Artificial Neural Network based Fiber Bragg Grating Sensor Interrogation

A thesis submitted in partial fulfillment of the requirement for the degree of

Master of Technology
in
Computer Technology

Submitted by

Dipten Kumar

Registration No.: 160091 of 2021-22
Examination Roll No.: M6TCT24016

Session: 2021-2024

Under the Supervision of

Prof. Debotosh Bhattacharjee

Department of Computer Science and Engineering
Jadavpur University,
188, Raja S.C. Mallick Rd,
Kolkata- 700032,
West Bengal, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that this is a bonafide record of the project entitled **"Artificial Neural Network based Fiber Bragg Grating Sensor Interrogation"**, submitted by Dipten Kumar (University Registration No.: 160091 of 2021-22, Examination Roll No.: M6TCT24016) is hereby approved of a creditable study of a technological subject carried out under my supervision and presented in a manner satisfactory to warrant its acceptance for partial fulfillment of the requirements of the degree of Master of Technology in Computer Technology. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Prof. Debotosh Bhattacharjee
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Countersigned

Prof. Nandini Mukherjee
Head, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Prof. Dipak Laha
Dean, Faculty of Engineering & Technology
Jadavpur University, Kolkata-32, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval[1]

This is to certify that the thesis entitled **"Artificial Neural Network based Fiber Bragg Grating Sensor Interrogation"**, is a bonafide record of work carried out by Dipten Kumar in partial fulfillment of the requirements of the degree of Master of Technology in Computer Technology in Department of Computer Science & Engineering, Jadavpur University during the period of October 2021 to May 2024. It is understood that by this approval the undersigned do not necessarily endorse any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

……..
Signature of Examiner 1
Date

……..
Signature of Examiner 2
Date

Only in case the thesis is approved[1]

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled **"Artificial Neural Network based Fiber Bragg Grating Sensor Interrogation"** contains a literature survey and original research work by the undersigned candidate as a part of his Degree in Master of Technology in Computer Technology.

All information has been obtained and presented in accordance with the academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Dipten Kumar

Registration No.: 160091 of 2021-2022

Examination Roll No.: M6TCT24016

Thesis Title: **Artificial Neural Network based Fiber Bragg Grating Sensor Interrogation**

……
Signature with date

ACKNOWLEDGEMENT

The successful completion of this thesis was made possible through the guidance and assistance of numerous individuals, and I am deeply grateful for their support throughout. I take this opportunity to express my heartfelt appreciation to all those who generously contributed their time, encouragement, and assistance to this endeavor.

First and foremost, I am immensely grateful to my supervisor, **Prof. Debotosh Bhattacharjee**, whose keen interest in my thesis work and unwavering guidance propelled me to the completion of this project. His continuous optimism, enthusiasm, encouragement, support, understanding, and constructive criticism, coupled with his illuminating perspectives on various issues, were instrumental in shaping this thesis.

I would also like to extend my gratitude to the Head of the Department of Computer Science and Engineering at Jadavpur University, **Prof. Nandini Mukherjee**, and all other faculty members for their constant encouragement and support. Special thanks are due to my MTCT course instructors, **Prof. Jamuna Kanta Singh, Prof. Nibaran Das, Prof. Diganta Saha, Prof. Kamal Sarkar, Dr. Chintan Kumar Mandal, Dr. Mridul Sankar Barik, Dr. Anasua Sarkar**, and our course coordinator, **Prof. Sushmita Ghosh(De)**, for their encouragement and valuable insights throughout this journey. I am also grateful to the non-teaching staff of the Department of Computer Science and Engineering for their continuous support.

I owe a debt of gratitude to the scientists of the Fiber Optics and Photonics Division of CSIR-Central Glass & Ceramic Research Institute, particularly **Dr. Somnath Bandyopadhyay**, Chief Scientist, and **Ms. Nandini Basumallick**, Sr. Principal Scientist, for their guidance, support, and encouragement. I also extend my thanks to all my colleagues in the Fiber Optics and Photonics Division for their support and encouragement. Furthermore, I am thankful to the Director of CSIR-Central Glass & Ceramic Research Institute Kolkata and the Head of the Fiber Optics and Photonics Division for granting me permission to undertake this course and for their encouragement.

I am profoundly grateful to my family members for their unwavering encouragement, moral support, and sacrifices. My heartfelt thanks go to my father, **Mr. Sankar Chandra Santra**, my mother, **Mrs. Alpana Santra**, my wife, **Smita**, and my son, **Dhruv**, for their understanding, support, and unconditional love.

Lastly, I extend my appreciation to all my well-wishers, relatives, friends, colleagues, and classmates who have been a constant source of help and inspiration throughout my thesis work. My sincere thanks to all for being part of this remarkable journey.

……
Dipten Kumar
Examination Roll No: M6TCT24016
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Abstract

This thesis proposes novel artificial neural network (ANN)-based Bragg wavelength detection algorithms for a given fiber Bragg grating (FBG) sensor. In the introductory part of this thesis, the FBG sensor, its construction, working principle, applications, and interrogation schemes are discussed. Two popular interrogation techniques are described: one utilizing a CCD array spectrometer and the other employing a tunable filter. Then, two separate albeit related ANN-based algorithms are proposed for each of these two interrogation methods. The proposed algorithms address the actual non-ideal behaviors of the components, such as optical intensity variation over time, as well as the poor repeatability and poor linearity of the tunable filter. These algorithms also take into account the fact that the nominal Bragg wavelength of a given FBG sensor often changes during field deployment due to variations in pre-strain or ambient conditions, ensuring that the trained model remains valid and usable even with these variations. Furthermore, the proposed ANN algorithms are compared with traditional FBG Bragg wavelength determination methods, such as curve fitting and centroid finding methods. Finally, the scope of further work in this direction is also discussed toward the end of this thesis.

TABLE OF CONTENTS

Sl. No.	Topics	Page No.
Chapter-1	Introduction	01
1.1	What is a Fiber Bragg Grating sensor?	01
1.2	Where do we use FBG sensors and Why?	03
1.3	Fabrication of FBG sensors	05
1.4	Working principle of FBG sensor (derivation of Bragg equation)	05
1.5	Interrogation of an FBG sensor	06
Chapter-2	Literature Survey	07
2.1	CCD Spectrometer based Interrogation Architecture and Working principle	07
2.2	Tunable Filter based Interrogation Architecture and Working principle	10
2.3	Problem Statements	19
Chapter-3	CCD array spectrometer-based FBG Interrogation using ANN algorithm	20
3.1	Set-up & Methodology of FBG Interrogation	20
3.2	ANN Training Algorithm for CCD array-based Interrogator	21
3.3	Bragg Wavelength Prediction using the proposed trained ANN for CCD array-based FBG Interrogator	30
3.4	Implementation and Performance of the model	34
3.5	Comparison with other FBG central wavelength detection algorithms	35
Chapter-4	Tunable Filter-based FBG Interrogation using ANN algorithm	39
4.1	Set-up & Methodology of FBG Interrogation	39
4.2	ANN Training Algorithm for Tunable Filter-based Interrogator	40
4.3	Bragg Wavelength Prediction using the proposed trained ANN for Tunable Filter-based FBG Interrogator	50
4.4	Implementation and Performance of the model	53
4.5	Comparison with other FBG central wavelength detection algorithms	54
Chapter-5:	Conclusion and Future Scope of Work	57
References		59

LIST OF FIGURES

Fig No.	Figure Description	Page No.
Fig 1.1	structure of FBG	01
Fig 1.2	property of FBG	02
Fig 1.3	Bragg wavelength shifting of FBG due to strain	02
Fig 1.4	FBG sensors suitable in high EMI and explosive environments	03
Fig 1.5	Multiplexing of FBG sensors	03
Fig 1.6	FBG sensors are used for structural health monitoring of bridges	04
Fig 1.7	FBG sensors embedded into aircraft wings and body for distributed sensing	04
Fig 1.8	FBG fabrication process	05
Fig 1.9	Light transmission and reflection through FBG	05
Fig 1.10	Bragg wavelength reflection condition	06
Fig 2.1	CCD based Interrogator architecture	08
Fig 2.2	Tunable LASER set-up	11
Fig 2.3	Typical Optical Gain characteristics of Tunable Filter (tuned to a particular passband)	12
Fig 2.4	Voltage vs Passband characteristics of a Tunable Filter	12
Fig 2.5	Tunable Filter based FBG interrogator set-up	13
Fig 2.6	Representational Optical Gain characteristics of Etalon	14
Fig. 3.1	.Block diagram of CCD-based interrogation	21
Fig. 3.2	shows the FBG sensor spectrum (blue line) sampled by CCD pixels at discrete points (red dots)	23
Fig. 3.3	FBG sensor spectrum (blue line) sampled by CCD pixels at discrete points (red dots)	26
Fig. 3.4	Architecture of ANN formed	26
Fig. 3.5	Voltages on CCD pixel array due to optical signal from FBG	31

Fig. 3.6	A simulated FBG peak with central wavelength 740 is sampled, and the Gaussian curve is fitted. The fitted curve's central wavelength is 735	36
Fig. 3.7	A simulated FBG peak with central wavelength 820 is sampled, and the Gaussian curve is fitted. The fitted curve's central wavelength is 825	36
Fig. 3.8	A simulated FBG peak with central wavelength 750 is sampled at intervals of 156 pm	37
Fig. 3.9	Error in central wavelength computed using centroid algorithm when the peak shifts left in wavelength and sampling is done at intervals of 156 pm. The leftward shift in wavelength in pm is the x-axis.	38
Fig. 3.10	The error in central wavelength is computed using the proposed ANN algorithm when the peak shifts left in wavelength. Sampling is done at intervals of 156 pm. The leftward shift in wavelength in pm is the x-axis	38
Fig. 4.1	Block diagram of Tunable Laser-based interrogation	40
Fig. 4.2	The spectral peak dense plot (blue line) is sampled at discrete points (red dot) (target value +32)	42
Fig.4.3	The spectral peak dense plot (blue) is sampled at discrete pts (red dot) (target value: -30)	45
Fig. 4.4	Architecture of ANN formed	46
Fig. 4.5	A simulated FBG peak with central wavelength 100 is sampled at intervals of 26 pm	55
Fig. 4.6	Leftward shift of spectral peak from its nominal position vs error in sub-sample timestamp when the sub-sample timestamp is computed using centroid algorithm	55
Fig. 4.7	Leftward shift of spectral peak from its nominal position vs error in sub-sample timestamp when the sub-sample timestamp is computed using proposed ANN algorithm	56

LIST OF TABLES

Table Number	Table Description	Page No.
Table 3.1	Table showing Pattern-Normalized and Pattern-Centered Pixel Amplitude of FBG sensor spectrum in Fig 3.2	24
Table 3.2	Table showing Normalized and Centred Pixel Amplitude with centroid	25
Table 3.3	Table showing voltage values on CCD pixels	31
Table 3.4	Table showing Pixel Amplitude Value	31
Table 3.5	Table showing Normalized Value	32
Table 3.6	Table showing Centered Normalized Value	32
Table 3.7	Final Training set	33
Table 4.1	Table showing Normalized and Sample Amplitude	44
Table 4.2	Table showing Normalized and Centered time-sample Amplitude with centroid	44
Table 4.3	Table showing Centered Normalized and Sample Amplitude	50
Table 4.4	Table showing input to ANN nodes	51

LIST OF ALGORITHMS

Algorithm Number	Algorithm Description	Page No.
Algorithm 3.1	Pseudocode to generate and save the training data for CCD spectrometer based Interrogation:	27-29
Algorithm 3.2	Pseudocode to build model from the training data for CCD spectrometer based Interrogation	30
Algorithm 3.3	Pseudocode to predict Central wavelength of FBG peak from the pixel data for CCD spectrometer based Interrogation	34
Algorithm 4.1	Pseudocode to generate and save the training data for Tunable Filter based Interrogation	46-49
Algorithm 4.2	Pseudocode to build model from the training data for Tunable Filter based Interrogation	49
Algorithm 4.3	Pseudocode to predict sub-sample timestamp of spectral peak from the time-voltage samples	52

Chapter-1: Introduction

1.1 What is a Fiber Bragg Grating sensor?

Fiber Bragg Grating sensor, also known as FBG sensor, is a passive fiber-optic device formed inside an optical fiber by modulating the core refractive index of the fiber. FBG sensors are formed in a region along the length of the optical fiber by creating alternative bands of higher and lower refractive index zones in the fiber core. The distance between two such consecutive bands is called the pitch of the FGB, and Λ denotes it. This region forming the FBG ranges from a few millimeters to a few centimeters of the fiber, and the pitch Λ is in the order of a few hundred nanometers to a few micrometers.

