under Review

Bibliography

[1] N.V.Kadam, *Practical Design of Flight Control Systems for Launch Vehicles and Missiles*. Allied Publisher Ltd., 2019, no. ISBN9789387997813.

[2] Tessy.Thomas and Dr.Jagirdhar, "Online trajectory reshaping for a launch vehicle to minimize the final error caused by navigation and guidance," *Defence Science Journal*, vol. 17, no. 3, pp. 254–261, 2013.

[3] Karthik.A, V. B. Reddy, and M. Kumar, "Numerical investigation of modal parameters for an airborne system," in *National Conference on Environmental Testing of Aerospace Systems – Advances & Future Trends (CETAS), Thiruvananthapuram, India*, October 11-12, 2019.

[4] Vernon.L.Alley and Sumner.A.Leadbetter, "Prediction and measurement of natural vibration of multistage launch vehicles," *JAIAA Journal*, vol. 1, no. 2, 1963.

[5] N.V.Kadam, "Practical design of flight control systems: Some problems and their solutions," *Defence Science Journal*, vol. 55, no. 3, pp. 211–221, July 2005.

[6] D.J.Ewings, *Modal Testing Theory, Practice and Application*, 2nd ed. Research Studies Press Ltd,Baldock, Hertfordshire, England, 2000.

[7] J. He and Z. Fu, *Modal Analysis*. Butterworth Heinemann, 2001, no. ISBN3540405305.

[8] Nuno.M.M.Maia and Julio.M.M.Silva., *Theoretical and Experimental Modal Analysis*. Research Studies Press Ltd., Baldock, Hertfordshire, England, 1997, no. ISBN0-86380-208-7.

[9] L. Ying, Papandreou.Suppappola, and Darryl.Morrell, "Instantaneous frequency estimation using sequential bayesian techniques," in *Fortieth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA*, 29 Oct – 01 Nov 2006.

[10] Kootsookos.P.J, "A review of the frequency estimation and tracking problems. crc for robust and adaptive systems," in *DSTO, Salisbury Site, Frequency Estimation and Tracking Project*, February 21, 1999.

[11] J. J. Wijker, *Mechanical Vibrations in Spacecraft Design.* Springer Verlag Berlin Heidelberg, 2004, no. ISBN3540405305.

[12] Griffin.S, Lane.S, and Leo.D, "Innovative vibroacoustic control approaches in space launch vehicles," in *29th International Congress and Exhibition on Noise Control Engineering, Nice, France*, 27-30 August 2000.

[13] A. J. W and Tischler.Mark.B, "Determining xv15 aeroelastic modes from flight data with frequency domain methods," *NASA Technical Paper 3330*, vol. 55, 1993.

[14] M. Feldman, "Non-linear free vibration identification via the hilbert transform," *Journal of Sound and Vibration*, vol. 208, no. 3, pp. 475–489, 1997.

[15] Uhl.Tadeusz, Petko.Maciej, Grzegorz, and Klepka.Andrzej, "Real time estimation of modal parameters and their quality assessment," *Shock and Vibration, Hindawi, IOS Press*, vol. 11, no. 15, pp. 299–306, 2008.

[16] Ruzzene.M, Fasana.A, Garibaldi.L, and Piombo.B., "Natural frequencies and dampings identification using wavelet transform: Application to real data," *Mechanical Systems and Signal Processing*, vol. 11, no. 2, pp. 207–218, 1997.

[17] Traperoa.Juan.R, Hebertt.Sira-Ramı´rezb, and Vicente.Feliu.Batllea, "A fast on-line frequency estimator of lightly damped vibrations in flexible structures," *Journal of Sound and Vibration*, vol. 307, no. 15, pp. 365–378, 2007.

[18] Jesús.Ponce.de.León, Beltrán.Fernando, and Beltrán.José.R., "A complex wavelet based fundamental frequency estimator in single channel polyphonic signals," in *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13). Maynooth, Ireland,*, September 2-5, 2013.

[19] Nielsen.Jesper.Kjær, Jensen.Tobias.Lindstrøm, Jensen.Jesper.Rindom, Christensen.Mads.Græsbøll, and Jensen.Søren.Holdt, "Fast and statistically efficient fundamental frequency estimation," in *Proceedings of I E E E International Conference on Acoustics, Speech and Signal Processing*, 2016.

[20] Santamaria.Iganacio, Pantaleon.Carlos, and Ibanez.Jesus, "Comparative study of high-accuracy frequency estimation methods," *Mechanical Systems and Signal Processing*, vol. 14, no. 0, 2000.

[21] Neild.S.A, McFadden.P.D, and Williams.M.S, "A review of time-frequency methods for structural vibration analysis," *Engineering Structures*, vol. 25, no. 0, p. 713–728, 2003.

[22] Bhairannawar, Satish.S, Sarkar.Sayantam, Raja.K.B, and Venugopal.K.R, "An efficient VLSI architecture for fingerprint recognition using O2D-DWT architecture and modified CORDIC-FFT," in *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2015, pp. 1–5.

[23] Gonzalez.Rafael.C, Woods.Richard.E., and Steven.L.Eddins, *Digital Image Processing*, 2nd ed. Pearson Education, 2004.

[24] J. W. Cooley, James W Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comp.*, vol. 19, pp. 297–301, 1965.

[25] Good.I.J, "The interaction algorithm and practical fourier analysis," *Journal of the Royal Statistical Society*, vol. Series B. 20 Addendum 22, no. 2, pp. 361–372, Addendum 373–375, July 1958, Addendum 1960.

[26] L. H. Thomas, "Using a computer to solve problems in physics," *Applications of Digital Computers,Boston: Ginn.*, no. 2, 1963.

[27] Rader.C.M., "Discrete fourier transform when the number of data samples is prime," vol. 56, no. 6, pp. 1107 – 1108, June 1968.

[28] Richard.Tolimieri, Chao.Lu, and Johnson.Robert.W., "Modified winograd FFT algorithm and its variants for transform size n = p^k and their implementations," *Advances in Applied Mathematics*, vol. 10, no. 2, pp. 228–251, June 1989.

[29] Brenner.N and Rader.C, "A new principle for fast fourier transformation," *IEEE Acoustics, Speech Signal Processing*, vol. 23, no. 3, Jun 1976.

[30] Bruun.G, "z-transform DFT filters and FFT's," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 56 – 63, February 1978.

[31] Rabiner.L and Gold.B, *Theory and Application of Digital Signal Processing*. Prentice Hall Inc., Englewood Cliffs, N. J., 1975, no. ISBN9789387997813.

[32] B. Baas, "A low-power, high-performance, 1024-point FFT processor," *EEE Journal of Solid-State Circuit*, vol. 34, no. 3, p. 380–387, 1999.

[33] H. Groginsky and G. Works, "A pipeline fast fourier transform," *IEEE Transactions on Computers*, vol. C-19, no. 11, pp. 1015 – 1019, Nov. 1970.

[34] V. Boriakoff, "FFT computation with systolic arrays, a new architecture," *IEEE Transactions on Circuits and Systems -11, Analog and Digital Signal Processing*, vol. 41, no. 11, pp. 1015 – 1019, Apr 1994.

[35] VJoseph.Ja'.Ja' and Robert.Michael.Owens, "An architecture for a VLSI FFT processor," *ntegration Journal*, vol. 1, no. 4, pp. 305–316, Dec 1983.

[36] Shousheng.He and Mats.Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference, Santa Clara, CA, USA*, 14 May 1998.

[37] T.Widhe, J.Melander, and L.Wanhammary, "Design of efficient radix-8 butteirfly PEs for VLSI, year=June 9-12, 1999, pages = 2084-2087,," in *IEEE International Symposium on Circuits and Systems, Hong Kong.*

[38] Park.Se.Ho, Dong.Hwan.Kim, Dong.Seog.Han, Kyu.Seon.Lee, S. Park, and Jun.Rim.Choi, "Sequential design of a 8192 complex point FFT in OFDM receiver," in *First IEEE Asia Pacific Conference on ASICs (Cat. No.99EX360),Seoul,*, 23-25 Aug. 1999.

[39] Yun.Nan.Chang and K. Parhi, "An efficient pipelined FFT architecture," *IEEE Transanctions on circuits and systems - II; Analog and digital signal processing*, vol. 50, no. 6, pp. 322–325, June 2003.