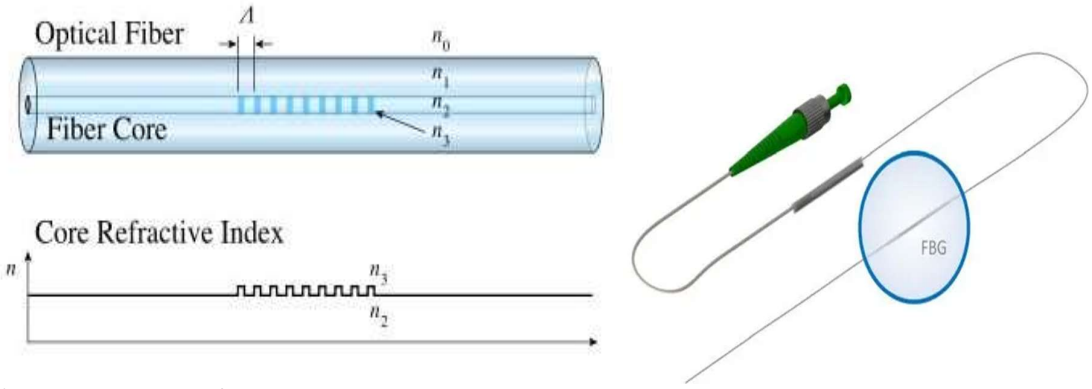


Fig 1.1: structure of FBG [24,31]

These sensors look similar to a normal optical fiber from the outside, but their optical wavelength transmission characteristics differ from those of a normal optical fiber. Unlike a normal fiber-optic cable, an FBG sensor does not allow transmitting all the wavelengths of an incoming optical signal. An FBG sensor acts like an optical notch reject filter in transmission mode, reflecting back a particular wavelength of incoming light while letting all other wavelengths of incident light pass through. Conversely, we can also say that the FBG sensor acts like an optical notch filter in the reflection mode as it reflects a particular wavelength of light when a broadband light is made to incident on it. We refer to this specific wavelength as the 'Bragg wavelength' of the FBG sensor, denoted by λ_B. The value of the Bragg wavelength of an FBG sensor at nominal condition (i.e., at nominal temperature and zero strain) can be controlled at the time of fabrication of the FBG sensor as it depends on pitch Λ and effective refractive index of the fiber core denoted by η_{eff} as per the following equation.

$$\lambda_B = 2\,\eta_{eff}\,\Lambda \qquad \text{(Bragg equation)}$$

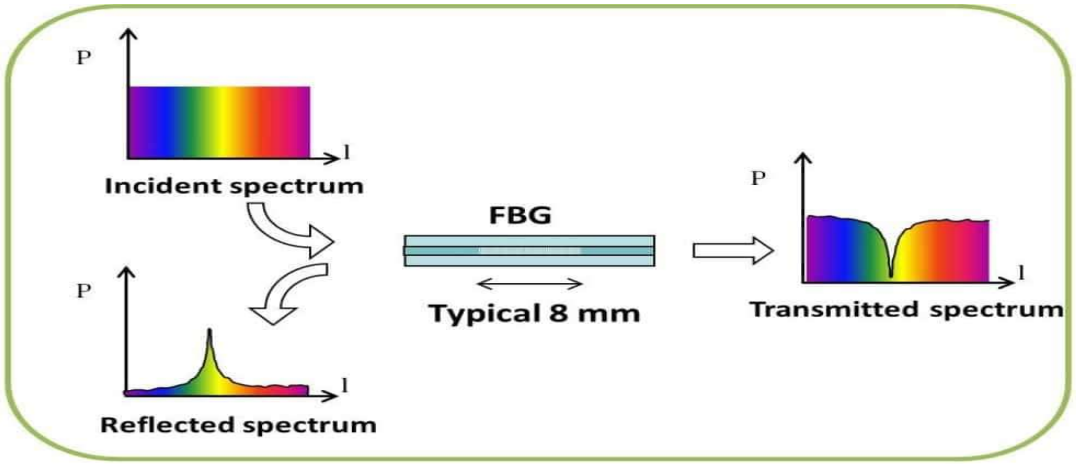


Fig 1.2: property of FBG [25]

However, when the FBG sensor is subjected to a temperature or strain change, its Bragg wavelength is shifted from its nominal value. Therefore, this property of dependence on the Bragg wavelength of an FBG with changes in temperature and strain can be exploited to make sensors out of them. Primarily, we can make a temperature and strain sensor with an FBG. However, other parameters, such as acceleration, velocity, displacement, force, stress, etc., can also be measured with an FBG sensor as derived parameters, which can be converted to a strain parameter with suitable packaging. Nowadays, FBG sensors are also used for chemical and biological sensing by applying a suitable chemical coating on them so that any change in these parameters causes a change in the temperature or strain value, which can be measured using an FBG sensor. For example, we coat an FBG sensor to measure moisture using a hygroscopic material. Such material absorbs water molecules, expands when there is more moisture in the environment, straining the FBG, releases the water molecules, and contracts when the humidity is less, easing the strain on FBG. Thus, the moisture information can be converted into strain information and measured using an FBG sensor.

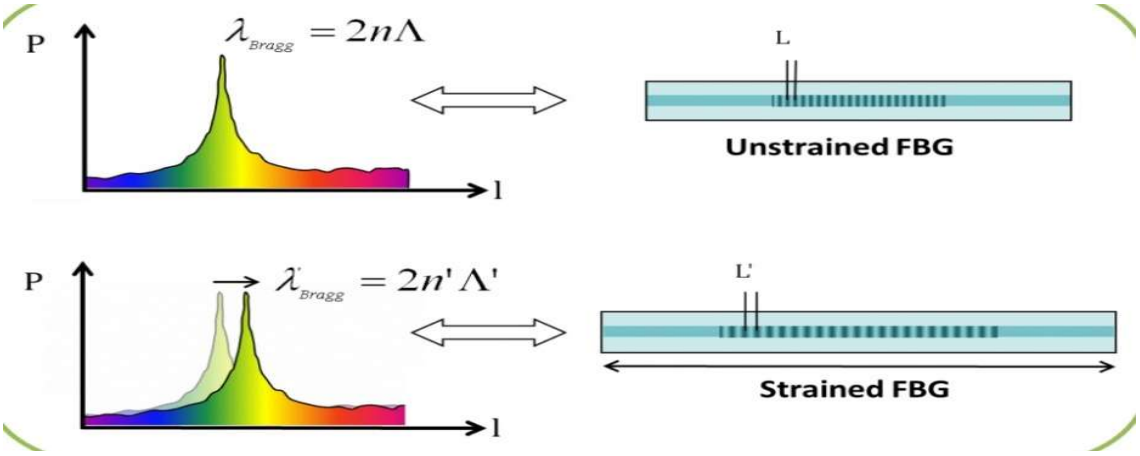


Fig 1.3: Bragg wavelength shifting of FBG due to strain [25]

1.2 Where do we use FBG sensors, and Why?

FBG sensors find their applications in situations or environments where traditional sensors like electronic sensors either do not work or cannot be used.
One such environment is the places with high electromagnetic interference (EMI). In high EMI environments, such as inside an electric power generator, transformer windings, high-tension transmission lines, railway pantographs, etc., conventional electronic sensors don't work. FBG sensors, being fiber-optic type sensors, are immune to EMI and can be used in such environments.

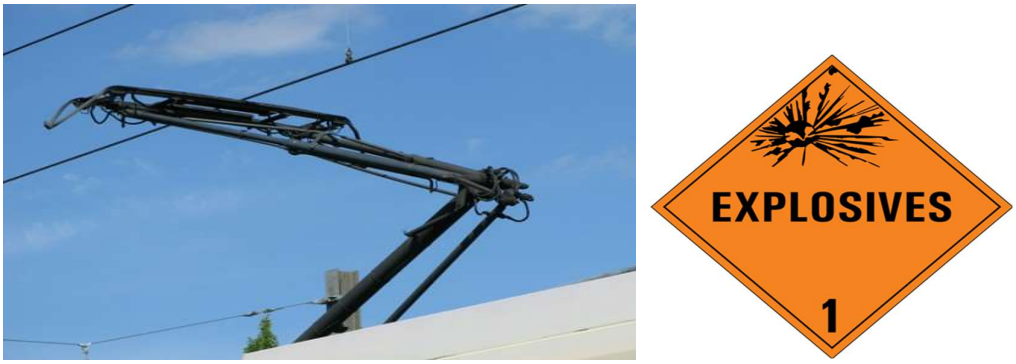


Fig 1.4: FBG sensors suitable in high EMI and explosive environments [32.33]

Again, using electronic sensors in highly explosive environments, such as inside a coal mine, is risky, where highly flammable gasses like methane are present. Electronic sensors are generally active devices, and they need electricity to run. In such sensors, there is always a risk of electric discharge that may pose a significant danger in such environments. Here, FBG sensors, being passive devices with electricity-free operation, are very suitable for application.

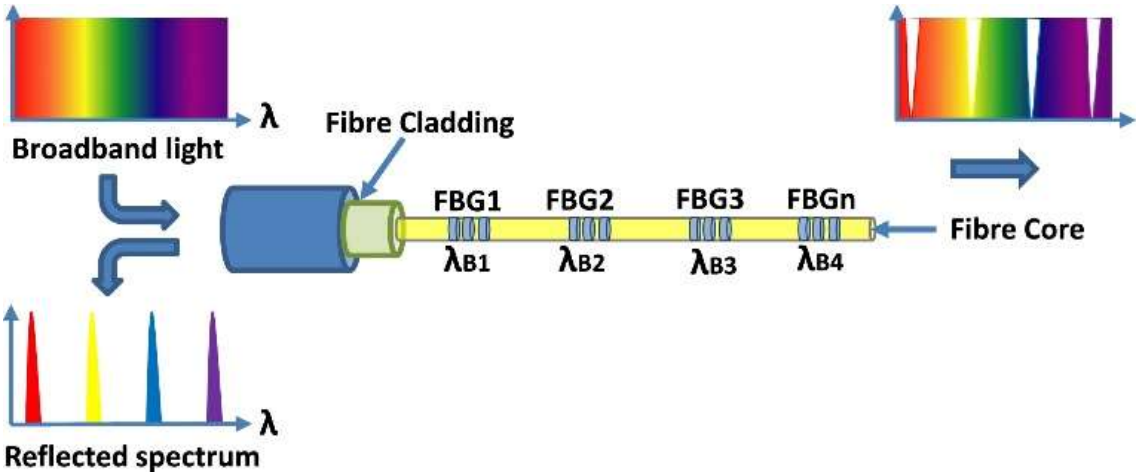


Fig 1.5: Multiplexing of FBG sensors [26]

In some applications like distributed sensing, electrical sensors are unsuitable because they generally do not offer multiplexing capabilities. In applications requiring many sensors to map many points, each electrical sensor sends its data on a separate wire. This dramatically increases the cabling overhead and may become difficult to manage. On the other hand, as multiple FBG sensors can be multiplexed in a single optical fiber, the cabling overhead is reduced significantly. That is why FBG sensors are becoming very popular in structural health monitoring applications, where we monitor multiple critical points of a bridge or building.



Fig 1.6: FBG sensors are used for structural health monitoring of bridges[34]

Another significant advantage of an FBG sensor is its slender and flexible shape. As an FBG is inscribed inside the core of an optical fiber, the sensor has the exact dimension of the optical fiber with a 125-micron diameter. Therefore, these FBGs are often embedded in carbon fiber reinforced polymer (CFRP) structures, such as an aircraft wing, spacecraft body, etc. The CFRP and FBG have similar dimensions (125 um), so the sensor is seamlessly integrated into the structure. This is a unique advantage that no electronic or mechanical sensor can provide.

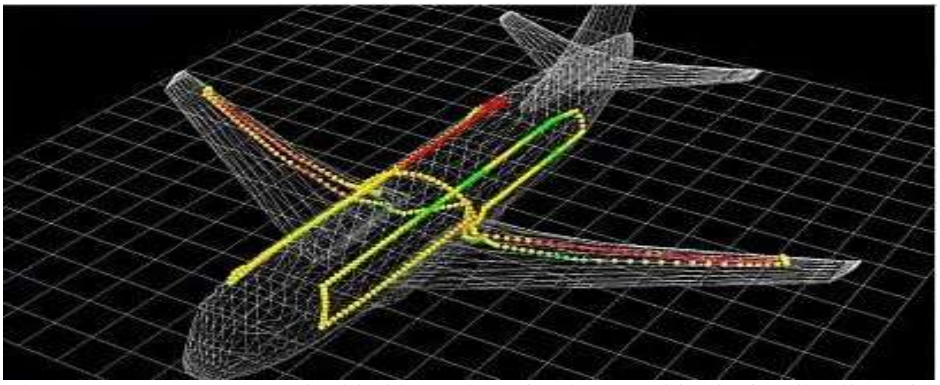


Fig 1.7: FBG sensors embedded into aircraft wings and body for distributed sensing[35]

Therefore, we can see that FBG sensors, due to their properties like immunity to electromagnetic interference, passive electricity independent working principle, slender physical shape, lightweight, high multiplexing capability, and high sensitivity, are suitable in various situations where other types of sensors are either unusable or impractical.