[40] G.Bi and E.V.Jones, "A pipelined FFT processor for word-sequential data," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-37, p. 1982–1985, Dec. 1989.

[41] C.Chiu, Wing.Hui, T.J.Ding, and J.V.McCanny, "A 64-point fourier transform chip for video motion compensation using phase correlation," *IEEE Journal of Solid-State Circuits*, vol. 31, p. 1751–1761, Nov. 1996.

[42] Mario.Garrido, J.Grajal, M.A.Sánchez, and Oscar.Gustafsson, "Pipelined radix- 2^k feedforward FFT architectures," *Transactions on very large scale integration (VLSI) systems*, vol. 22, no. 1, Jan 2013.

[43] Shousheng.He and Mats.Torkelson, "A new approach to pipeline FFT processor," in *Proceedings of International Conference on Parallel Processing, Honolulu, HI, USA*, 15-19,April 1996.

[44] Yun.Nan.Chang, "An efficient vlsi architecture for normal i/o order pipeline FFT design," *IEEE Transactions on circuits and systems - II: Express Briefs*, vol. 55, no. 12, p. 1751–1761, Dec 2008.

[45] Marshall.C.Pease, "An adaptation of the fast fourier transform for parallel processing," *Journal of Association for Computing Machinery*, vol. 15, no. 2, pp. 252–264, April 1968.

[46] John.E.Whelchel, John.P.O'Malley, William.J.Rinard, and James.F.McArthur, "The systolic phase rotation FFT - a new algorithm and parallel processor architecture," in *International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, USA*, 3-6 April 1990.

[47] Manohar.Ayinala, Michael.Brown, and Keshab.K.Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 20, no. 8, June 2012.

[48] Sayed.Ahmad.Salehi, Rasoul.Amirfattahi, and Keshab.K.Parh, "Pipelined architectures for real-valued FFT and hermitian-symmetric IFFT with real datapaths," *IEEE Transactions on circuits and systems - II, Express Briefs*, vol. 60, no. 8, August 2013.

[49] Kenneth.S.Stevens and Bruce.W.Suter, "A mathematical approach to a low power FFT architecture," in *IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA,USA*, 31 May-3 June 1998.

[50] Han.Wei, Erdogan.AT, Arslan.T, and H. M, "Multiplier-less based parallel pipelined FFT architectures for wireless communication applications," in *International conference on acoustics, speech and signal processing (ICASSP 2005)*, 2005, pp. 45–48.

[51] H.S.Lee.et.al, "Systolic array architecture for VLSI FFT processor," *International Journal of Mini and Microcomputers*, vol. 6, no. 3, pp. 49–54, 1984.

[52] G.E.Bridges.et.al, "Dual systolic architectures for VLSI digital signal processing systems," *IEEE Trans. on Computers*, vol. 35, no. 10, pp. 916–923, 1986.

[53] H.T.Kung and J.A.Webb, "Mapping image processing operations onto a linear systolic machine," *Distrib. Comput*, vol. 1, pp. 246–257, 1986.

[54] Jen.Chih.Kuo, Ching.Hua.Wen, and C. ande An.Yeu.Wu, "VLSI design of a variable-length FFT/IFFT processor for OFDM-based communication systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 439360.

[55] Jianming.Wu, Ke.Liu.Bo.Shen, and Hao.Min, "A hardware efficient VLSI architecture for FFT processor in OFDM systems," in *6th International Conference on ASIC, Shanghai, China*, 24-27 Oct. 2005.

[56] S.S.Abdullah, H.Nam, M.McDermot, and J.A.Abraham, "A high throughput FFT processor with no multipliers," in *ICCD 2009. IEEE International Conference on. Computer Design*, 2009, p. 485–490.

[57] M.Garrido, R.Andersson, F.Qureshi, and O.Gustafsson, "Multiplierless unity-gain SDF FFTs," *EEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 9, p. 3003–3007, 2016.

[58] Jinti.Hazarika, Mohd.Tasleem.Khan, Shaik.Rafi.Ahamed, and Harshal.B.Nemade, "High performance multiplierless serial pipelined VLSI architecture for real-valued FFT," in *2019 National Conference on Communications (NCC), Bangalore, India*, 20-23 Feb. 2019.

[59] Hongjiang.He and Hui.Guo, "The realization of FFT algorithm based on FPGA co-processor," in *2008 Second International Symposium on Intelligent Information Technology Application,Shanghai, China*, 20-22 Dec. 2008.

[60] Harikrishna.K, K.Harikrishna, T. Rao, and Valadimir.A..Labay, "An efficient FFT architecture for OFDM communication systems," in *2009 Asia Pacific Microwave Conference,Singapore*, 7-10 Dec. 2009.

[61] Zahra.Haddad.Derafshi, Javad.Frounchi, and Hamed.Taghipour., "A high speed FPGA implementation of a 1024-point complex FFT processor," in *2010 Second International Conference on Computer and Network Technology, Bangkok, Thailand*, 23-25 April 2010.

[62] Stefan.Langemeyer, Peter.Pirsch, and Holger.Blume, "A FPGA architecture for real-time processing of variable-length FFTs," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),Prague, Czech Republic*, 22-27 May 2011.

[63] Garrido.M and Grajal.J, "Efficient memoryless CORDIC for FFT computation," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, 2007.

[64] Sharath.Chandra.Inguva and J.B.Seventiline, "Implementation of FPGA design of FFT architecture based on CORDIC algorithm," *International Journal of Electronics*, vol. 108, no. 11, 2021.

[65] Sunil.P.Joshi and Roy.Paily, "Distributed arithmetic based split-radix FFT," *Journal of Signal Processing Systems*, vol. 75, p. 85–92, 2014.

[66] M.Rawski, M.Wojtynnski, T.Wojciechowski, and P.Majkowski, "Distributed arithmetic based implementation of fourier transform targeted at FPGA architectures." in *14th International Conference on Mixed Design of Integrated Circuits and Systems,Ciechocinek, Poland.*, 21-23 June 2007.

[67] www. ni.com/instrument fundamentals, "Understanding FFTs and windowing," 22 May 2017.

[68] Trethewey.M.W., "Window and overlap processing effects on power estimates from spectra." *Mech. Syst. Signal Process*, vol. 14, no. 2, p. 267–278, 2000.

[69] M. Zivanovic and A. Carlosena, "On asymmetric analysis windows for detection of closely spaced signal components," *Mech. Syst. Signal Process*, vol. 20,, no. 3, p. 702–717, 2006.

[70] J. Muthuswamy and N. Thakor, "Spectral analysis methods for neurological signals," *Journal of Neuroscience Methods*, vol. 83, no. 1, pp. 1–14, 1998.

[71] Xu.X.M and Liu.Q.H., "Fast spectral-domain method for acoustic scattering problems," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 48, no. 2, pp. 522–529, 2001.

[72] Ombao.H and Ringo.Ho, "Time-dependent frequency domain principal components analysis of multichannel non-stationary signals," *Comput. Stat. Data Anal*, vol. 50, p. 2339–2360, 2006.

[73] Higgins.R.J, *Digital signal processing in* VLSI. Prentice Hall, Englewood, Cliffs, 1990, no. I.

[74] R.Srinivasan, Tessy.Thomas, and B.Lakshmi, "Power spectral density computation and dominant frequencies identification from the vibration sensor output under random vibration environment," *Defence Science Journal*, vol. 70, no. 6, pp. 692–700, November 2020.

[75] S.Kirkpatrick, CD.Gelatt.Jr, and M.P.Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 13 May 1983.

[76] Y.W.Lin, H.Y.Liu, and C.Y.Lee, "A 1-gs/s FFT/IFFT processor for UWB applications," *IEEE Journal Solid-State Circuits*, vol. 40, no. 8, p. 1726–1735, Aug 2005.

[77] Y.W.Lin and C.Y.Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, p. 807–815, Apr. 2007.

[78] A.Chinnapalanichamy and K.K.Parhi, "Serial and interleaved architectures for computing real FFT," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, p. 1066–1070.

[79] M.Garrido, N.K.Unnikrishnan, and K.K.Parhi, "A serial commutator fast fourier transform architecture for real-valued signals," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2017.