1.3 Fabrication of FBG sensors

Fabricating Fiber Bragg Gratings (FBGs) typically involves a process known as photo-inscription. Initially, a segment of single-mode optical fiber, usually silica, is selected. This fiber is then exposed to a pattern of intense ultraviolet (UV) light generated by a phase mask or interferometric setup. The UV light interacts with the fiber core, inducing a periodic modulation of the refractive index along its length. This modulation creates alternating higher and lower refractive index regions, forming the grating structure. The interference pattern between the incident light and the phase mask or interferometer determines the precise periodicity of the grating. After fabrication, the FBG may undergo optional annealing for stabilization. Once fabricated, FBGs can be packaged with protective coatings and connectors for integration into sensing systems.

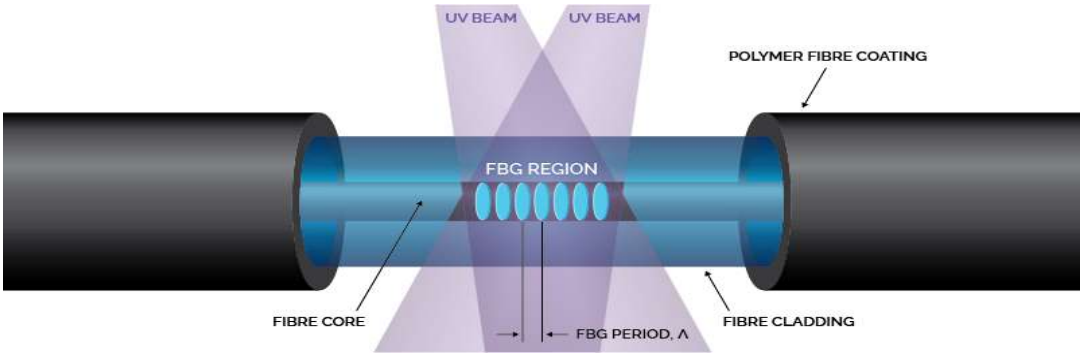


Fig 1.8: FBG fabrication process [27]

1.4 Working principle of FBG sensor (derivation of Bragg equation)

When a broadband light (say spanning a wavelength range from 1500nm to 1600nm) is launched in the fiber core, it travels through it obeying the principle of 'Total Internal Reflection' of light and is incident on the alternating high and low refractive index bands of the FBG. At each boundary between a higher and lower refractive index band, a portion of the light incident on the boundary is reflected, and the remaining portion is transmitted, as shown in Fig 1.9.

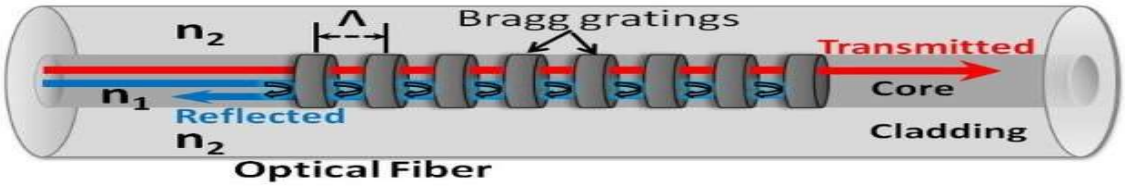


Fig 1.9: Light transmission and reflection through FBG [28]

However, not all photons of light are reflected equally. If the wavelength associated with a photon of light in the fiber λ_c is such that it is double the pitch (Λ), i.e., $\lambda_c = 2\ \Lambda$, and it reflects from the boundary surface, then it will be in constructive interference with the photon having same wavelength that is reflected from the next such boundary. This is because the path difference between the two photons reflecting from consecutive boundaries is always equal to 2Λ as the photon reflecting from the further boundary covers an additional distance which is twice the distance between the boundaries (which is the same as pitch), once while going

forward and then while coming backward after reflection. So, if the wavelength (λ_c) associated with both these photons in the fiber is also the same and equal to 2Λ, then the two reflecting photons will undergo constructive interference as their path difference (2Λ) is equal to their wavelength (λ_c) because of the relation mentioned above $\lambda_c = 2\Lambda$. Similarly, all photons having a wavelength λ_c getting reflected from boundaries that are spaced at an interval of Λ also undergo constructive interference as their path difference is equal to some multiple of 2Λ which means their path difference is equal to some multiple of their wavelength in the fiber λ_c. These photons (having an associated wavelength of λ_c), which satisfy this condition of constructive interference for reflection, mainly get reflected from the FBG, while all other photons that do not fulfill this condition primarily get transmitted through the FBG. This is shown in the Fig: 1.10.

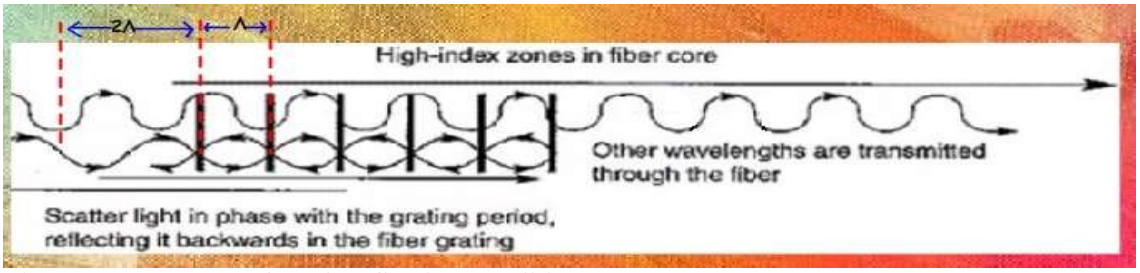


Fig 1.10: Bragg wavelength reflection condition [29]

The effective refractive index of the fiber through which this photon with wavelength λ_c in fiber is propagating is $\eta_{eff.}$ If the wavelength of the same photon in free space is denoted by λ_B, then we have the relation as per Snell's law: $\lambda_B = 2\eta_{eff}\,\lambda_c$. Combining this with the earlier equation, $\lambda_c = 2\Lambda$, we get the Bragg equation $\lambda_B = \eta_{eff}\,\Lambda$.

1.5 Interrogation of an FBG sensor

Measuring the 'Bragg wavelength' (λB) of the FBG sensor is called the 'Interrogation of FBG sensor'; the device performing this process is called the 'FBG Interrogator.' Typically, an FBG spectrum has a spread of a few hundred picometres, whereas the desirable accuracy of an FBG sensor is within 5 to 6 picometres; therefore, for every FBG spectral peak, to find the Bragg wavelength of an FBG sensor, we need to find the central wavelength of its spectral peak. Moreover, it is generally impossible to sample this FBG spectrum densely at a resolution of every pico-meter in real-time applications while the FBG sensor is deployed in the field. Therefore, a significant step in interrogation is to sample the FBG spectrum sparsely (say at a resolution of 100pm to 160pm) and predict the central wavelength of the FBG spectrum by using some algorithm. Many algorithms have been used to achieve this. An algorithm that can predict the central wavelength more accurately from a given set of samples is considered a better algorithm for interrogation. Other parameters to compare the interrogation algorithms are their time complexity, space complexity, hardware or FPGA implementability, etc.

Chapter 2: Literature Survey

Several fiber Bragg grating (FBG) interrogation methods have been explored in the existing literature. However, as discussed in [6] by Daniele Tosi, two setups (or hardware platforms) have become particularly popular for the FBG interrogation: the first setup is based on a broadband source and a spectrometer (often also referred to as FBG analyzer), and the second setup is based on a fast-scanning tunable laser (or monochromatic light) that sweeps a relatively broad wavelength range (typically 40–100 nm) and a photodetector. This scanning tunable laser (or monochromatic light) uses a tunable filter at its heart; hence, it is referred to as a tunable filter-based interrogation set-up in this document. Both of these interrogation set-ups have been developed and commercialized.

In a spectrometer-based interrogation set-up, the reflected light from an FBG is directed onto a one-dimensional CCD array through a diffraction grating. The wavelength of the reflected light dictates which set of pixels on the CCD array is illuminated and to what extent. By analyzing the optical spectrum distribution across the pixels, the Bragg wavelength of the light can be determined. In a tunable filter-based interrogation set-up, a laser or monochromatic light with a narrow bandwidth is swept across the entire range of wavelength spectrum. The Bragg wavelength is then determined from the time-stamp of the samples of the reflected optical signal from the fiber Bragg grating (FBG) as recorded by a photodetector. Both techniques are discussed in more detail in the following section.

2.1 CCD Spectrometer-based Interrogation Architecture and Working principle:

The CCD spectrometer-based technique has been discussed in [1, 4, 8]. This technique's reference is also available for commercialization in [9, 10, 11, 12] and [13-15]. This spectrometer-based FBG interrogation set-up comprises the following components:

(i) **Broadband Light Source**: This wideband light source launches IR light into optical fiber from 1510nm to 1590 nm in the continuous band. This light source is connected at the input of port 1 of the circulator.

(ii)**Optical Circulator:** Optical circulator is a three-port passive optical device out of which 'port 1' is an input port, 'port 3' is an output port, and 'port 2' is an input and output port. The optical signal applied to port 1 is output to 'port 2'. This 'port 2' is connected to 'Fiber Bragg Gratings sensors. Thus, the optical signal applied to 'port 1' of the circulator comes out from its 'port 2' and is incident on these 'Fiber Bragg Gratings'. The light reflected from these Gratings (i.e., the Bragg wavelengths of these Gratings) get back into 'port 2' of the circulator as input-output to port 3. The production of port-3 is connected to the input of CCD-based spectrometers.

(iii) **CCD-based spectrometer:** A CCD spectrometer is a type of spectroscopic instrument that uses a Charge-Coupled Device (CCD) as a detector to measure light intensity as a function of its wavelength or frequency. The spectrometer typically consists of a slit to allow a narrow

beam of light to enter, a collimator to create a parallel beam of light, a diffraction grating to disperse the light into its component wavelengths, and a CCD detector to record the intensity of light at each wavelength. The diffraction grating separates the light into its component wavelengths, and the CCD detector records the intensity of each wavelength. In a typical setting, an array of 512 CCD detectors is arranged in a single dimension, and each CCD detector positioned in an array is generally referred to as a CCD pixel or just a Pixel. Each incoming Bragg wavelength illuminates adjacent pixels determined when manufacturing the device.

(iv)**Fiber Bragg Gratings**: On the port-2 of the optical circulator, a series of FBGs are connected. Each FBG in the series has a different value of nominal Bragg wavelength, which is achieved by fabricating each grating in the optical fiber with a different periodicity Λ. The nominal Bragg wavelength of any FBG is the value of its Bragg wavelength when subjected to an ambient temperature of 25°C, and there is no strain on it. Now, when any FBG is subjected to any change in temperature (from nominal 25°C) or strain, its Bragg wavelength changes from nominal. This change in the Bragg wavelength is called the 'Bragg Wavelength Shift'. FBGs can be used as sensors by subjecting them to direct indirect shifts in temperature and strain, measuring their change in Bragg wavelength from nominal, and then referring to a calibration table. Therefore, the primary operation in an FBG Interrogator is to determine the Bragg wavelength of the grating.

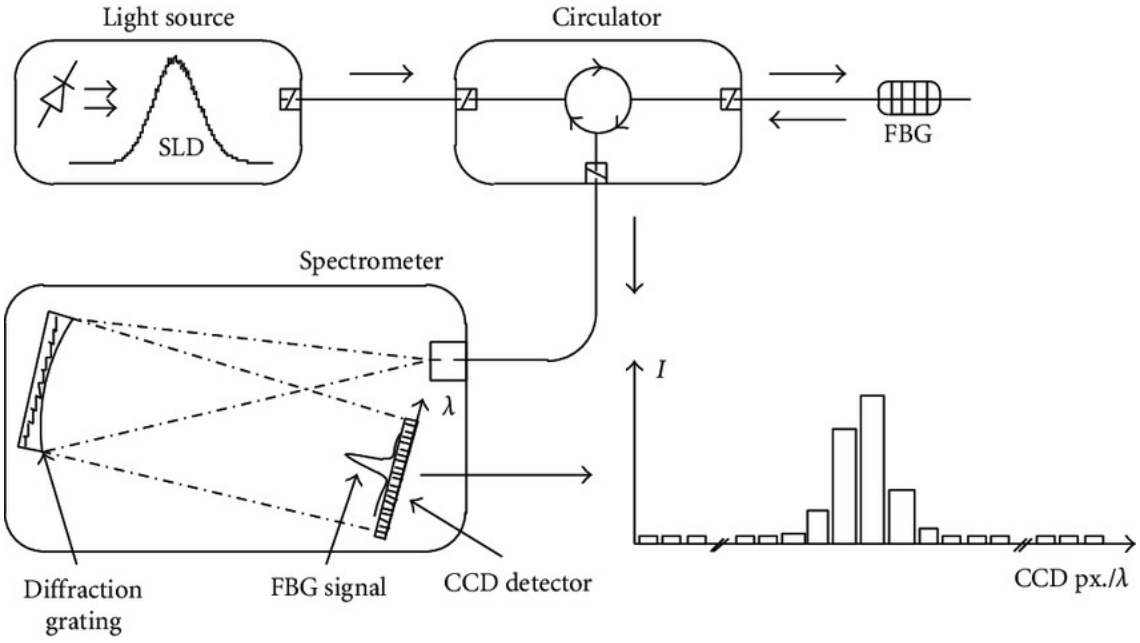


Fig 2.1: CCD-based Interrogator architecture [30]

(v)**Data Acquisition module**: When illuminated with optical signals, the CCD pixels give up electrons and become positively charged (photoelectric effect) and hence get a positive voltage. This positive voltage of each pixel is sampled by ADC and recorded as a pixel number vs ADC count table, which is passed to the software for further processing. The software module receives the pixel number vs. ADC count table, and it runs a curve-fitting algorithm on this data to determine which subpixel of the pixel array corresponds to the crest of the optical signal peak (or the Bragg peaks). It may be emphasized here that as the optical wavelength range of 80nm spans over a linear one-dimensional array of 512 pixels and we need a resolution of around 5 pm, therefore for a given FBG peak, we need to determine its corresponding subpixel

up to two decimal places. Several algorithms are used to determine the subpixel value (which corresponds to the Bragg wavelength of the FBG) from the pixel vs. voltage value of the CCD pixel array.

(vi)**Algorithms for Bragg Wavelength determination for a CCD spectrometer-based Interrogator**

The optical signal from the FBG gratings is applied to the CCD pixel array (arranged spatially in a one-dimensional line) through a diffraction grating. The diffraction grating distributes the optical signal on the CCD array so that one end of the Grating array is illuminated by the lowest wavelength of the spectrum and the other end of the CCD array is illuminated by the highest wavelength of the spectrum. All other intermediate wavelengths of the spectrum illuminate the pixels in the intermediate sections of the CCD array such that there is a monotonic relationship between pixel number and wavelength of the optical spectrum. This exposure of the optical signal from FBG sensors on the CCD array is kept for some time, during which voltage builds up in the pixels depending on the optical signals it receives. It should be noted that the optical signal received by any particular pixel depends on its position in the array and the signal strength of the optical wavelength in the spectrum corresponding to that position in the array. After the voltages are built in each pixel, the exposure of the optical signal on the CCD array is withdrawn, and the voltage values of each pixel are scanned and digitized by ADC. Now, from the voltage distribution of pixels, the corresponding sub-pixels of the FBG peaks are estimated using an estimation algorithm to determine the correspondence between the voltage distribution over the pixels and its corresponding sub-pixel and hence Bragg wavelength. Researchers have proposed different estimation algorithms for determining the central wavelength of the FBG spectral peak (i.e., Bragg wavelength).

In [1], J. Alves et al. discussed the 'Central Detection Algorithm' (CDA) and 'Polynomial Approximation Algorithm' (PAA) for sub-pixel determination. The CDA is a centroid computation-based algorithm that computes the 'mass-center' of the peak, considering all the points that compose the reflection spectrum of the grating by the following formula.

$$\lambda_B = \frac{\sum_k \lambda_k i_k}{\sum_k i_k}$$

where λ_B is the Bragg wavelength and λ_k and i_k represents the center wavelength and the intensity of the light incident on the k^{th} pixel, respectively

In this method, while computing the Bragg wavelength λ_B, only those pixels are considered where the intensity value is above a particular threshold value. This threshold value is called the 'Minimum Acquisition Limit' in the research paper. In this paper, it has been explained that below this 'Minimum Acquisition Limit' threshold value, all points are rejected because the SNR is considered too low. Such a minimum threshold is also required to separate one FBG from its adjacent peaks. Another algorithm used for the subpixel determination was PAA, a curve-fitting-based algorithm. It is mentioned in the paper that a 2nd-order to 4th-order polynomial fit was performed for the FBG spectral peak. However, it is also noted in this paper that the polynomial curve fitting algorithm are iterative in nature and not suitable for applications in which there is a requirement for stringent time performance.

In another work in [4], Thomas Bodendorfer et al. compared different peak detection algorithms with CCD spectrometer-based FBG interrogation systems. Here, they compared several sup-pixel finding and Bragg wavelength detection algorithms such as (i) Maximum,

(ii) Centroid, (iii) Gaussian fit, (iv) Quadrature fit, and (v) Linear phase operator. The Maximum sub-pixel finding algorithm does a cubic spline interpolation between the sampled points. Then, the algorithm's output is obtained by searching for the maximum wavelength of the interpolating data. The Linear Phase Operator algorithm combines FIR filtering and linear interpolation over the samples to determine the Bragg wavelength. The Gaussian fit and Quadrature fit algorithms try to fit a Gaussian curve and second-order polynomial curve over the sampled points, respectively, and then determine the sub-pixel or wavelength corresponding to the maxima of the FBG peak. So, these algorithms are mathematically quite complex compared to the centroid algorithm and require interpolation, curve fitting, and filtering operations. On the other hand, although the centroid algorithm is mathematically more superficial and can be implemented on FPGA hardware and free from iterations, it was noted in the paper as having worse performance than other algorithms.

In another work mentioned in [8], Timur Agliullin et al. attempted the same problem of central wavelength (i.e., Bragg wavelength) determination of fiber optic FBG sensors using the ANN algorithm for a CCD spectrometer-based FBG Interrogation System. This work used a 512-pixel IMON 512 as a CCD spectrometer. The FBG peak spectrum Bragg wavelength was varied over a fixed range of 3nm, and the voltage values acquired at the pixels were used to train a three-layered ANN comprising 19 nodes in the input layer, ten nodes in the hidden layer, and a single node in the output layer. The trained ANN model predicts Bragg wavelength using the ANN, as the problem is conceived of as a regression problem. However, once the model is trained in the 3nm bandwidth and the FBG spectrum moves out of the 3 nm optical band used for training due to some pre-strain, etc., the same trained model no longer applies for prediction. In a real-life scenario, the intensity of the optical signal from the FBG sensors may vary depending on the source intensity variation and channel attenuation variation over time. This algorithm does not discuss how such variations will be handled while training the network or predicting the central wavelength.

2.2 Tunable Filter-based Interrogation Architecture and Working principle:

The tunable filter-based interrogation set-up is described in [2, 5, 7]. This technique and its industrialization are referenced in [16 - 23]. This spectrometer-based FBG interrogation set-up comprises the following components:

(i) **EDFA-based ring Laser Source**: An EDFA Laser source is set up for this type of interrogator, which outputs a laser light in the required optical band of 1530nm to 1570nm range, also known as the C band. This set-up consists of a 980nm laser diode, which pumps an Erbium-doped active fiber through a WDM coupler. The electrons in the Erbium-doped fiber are excited to a higher energy state by the photons from the 980nm laser. Without any additional stimulus, these excited electrons emit photons in the entire C band region, forming an Amplified Spontaneous Emission (ASE) Spectrum in the C band. However, when a monochromatic light having a particular wavelength in the C band is applied to the input of the Erbium-doped active fiber as a stimulus, most of the excited electrons emit photons matching with the input monochromatic light wavelength, making a LASER in that wavelength. This additional stimulus of monochromatic light is applied to the Erbium-doped active fiber through the other branch of the WDM coupler input, which is connected to the Tunable filter, and its passband wavelength determines the laser output wavelength. The topology of the ring laser circuit is shown in the figure below. The optical output of the Erbium-doped fiber is given to a

WDM splitter, which splits this optical output into two parts so that unused 980nm light is discarded and the valuable light in the range 1530nm to 1570nm is applied to the Tunable Fabry Perot filter through a fiber-optic isolator. Fiber optic isolators are fiber-optic devices that allow light to pass through it in only one direction (direction as shown by arrowhead) and are used to prevent the back reflection of light in the circuit. The output of the Tunable filter is applied to a 1x2 fiber-optic splitter through another isolator, which outputs a part of the optical signal (10%) and loops the remaining portion of the optical signal back into the ring laser circuit.

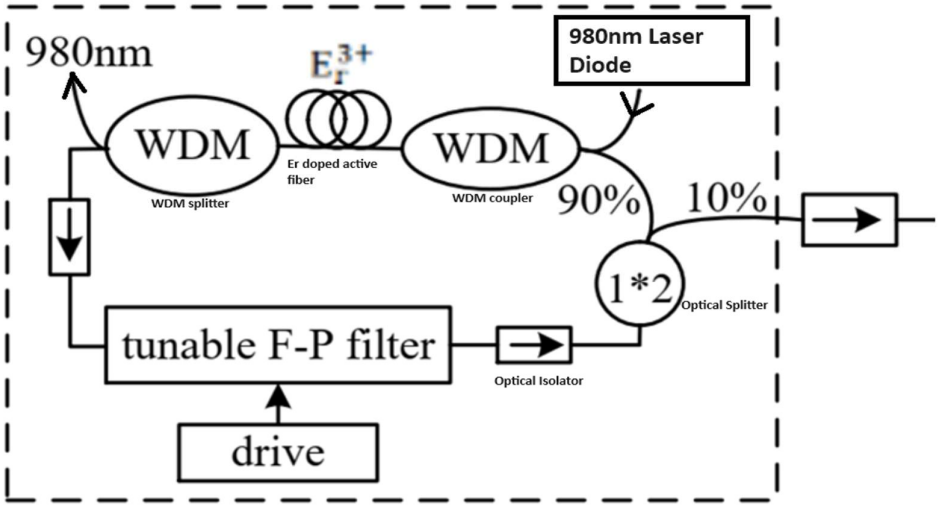


Fig 2.2: Tunable LASER set-up [7]

(ii) **Tuneable notch pass optical filter**: This is an optical Fabry Perot filter from 'Micron Optics' (model FFP-TF2). This device has two optical ports (one input and one output) and one electrical input port. In this circuit, along with the laser, there is some residual broadband ASE in the C band that is input to it. A voltage signal between 0V and 20V is applied to its input electrical port from its controller. Being a notch-pass optical filter, it allows only a monochromatic (single wavelength) beam of light to be transmitted through its optical output port and blocks all other wavelengths input to it through its input port. Among all the wavelengths of the ASE spectrum light applied to its input port, the wavelength gets transmitted to its output port and is determined by the value of the voltage applied to its electrical port. In the Micronoptics FFP-TF2 filter that is used in this setup, for higher voltages (20V), the pass band of the filter is tuned near lower wavelengths (i.e., around 1520nm), and for lower voltages (0V), the pass band of the filter is tuned near higher wavelengths (i.e., around 1570nm). So, the voltage vs passband wavelength relation is monotonic (pass band wavelength reduces with increasing voltage). However, there is neither linearity nor repeatability in this relation, and these limitations are discussed in [2] and [7].

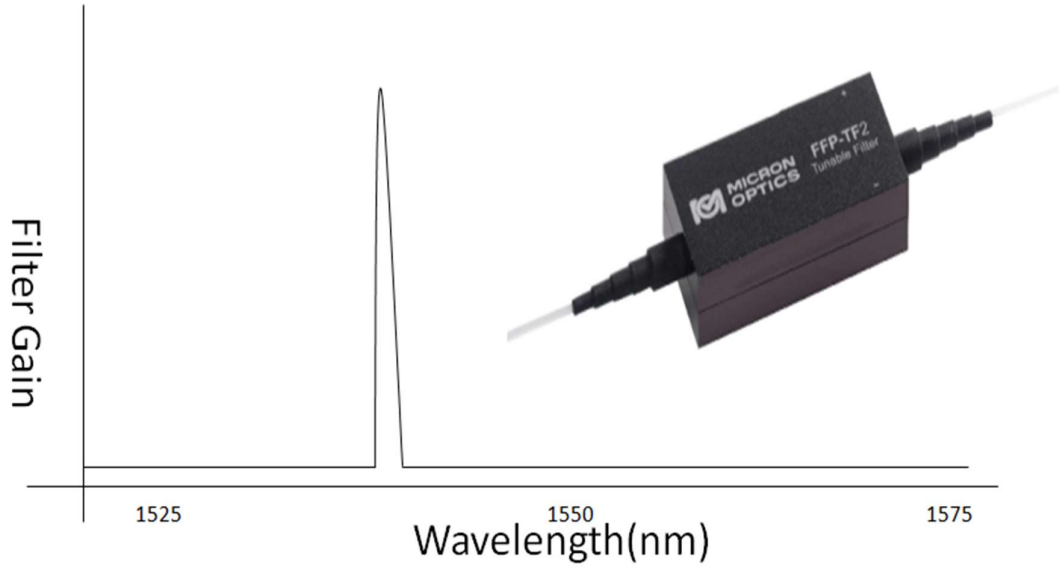


Fig 2.3: Typical Optical Gain characteristics of Tunable Filter (tuned to a particular passband)

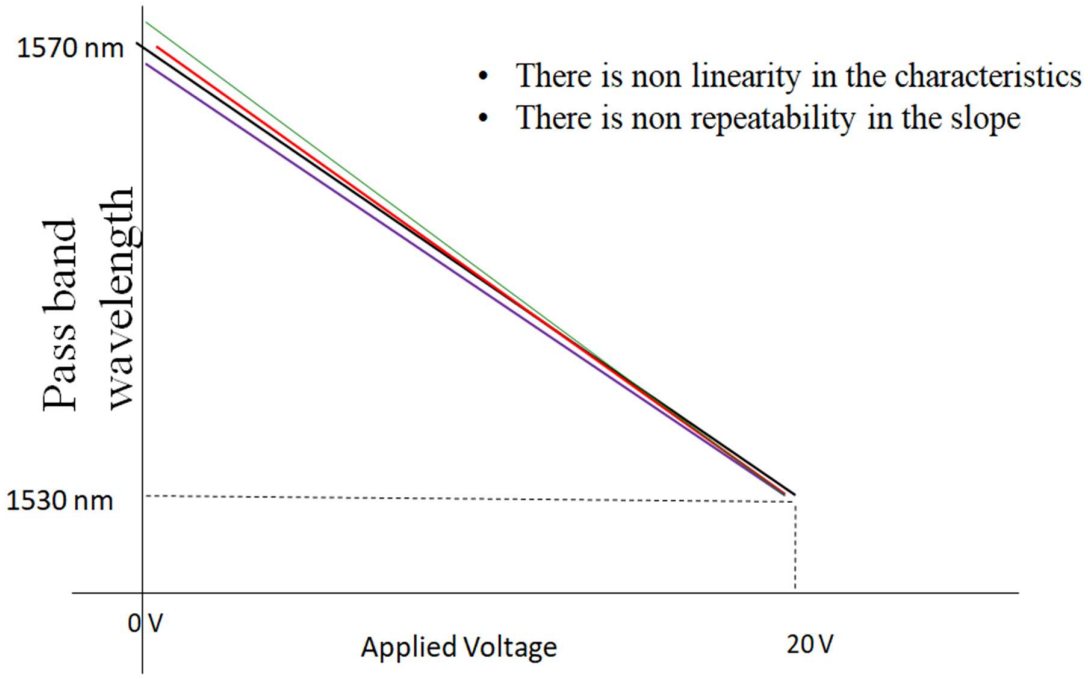


Fig 2.4: Voltage vs Passband characteristics of a Tunable Filter

The Controller of this FFP-TF2 Filter is a function generator that generates a triangular voltage signal from 0V to 20V to sweep the tuneable filter passband at a user-selected frequency. Along with the triangular wave, it generates a square sync wave in phase with it for triggering purposes. This square wave is High (i.e., 5V level) at the falling slope of the triangle wave and Low (i.e., 0V level) at the rising slope of the triangle wave.

Now, when the controller of the tunable filter generates a sweeping voltage from 0V to 20V or vice-versa, the scanning wavelength LASER is output sweeping from 1570nm to 1530nm or vice-versa. This report calls this scanning wavelength-swept LASER a Tunable LASER Source. Additional components like athermal etalon are used in the circuit, as shown in the schematic diagram below, to address non-linearity and non-repeatability issues.

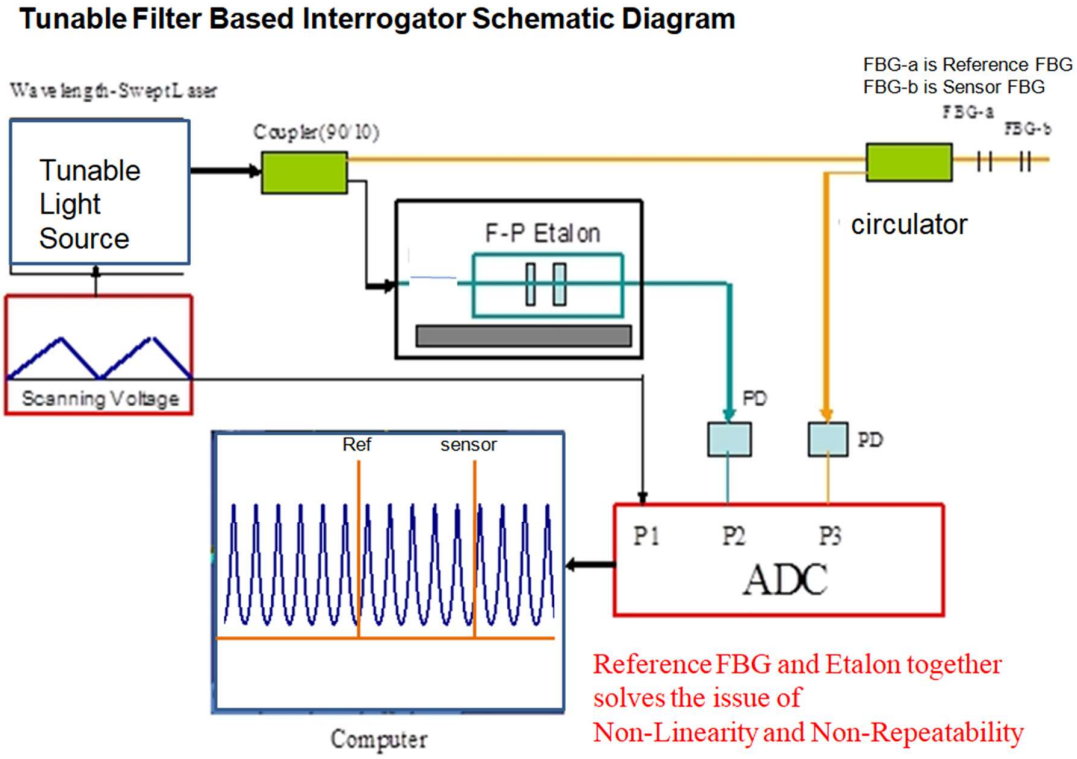


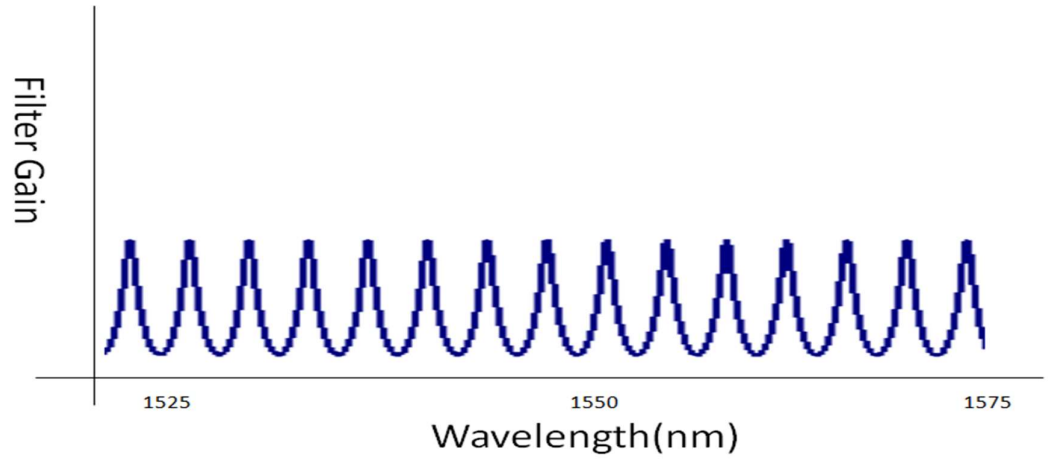
Fig 2.5: Tunable Filter based FBG interrogator set-up

(iii)**Optical Splitter:** This is a port optical device with one optical input port and two optical output ports. This device bifurcates the input scanning wavelength LASER into two branches in the ratio of 90:10, each going to one output port. An optical circulator applied 90% of the LASER to a series of fiber bracket gratings. The remaining 10% of the LASER is applied to an 'Athermal Etalon' (another optical component described later).

(iv)**Optical Circulator:** The optical circulator is a three-port passive optical device. In this circuit, the scanning wavelength tunable LASER (coming from 90% output branch of the splitter) is applied to port 1 of the circulator so that it is incident on the 'Fiber Bragg Gratings' that is connected to port 2 of the circulator. The light reflected from these Gratings (i.e., the Bragg wavelengths of these Gratings) get back into 'port 2' of the circulator as input-output to port 3. The production of port-3 is connected to a photodetector.

(v)**Athermal Etalon**: Athermal Etalon is a two-port optical device with one input and output port. It is a multiple-notch pass optical filter. It allows only specific predetermined optical wavelengths to pass through and blocks all other wavelengths. Only the optical wavelengths of the ITU-T grid get transmitted through this filter, and all other wavelengths get blocked. The output 2 of the optical splitter is applied to its input, and its output is connected to a photodetector. This device is termed 'Athermal' since any ambient temperature change does not affect its passband's wavelengths.

Fig 2.6: Representational Optical Gain characteristics of Etalon



(vi)**Fiber Bragg Gratings**: On port 2 of the optical circulator, a series of FBGs are connected. The series has two types of FBGs: one Reference Fiber Bragg Gratings and one or more FBG sensors.

- **Reference Fiber Bragg Grating**: Out of all the Fiber Bragg Gratings connected to the port-2 of the circulator, one is always kept in its nominal state (or almost in the nominal state). It is taken care that the Bragg wavelength of this FBG does not change much. This is done by keeping it unstrained and protecting it from sizeable ambient temperature shifts. This is the only FBG in the system that is not used as a sensor.
- **FBG Sensors**: All other FBGs apart from the 'reference grating' are used as sensors by subjecting them to direct indirect shifts in temperature and strain, measuring their shift in Bragg wavelength from nominal, and then referring to a calibration table. So, the primary operation in an FBG Interrogator is to determine the Bragg wavelength of the grating. How this is done is explained in the 'working principle' section.

(vii)**Photo detector module:** Two separate Photodetectors (PD1 and PD0) are used to acquire optical signals from two branches. PD1 is connected to port 3 of the optical circulator, whereas PD0 is connected to the output port of Etalon. Photo detectors optical are intensity-measuring devices. Its input is an optical signal, and its output is a voltage signal. The higher the intensity of the optical signal input, the higher its output voltage signal value.

(viii)**Data Acquisition module**: The output voltages of a photodetector are sampled by an ADC and can be further processed by a signal processing circuit implemented on a

microprocessor or FPGA. The sync signal from the tunable filter controller circuit is also fed into the signal processing circuit as a digital input. With all this information, the signal processing circuit is required to predict the Bragg wavelength for each FBG sensor for interrogation.

When we apply a triangular wave to the tunable filter's electrical input port, we get a tunable LASER sweeping across the wavelength range of the filter (i.e., 1530nm to 1570nm). The splitter bifurcates This LASER output and is applied to PD1 through FBGs and circulator in the first branch (90%) and to PD0 through Etalon in the second branch (10%). Whenever the wavelength of the sweeping LASER matches any of the Bragg wavelengths of the gratings in the first branch, we get a high voltage output for PD1. Similarly, in the second branch, whenever the wavelength of the sweeping tunable LASER (i.e., the passband of tunable filter) matches with any one of the pass bands of Etalon (i.e., any of the ITU-T grid wavelengths), we get a high voltage output for PD0. So, if we draw a time domain graph plot for 'photodetector output voltage' vs. 'time' (with t_0 at the start of the rising slope of the triangular wave of tunable filter), we will get one peak for each FBG in PD1 and one peak for each ITU-T grid wavelength in PD0. In such time domain plots, the timestamp of the tip of the peak will correspond to a particular wavelength. We need to find the peak tip precisely as we need a resolution of 5 to 10 pm in wavelength, whereas the FWHM (full width at half maximum) of these peaks are around 200pm to 300pm. The ADC generally is not able to do dense sampling (i.e., sampling at every pico-meter shift of the LASER wavelength) for high scan rate applications, and there is a requirement to predict the central wavelength of the spectral peaks of FBG sensors and etalon passbands from the sparsely sampled time domain spectrum. This is done by predicting the sub-timestamp value corresponding to the maxima of the spectral peak from the timestamp of the samples recorded by ADC. This is similar to finding the sub-pixel from the pixel value in a CCD spectrometer-based interrogator. By finding the sub-sample timestamp from the timestamps of the acquired ADC samples, the resolution of the ADC is synthetically increased. After the sub-timestamp values for each spectral peak are determined, the Bragg wavelength of each FBG sensor can be computed from them. There are several algorithms for the sub-sample timestamp determination and Bragg wavelength computation, as discussed in the following section.

(ix)**Algorithms for Bragg Wavelength determination for a Tunable filter-based Interrogator**

There are two parts to the algorithm to determine Bragg wavelength from the timestamps of the ADC samples. The first step is to predict the sub-sample timestamp value corresponding to the maxima of the spectral peak from the timestamp of the samples recorded by ADC. Then, the second step computes the Bragg wavelength of the FBG sensor from its sub-timestamp value, which was obtained from the first step. In the case of CCD spectrometer-based interrogation, the second step was very trivial because the relation between sub-pixel to wavelength is a repeatable relation that does not change from one scan to another. So, we can have a look-up table to convert the sub-pixel value to Bragg wavelength value. But in the case of tunable filter-based interrogation, there is no repeatability between the sub-sample timestamp value and the Bragg wavelength value (the same as the tunable LASER instantaneous wavelength value) of the FBG sensor. So here, the Bragg wavelength of the sensor FBG is determined with the help of 'sub-sample timestamp vs reference FBG Bragg wavelength relation' and 'sub-timestamps vs etalon passband wavelengths relation' as both reference FBG Bragg wavelength value and Athermal etalon passband wavelength channels

have known and constant values. The following section explains both parts of the algorithm in detail.

Several algorithms have been discussed in the literature to compute the sub-timestamp from the timestamps of the sampled ADC values (i.e., the first step, as discussed before). In [5], Lucas Negri et al. discussed and compared several 'peak detection' algorithms for tunable filter-based interrogation systems. Here, 'peak detection' means predicting the position of the maxima or crest of the peak from the position and values of the acquired samples. This means finding the sub-sample timestamp from the values and timestamps of the acquired samples in our context. The first algorithm discussed is the 'Maximum algorithm' in which the timestamp of the maximum valued sample is assumed to be the sub-sample timestamp of the spectral peak. As expected, this algorithm is reported to be very inaccurate. The second algorithm discussed is the 'FIR filter-based algorithm' discussed in both [5] and [7]. In this algorithm, the spectral peak is smoothed (to remove noise) and then differentiated so that the sub-sample timestamp of the maxima of the spectral peak can be identified by finding the zero crossing of the differentiated peak by the interpolation method. The third popular algorithm is the centroid algorithm, which is discussed in both [5] and [7] and which is similar to the one discussed earlier in the case of the CCD spectrometer-based interrogator, the only difference being that here we determine sub-sample timestamp from sample values and timestamps instead of sub-pixel from pixel values. Centroid algorithm has the advantage that it is straightforward and can even be implemented in an FPGA. However, the negative side is that it tends to be less accurate and shows systemic error when sparse sampling is done. Another method discussed in both [5] and [7] is the curve fitting method, which is generally quite accurate but unsuitable for time-critical and real-time applications as it is an iterative algorithm requiring several iterations to find the best-fitted curve. In [5], an Artificial Neural Network (ANN) based peak detection (sub-sample timestamp in our context) algorithm has also been proposed. It has been argued that as ANN is a universal function approximator, it can train a network where the sub-sample timestamp of the FBG peak will be output when the sample values and their timestamps are fed into the network. However, in this proposal, the complexities arising from issues like non-linearity and non-repeatability of 'voltage vs Tunable LASER wavelength' as discussed in [2] and [7] are neither discussed nor addressed. As the slope of the 'voltage vs Tunable LASER wavelength' may change from one sweep of the Tunable LASER to another, the same spectral peak may be captured by different samples. Also, the amplitude of spectral peak may change (with shape unchanged) from one scan to another due to variation in channel attenuation (i.e., attenuation of optical signal getting transmitted through the fiber), which is very common in fiber optic circuits. These anomalies were also not incorporated in the training methodology of the described ANN.

After each spectral peak of the etalon passband or FBG, reference or sensor FBG is assigned a sub-sample timestamp per the algorithm's first part, as discussed above. The second part of the algorithm aims to predict the sensor FBG Bragg wavelength value from the sub-sample timestamps of the spectral peaks as computed above. The algorithm to achieve this has been discussed in [2] and [7]. Whenever the tunable LASER sweeps across its range once, two sets of sub-sample timestamp values are obtained, one for the channel containing the athermal etalon and the other containing the reference and sensor FBGs

The algorithm takes in the timestamp values of both channels. Using these timestamp values, this algorithm calculates the Bragg wavelength of the FBG sensors. The nominal values of each FBG sensor, reference FBG, and the wavelengths of the ITU-T grid (i.e., etalon passband central wavelength) are known from before (stored in tables). As discussed earlier, the sub-sample timestamp values correspond to the central wavelength of the spectral peaks in the time

domain plot for both FBGs and the Etalon pass band. As the Tunable Filter is not linear, the spacing between wavelengths and intervals between their corresponding timestamp values are not linearly related. But, with the train of spectral peaks that we get from the Etalon channel, the non-linear range of wavelength of the Tunable Filter can be divided into small segments, which may be considered piecewise linear. Thus, the problem of the non-linearity of the Tunable filter is addressed. However, there is another issue of non-repeatability of Tunable LASER wavelength vs voltage applied in a tunable filter. Suppose the same voltage is applied to the tunable filter at different times. In that case, the LASER wavelength may be tuned to other values, so when the same triangular voltage wave is applied to the tunable filter by its controller, the minimum wavelength, maximum wavelength, and the range of wavelength sweep of the Tunable LASER changes considerably in each wavelength sweep. So, the number of peaks in the etalon channel differs in each scan. So, although we get a train of peaks (or array of sub-sample timestamps in our case) for the etalon channel, we do not know which peak (or sub-sample timestamp) corresponds to which wavelength of the ITU-T grid. In such a situation, the reference FBG (the only FBG maintained in nominal conditions and whose Bragg wavelength is known) helps to determine which etalon peak corresponds to which wavelength of the ITU-T grid. The algorithm for finding the wavelength shifts of FBG sensors is illustrated below.

A sweeping voltage from 20V to 0V is applied to the FFP-TF2 Tunable Filter, resulting in a wavelength sweep of the Tunable LASER from around 1530nm (appx) to around 1570nm(appx). As the tunable filter is not repeatable, the lower and upper wavelength limits vary in each sweep.
Now, this sweeping wavelength is applied to FBGs (one of which is reference FBG, and the rest are sensor FBGs) and Etalon separately with the help of an optical splitter. The reflected spectrum of the FBGs and transmission spectrum of the etalon is captured in two photodetectors (PD1 and PD0). PD0 outputs a train of voltage peaks in each sweep, one for each channel of the etalon. But now, we do not know which channel of etalon corresponds to what wavelength. PD1 outputs as many voltage peaks as many FBGs are connected. In our setup, the reference FBG has a nominal wavelength of 1540.7nm, and the nominal wavelengths of FBG sensors are 1545nm to 1569nm. So, the first peak of PD1 corresponds to the reference grating.

The ADC captures these voltage peaks from PD0 and PD1, and the first part of the algorithm (discussed before) computes a timestamp for each peak, referred to as its sub-sample timestamp. It may be noted here that, as both channels are acquired simultaneously, a lower timestamp corresponds to a lower wavelength across both channels. The timestamps of both channels are recorded for further processing.

The algorithm first finds out timestamps in the Etalon channel that are just before and just after the timestamps of the reference FBG. As the wavelength of the reference channel is already known, the timestamp in the etalon channel just preceding the timestamp of reference FBG corresponds to the wavelength in the ITU-T grid that comes just before the wavelength of reference FBG. Similarly, the timestamp in the etalon channel coming just after the timestamp of reference FBG corresponds to the wavelength in the ITU-T grid that comes just after the wavelength of reference FBG. For example, if the nominal wavelength of our setup is 1540.7nm and its timestamp is T_r. Now, with this timestamp T_r, we will search for the two timestamps in the etalon channel that come just before and after T_r. In other words, we search for two consecutive timestamps in the Etalon channel such that the timestamp of the reference grating T_r lies between them. Say, these two timestamps are T_{-1} and T_{+1}. So, referring to the ITU-T grid, we can say that timestamp T_{-1}corresponds to the wavelength 1540.56nm and

timestamp T_{+1} corresponds to the wavelength 1541.35nm because 1540.56nm and 1541.35nm are two consecutive wavelengths in ITU-T grid between which our reference grating wavelength of 1540.7 sits.

Now, if in a train of timestamps for etalon channel wavelengths corresponding to any one can be determined, then just the wavelengths corresponding to all others can be resolved by consulting the ITU-T grid chart. Here, using the reference FBG, we found two of such etalon timestamps; the algorithm determines the wavelengths correspond to each etalon timestamp.

Now, after all the sub-sample timestamps in the etalon channel have been assigned a corresponding wavelength from the ITU-T grid, then just by linear interpolation method, the corresponding wavelengths of FBG sensors can be determined from their timestamp. Although the timestamp vs. wavelength relation is non-linear, as the etalon wavelengths are closely spaced, we can approximate this non-linear relation as a piecewise linear relation and apply linear interpolation to find the FBG sensor wavelength from its timestamp.

2.3 Problem Statements

In light of the existing work in this domain, as discussed in the literature review, in this thesis, novel ANN-based algorithms to determine the central wavelength of a spectral peak from the sampled values obtained by sparse sampling have been proposed for FBG interrogation. The proposed ANN-based algorithms are an improvement over the other existing algorithms (ANN-based or otherwise) as they take care of the non-ideal behaviors of the FBG sensor signals often encountered in their field applications. The training steps of the proposed ANN-based algorithms have an advantage over the existing ANN-based algorithms in that they require less training data, and the developed models are more robust for the applications where FBGs need a pre-strain or any other pre-conditions where nominal Bragg wavelength of the FBG sensors get arbitrarily shifted from their nominal value. The prediction steps of the proposed algorithms are suitable for field applications with stringent requirements for time-bound performance.

As there are two different types of popular FBG interrogation techniques, two different albeit related ANN-based FBG Bragg wavelength detection algorithms must be proposed, one for CCD spectrometer-based interrogation and the other for tunable filter-based interrogation.

For CCD spectrometer-based interrogation, an ANN-based algorithm needs to be proposed such that the developed model can accurately predict the Bragg wavelength of the FBG sensor in stringent real-time applications and should work even when the optical signal intensity from the FBG sensor fluctuates. Once developed after training, the model should be valid and valuable for prediction even if the Bragg wavelength of the FBG sensor shifts significantly from its nominal position, i.e., the training algorithm and amount of training data should be independent of the Bragg wavelength range of the FBG sensor.

Similarly, for tunable filter-based interrogation, an ANN-based algorithm needs to be proposed such that the developed model can accurately predict the Bragg wavelength of the FBG sensor in stringent real-time applications and should work even when the optical signal intensity from the FBG sensor fluctuates. Once developed after training, the model should be valid and valuable for prediction even if the Bragg wavelength of the FBG sensor shifts significantly from its nominal position, i.e., the training algorithm and the amount of training data should be independent of the Bragg wavelength range of the FBG sensor. Additionally, as the tunable filter's 'tuning voltage vs passband wavelength' relation is neither linear nor even repeatable, the algorithm should withstand such non-ideal properties provided they are within some limit.

Chapter-3: CCD spectrometer-based FBG Interrogation using ANN algorithm

This chapter proposes a novel ANN-based algorithm for a CCD spectrometer-based interrogation. A model using the proposed algorithm has been developed. This developed model takes care of fluctuations in the intensity of the optical signal applied to the CCD spectrometer array from the FBG sensor. Further, once developed by the proposed algorithm, the model remains usable and valid for predicting the central wavelength of the FBG spectral peak even if it shifts significantly from its nominal position. Moreover, the proposed model is an ANN, which can also be implemented on an FPGA[3], making it very suitable for real-time application.

In the subsequent sections of this chapter, we first discuss the set-up and methodology of FBG interrogation using a CCD spectrometer, an algorithm for training the model, followed by a prediction algorithm using the model. Finally, we compare the performance of our proposed model with the existing algorithms, such as curve-fitting and centroid-finding algorithms.

3.1 Set-up & Methodology of FBG Interrogation

In CCD array-based interrogation, an optical circulator directs a broadband light onto an FBG sensor. The reflected light from the FBG sensor is then directed onto a one-dimensional array of CCD pixels through a diffraction grating. The diffraction grating spatially distributes the incoming reflected light from the FBG sensor such that, based on the wavelength spectrum of this light (dependent on the FBG sensor spectrum), it is determined which pixels are illuminated and to what extent. The intensity of light incident on a given pixel corresponds to a charge and resultant voltage build-up in the pixels. Thus, when pixel voltages are read out, we get higher voltages for pixels on which more optical intensity is incident and lower voltages for pixels subjected to lower optical signal intensity.

The pixel voltages serve as samples of the FBG sensor spectrum profile. Each pixel's voltage value is digitized and tabulated along with the pixel number. Subsequently, the exact FBG sensor spectrum central wavelength is determined from these pixel-voltage values. This process is akin to predicting the FBG spectrum from a few spatial samples and then determining its central wavelength. All the algorithms discussed in this paper for this interrogator, including curve-fitting, centroid, and the proposed neural network-based method, aim to predict the central wavelength from the voltage distribution in the pixels.

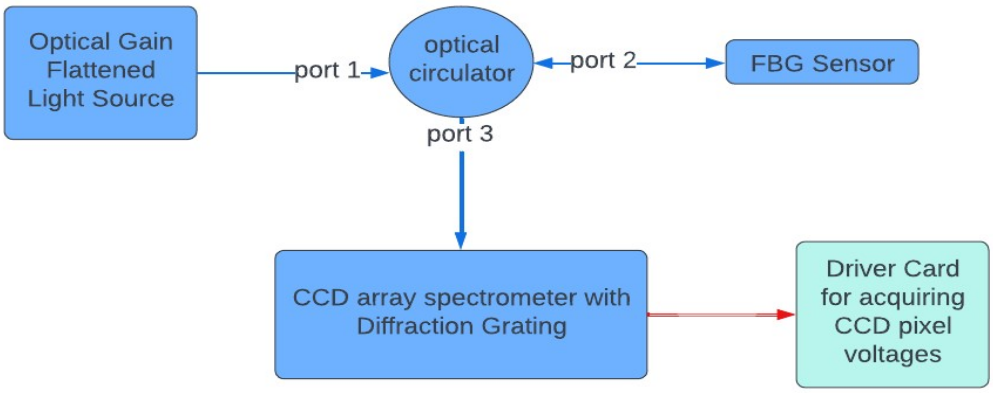


Fig. 3.1. Block diagram of CCD-based interrogation

3.2 ANN Training Algorithm for CCD array-based Interrogator

This paper proposes a two-layer neural network-based algorithm for predicting the FBG sensor Bragg wavelength in a CCD array-based interrogation system. This prediction is based on the voltage values of the CCD pixels when the reflected optical signal from an FBG sensor is incident on the CCD array. The number of nodes in the input layer depends on parameters like the 'wavelength per pixel' of the CCD array, the bandwidth of the FBG sensor spectrum, and the required interrogation resolution. In our case for illustration, we consider a 512-pixel CCD interrogator with a 'wavelength per pixel' of 156 pm having a wavelength range spanning over 79872 pm from 1510.064nm to 1589.936nm. The FBG spectrum simulated in our case has a symmetric normalized profile 1500 pm wide with FWHM (Full Width Half Maximum) of 250 pm. Our target resolution is 1 pm.

This problem of predicting the FBG sensor central wavelength (or Bragg wavelength) from the pixel voltages of a CCD array-based sensor is framed as a classification problem. Given that the spectrum of the FBG sensor is 1500pm wide and the 'wavelength per pixel' is 156 pm, a maximum of ceil(1500/156) = 10 pixels will be required in our input layer. However, to maintain an odd number of nodes for symmetry reasons and allow for flexibility with broader spectrums of FBG sensors in the input layer, we opt for 15 nodes. Since ‘wavelength per pixel’ is 156 pm and the required resolution of the interrogation is 1 pm, the number of nodes in the output layer is 156.

Now, when there are adjacent pixels in the CCD array with voltages surpassing a certain threshold, that implies some FBG optical signal in that corresponding region of the spectrum. Each pixel in the CCD array corresponds to a specific wavelength in the optical spectrum, information typically provided by the device manufacturer. However, with 79872 wavelengths in increments of 1 pm and only 512 pixels, not all wavelengths have a corresponding assigned pixel number. Using the voltage values of the adjacent pixels, the ANN will discern patterns and determine the central wavelength of the FBG sensor. Relative to the pixel with the highest voltage value, the ANN outputs an integer value ranging from -77 to +78.

If the pixel number with the maximum voltage is P_{max}(which lies between 0 and 511), its corresponding wavelength is W_{max}(which lies between 1510064pm and 1589936 pm), and the output of the ANN is X(which lies between -77 and +78), then the predicted Central Wavelength of the FBG sensor is W_{max}+ X pm.

In the proposed training method, a set of training data is generated for each FBG sensor. At first, the profile (or shape) of the FBG is plotted densely (i.e., one sample at each pm). This can be done with a high-resolution Optical Spectrum Analyzer. Then, this densely plotted profile shape can be sampled arbitrarily at all discrete points, maintaining the constraint that any two consecutive discrete samples are fixed wavelengths apart, the same as the difference in corresponding wavelength between two successive pixels. From the dense plot, we know where precisely the central wavelength lies. So, the wavelength difference between the 'corresponding wavelength of sampled point with maximum value' and the 'central wavelength of the FBG spectrum' is also known.

This is illustrated with an example below. Consider the blue line in Fig:3.2 below, which is the normalized (i.e., max value is 1) spectrum of a given FBG sensor. The part of the FBG sensor spectrum shown in the figure is 1500 pm wide. This spectrum is incident on a 512-pixel CCD array with a wavelength range of ~80nm. So, the wavelength difference between two consecutive pixels is ~80nm/512 = 156pm. The red dot marks are the discrete points where this spectrum is sampled. In this case, the discrete samples are 156pm apart, as it is the wavelength difference between the corresponding wavelengths of two consecutive pixels. This FBG spectrum is sampled arbitrarily at all possible points with the only constraint that the two successive discrete sample points are permanently fixed distance apart (156 pm in our case). As the consecutive samples are always 156 pm apart, there will be only 156 distinct sample sets, after which, due to the periodicity of sampling points, we get a repetition of the same sample sets.

In Fig 3.2, we can see that the highest-valued sample is at x=691. (x is the x-axis of the plot representing the relative wavelength of the spectrum in pm.) As the central wavelength of the FBG is at x=750, it is +59 pm higher than the highest-valued sample.

Therefore, our training data includes all possible sample sets, in which each pattern set (i.e., a set of discrete samples from the dense plot of the spectrum) is accompanied by its respective target value. The target value is determined as the FBG sensor spectrum's central wavelength minus the corresponding wavelength value of the pixel with the highest sampled value in the pattern set. To create the training data, the sample amplitudes within each pattern set are normalized relative to the maximum amplitude in the set (pattern normalization). Additionally, the amplitudes are centered so that the maximum amplitude is shifted to the middle position of the pattern-set (pattern-centering). By pattern normalization, we mean that the value of the highest-valued sample in the pattern set is scaled to 1. Other sample values in the pattern set are scaled accordingly so that their relative magnitudes among themselves and concerning the highest value sample are not altered.

Similarly, by pattern-centering, we mean that the samples in the pattern set are arranged so that the sample with the highest value is the middle sample of the set. So, after pattern-normalization and pattern-centering of the pattern set, the central value of the set (which is at index seven as the set contains 15 elements) contains value 1(the highest value in the set). This central value is replaced by the [centroid value of the pattern set -6] and fed into ANN as input nodes. The target output value computation has already been discussed earlier. So, the steps of the training algorithm are illustrated as follows.

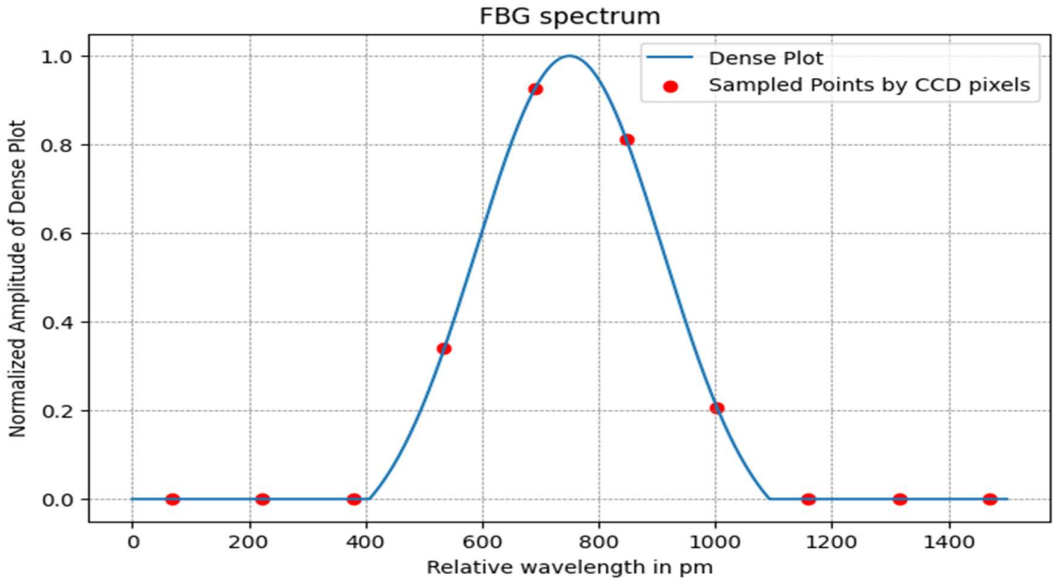


Fig. 3.2 shows the FBG sensor spectrum (blue line) sampled by CCD pixels at discrete points (red dots)

Steps of Training Algorithm. Using a training algorithm, training data is formed. The training data is a matrix in which each row is a training sample. Each training sample has two components: the pattern set and a target value. All elements, excluding the last one, compose the pattern set in each row, while the previous element represents the target value.
Consequently, the training matrix's last column represents the target set, containing the training samples' target values. In the following steps the first five steps describe how the pattern set of the training data is formed. The 6th step describes how the corresponding target value for each pattern set is computed. The training data for a given FBG sensor spectrum is done in the following steps:

1. First, the normalized profile of the FBG sensor spectrum is plotted. This is done in two steps. First, the reflection intensity value of the sensor is densely tabulated in a row for every wavelength in steps of 1 pm. Then, this FBG spectrum is normalized by dividing all the values in the row by the maximum value. So, after normalization, the peak value of the FBG spectrum is 1. This spectrum is plotted with the x-axis representing the wavelength axis of the spectrum. This wavelength, denoted by the x-axis, is not the actual wavelength of the spectrum but a relative wavelength with respect to the lowest wavelength in the FBG sensor spectrum. The unit of the x-axis is pico-meter (pm). In the above-illustrated example (Fig 3.2 blue line), in a single row, we have 1500 values as the FBG sensor spectrum is plotted for 1500pm, and the spectrum is symmetric and peaks at x=750nm.

2. After the FBG spectrum profile is plotted, next in our illustrated case, it is sampled at all possible points so that consecutive samples are 156 pm apart. In the illustrated case, 156 sample sets (called pattern sets) are possible if we sample at discrete pm. These sets are amplitudes at x=[0, 156, 312, 468, 624, 780, 936,1092,1248, 1404], amplitudes at x=[1, 157, 313, 469, 625, 781, 937,1093,1249, 1405],amplitudes at x=[2, 158, 314, 470, 626, 782, 938,1094,1250, 1406] and so on till amplitudes at =[155, 311, 467, 623, 779, 935,1091,1247, 1403]. After that, we again get the same sets over and over again.

3. All the amplitude values in each such pattern set are then divided by the maximum value of the sample amplitude in the set. Therefore, the maximum amplitude value in the pattern set becomes 1, and the rest of the amplitude values in that set become ≤ 1. This step is named 'pattern-normalization' in this document.

4. Now, we center the sample points so that the highest sample is in the middle of the row. In our case, we will take a 15-point row. So, the highest amplitude value (1 after pattern normalization) is kept at index 7 (indexing starts at 0). All other sample values are mapped accordingly. Where there is no value to map, it is padded with zero. This step is named 'pattern centering' in this document.

Table 3.1. Table showing Pattern-Normalized and Pattern-Centered Pixel Amplitude of FBG sensor spectrum in Fig 3.2

Pixel No. Relative to the peak	Normalized and Centered Pixel Amplitude
0	0
1	0
2	0
3	0
4	0
5	0
6	0.366
7	1
8	0.878
9	0.223
10	0
11	0
12	0
13	0
14	0

5. Then, the centroid of this normalized and centered set is computed by the formula:

$$centroid = \frac{\sum_{n} \{n * max(0, (A_n - A_{mp}))\}}{\sum_{n} max(0, (A_n - A_{mp}))} \quad , n = 0, 1, \ldots 14$$

Where n is the index number, A_n is the normalized and centered pattern set Amplitude, and A_{mp} is the minimum positive amplitude in the pattern set.

In our example :

centroid = {0x0+1x0+2x0+3x0+4x0+5x0+6x(0.366-0.223) +7x(1-0.223) + 8x(0.878- 0.223) + 9x(0.223 - 0.223)+10x0+11x0+12x0+13x0+14x0} / {0+0+0+0+0+0 + (0.366-0.223) + (1-0.223) + (0.878- 0.223) + (0.223 - 0.223) +0+0+0+0+0}= 11.537/1.575 = 7.325

Now, finally, '1' at index seven is replaced by centroid – 6 = 7.325-6 = 1.325

Therefore, now the final pattern set is given in Table 3.2:

Table 3.2. Table showing Normalized and Centred Pixel Amplitude with centroid

Pixel No. Relative to the peak	Normalized and Centered Pixel Amplitude
0	0
1	0
2	0
3	0
4	0
5	0
6	0.366
7	1.325
8	0.878
9	0.223
10	0
11	0
12	0
13	0
14	0

6. The target value for a particular sample set is derived by subtracting the corresponding wavelength for the pixel with the highest amplitude from the central wavelength of the FBG spectrum. For the case illustrated in Fig 3.2, the sample (pixel) with the highest amplitude has its corresponding wavelength at 691pm, and the FBG spectrum has a central wavelength at 750pm. The target for this set of samples is 750-691 = +59. This means that the central wavelength of the FBG sensor spectrum is 59 pm, more than the wavelength corresponding to the 7th index of the table.

Another training sample is illustrated in the following figure (Fig 3.3). The sample (pixel) with the highest amplitude has its corresponding wavelength at 811pm, and the FBG spectrum has a central wavelength of 750pm. The target for this set of samples is 750-811 = -61. This means that the central wavelength of the FBG sensor spectrum is 61 pm less than the wavelength corresponding to the seventh index of the table.

While generating the network's training data, the lower amplitudes were systematically dropped in some pattern sets to make the model more sensitive to higher amplitudes. Lower amplitudes are often noise-prone; hence, they are discarded unless they are above a certain threshold. However, dropping the lower amplitudes ensures that at least two of the highest amplitudes are present in the pattern set.

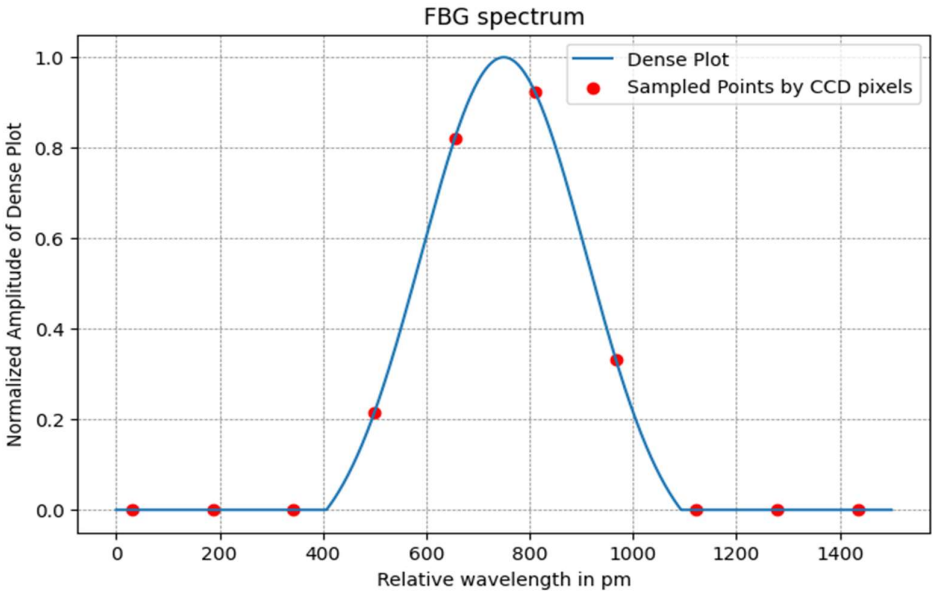


Fig. 3.3. FBG sensor spectrum (blue line) sampled by CCD pixels at discrete points (red dots)

She proposed an Artificial Neural Network. The ANN has two layers for this purpose. The input layer comprises 15 nodes corresponding to the 15 elements of normalized, centered, and padded rows, with the central element (index 7) replaced with a centroid value. The input layer is densely connected to the nodes of the output layer. The output layer has 156 nodes, each corresponding to a 1 pm shift left or right of the pixel with maximum amplitude. Adam optimizer was used with softmax activation function to train this neural network, and the learning rate was 0.1.

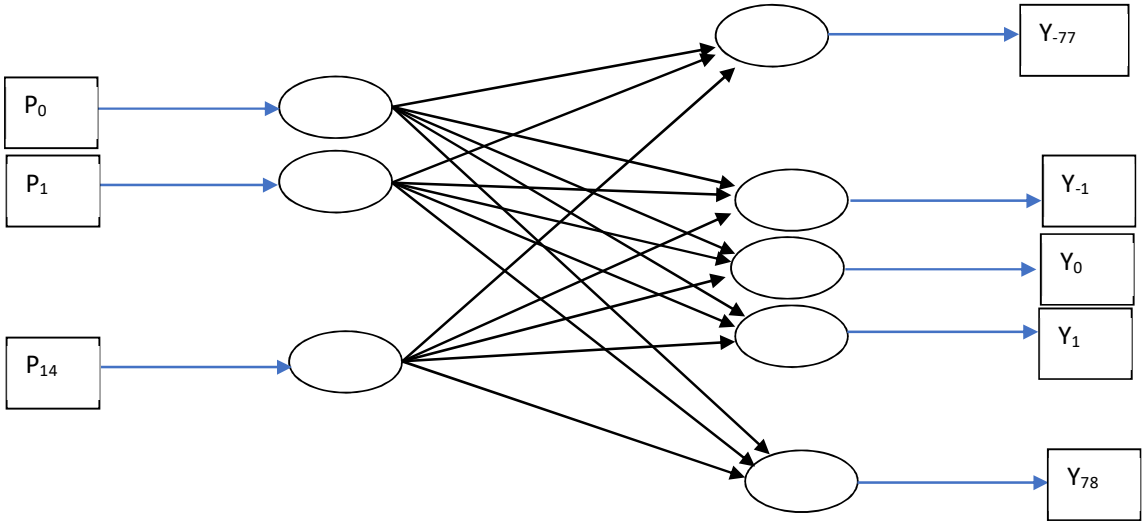


Fig. 3.4. Architecture of ANN formed.

The algorithm to generate the training data and save it in a file is described in the following 'ALGORITHM:3.1', and the algorithm to build a model from this training data is described in 'ALGORITHM:3.2' respectively.

Algorithm 3.1: Pseudocode to generate and save the training data for CCD spectrometer based Interrogation:

```
# Function for Gaussian curve
FUNCTION gaussian_waveform(x, amplitude, mean, stddev):
   RETURN amplitude * EXP(- (x - mean)^2 / (2 * stddev^2))

# Parameters
num_points = 1501 # Number of points (pico-meter) in the FBG spectrum
amplitude = 1 # maximum value of the peak, i.e., normalized plot
mean = 750  # This peak has a centre at position  index 750 pm (relative)
x_values = ARRAY of length num_points initialized with values from 0 to num_points-1 # making the x-axis of
gaussian waveform simulating dense plot of FBG spectral peak

# Generate the distorted Gaussian waveform
gaussian_wave = CALCULATE [(gaussian_waveform(x_values, amplitude, mean, stddev=160) – 0.1)/0.9]
#making the distorted gaussian wave which is simulating dense plot of FBG spectral peak
threshold = 0.0 # minimum value of the waveform is zero, i.e. no negative amplitude
# resetting the negative values to zero
FOR each element in gaussian_wave:
      IF element < threshold:
         set element to 0

# Here we sample the Gaussian waveform at intervals and make a training data matrix
num_samples = 156 # sampling interval is 156 pm
lst_sample_pts = Empty  List of arrays  # list of arrays (rows) of sampled wavelengths
lst_sample_vals = Empty  List of arrays # list of arrays (rows) of sampled values
FOR starting_pt IN RANGE(num_samples): # the start position is varied to generate all possibilities
    sampling_points = array of integers in the range from zero upto num_points sampled at values starting from
                                        starting_pt with step size num_samples
   APPEND sampling_points to lst_sample_pts
   APPEND 'gaussian_wave sampled at indices which are elements of array sampling_points' to lst_sample_vals


# Here, the sampled values are normalized by dividing each row by its max element
FOR i IN RANGE(number of rows in lst_sample_vals):
   max_value = MAXIMUM value in lst_sample_vals[i] # Find the maximum element in the current row
   # Divide each element of the row by the maximum element of the row
   each element of lst_sample_vals[i] = (each element of lst_sample_vals[i]) / max_value


# max_indices contains the indices of maximum sampled values(i.e. 1) of each row before centering
max_indices = ARRAY of indices containing the index of the maximum element in each row of lst_sample_vals


# In this step, the sampling wavelength and sampled value arrays are padded with a calculated number of
zeros
   such that the number of elements in each array is 15. This step is required before centering arrays
FOR each row in lst_sample_pts:
      padded_lst_sample_pts  = row padded with [15- len(row)] number of zeros to make its length 15
      APPEND row to training_matrix1
     # training_matrix1 contains the suitably padded array of sampling wavelengths
```

```
FOR each row in lst_sample_vals:
      padded_lst_sample_vals  = row padded with [15- len(row)] number of zeros to make its length 15
      APPEND row to training_matrix1g
     # training_matrix1g contains the suitably padded array of sampled values



# Next, the sampled values and sampling points arrays are centered adequately so that the element at index position 7 has the highest sampled value and its corresponding sampling wavelength, respectively.
# the centered sample value array (centered_array_g) will be used to make a pattern set (i.e., X of training data)
# centered sampling wavelength (centered_array) will be used to compute the 'wavelength difference between FBG central wavelength and corresponding wavelength of max sampled value' (target of training data)

centered_array = 2D ARRAY of zeros with dimensions same as training_matrix1 # initialized with zero
FOR i IN RANGE(length of training_matrix1):
   centered_array[i][(7 - max_indices[i]) to end of array] = training_matrix1[i][ 0 to (8+max_indices[i])]

centered_array_g = 2D ARRAY of zeros with dimensions same as training_matrix1g # initialized with zero
FOR i IN RANGE(length of training_matrix1g):
   centered_array_g[i][(7 - max_indices[i]) to end of array] = training_matrix1g[i][0 to (8+max_indices[i])]

# the corresponding wavelength of the maximum sampled value is stored in max_sampled_val_wl
# this wavelength can be found as it will be in index 7 of each array of centered_array due to centering
max_sampled_val_wl = ARRAY containing wavelengths at index 7 of each row of centered_array matrix

# shift_wl is the wavelength difference between FBG central wavelength and corresponding wavelength of max sampled value
# this will be used to target training data (i.e., y of training data).
shift_wl = 750 - max_sampled_val_wl

# Here, the X and y of the  training data are joined (normalized centered pattern data is joined with target value)
result_2d_array = CONCATENATE centered_array_g with shift_wl


# Next, from the training matrix result_2d_array, systematically lower amplitudes in the pattern set are dropped, and the highest amplitude (which is always one due to normalization) is replaced with the centroid value of the retained amplitudes to make a new training matrix np_matrix10. While dropping the lower amplitudes, it is ensured that at least two amplitudes in the pattern-set of each training sample, i.e., the highest two amplitudes in each pattern set, are always retained. While computing the centroid, instead of considering the normalized amplitude directly, we believe the differential value of normalized amplitude with respect to the smallest retained normalized amplitude in the pattern set.
np_matrix = result_2d_array
np_matrix10 = EMPTY 2D ARRAY with zero rows and 16 columns
# np_matrix10 will be  used to store the complete training data, including patterns in which lower amplitudes have been discarded
max_iterations = 15
# since there are 15 elements in each row of the pattern set, no more than 15 iterations are required to form a pattern set by discarding lower-valued samples
np_matrix_w1 = COPY of np_matrix # making a working copy of the training set

FOR iteration_count IN RANGE(1, max_iterations + 1): #Iterate for generating training data
{
```

```
   change_flag = 0 # flag is set if there is any change with respect to the previous iteration
   min_row_pos_num = Empty  List # minimum positive element in each row will be stored in this list
   gt_min_row_num = Empty  List # count of amplitudes greater than minimum positive element in each row
                                  will be stored in this list
   min_row_pos_index = Empty  List # indices of the minimum positive element in each row will be stored here

   # in the following section, we check some status of np_matrix_w1 matrix related to its minimum positive
     element, and number of values above the minimum positive element in each row
   # while performing row-wise iterations, the last column containing target values of training data is dropped
   FOR row IN np_matrix_w1 with target column dropped:
      positive_numbers = POSITIVE numbers in row
      IF length of positive_numbers > 0:
         min_positive = MINIMUM value in positive_numbers in row
         gt_min_pos = COUNT of numbers in positive_numbers GREATER THAN min_positive in row
         min_index = INDICES of row where min_positive is present
         APPEND min_positive to min_row_pos_num
         APPEND gt_min_pos to gt_min_row_num
         APPEND min_index to min_row_pos_index
      ELSE:
         APPEND 0 to min_row_pos_num
         APPEND 0 to gt_min_row_num
         APPEND -1 to min_row_pos_index

# next np_matrix_w2 and np_matrix_w3  matrix is created from np_matrix_w1.
   np_matrix_w2 = COPY of np_matrix_w1 # np_matrix_w2 will be used for computing centroid
   np_matrix_w3 = COPY of np_matrix_w1 # np_matrix_w3 is for appending to training data

   FOR i IN RANGE(number of rows of np_matrix_w1):
      IF gt_min_row_num[i] >= 2: #if there are at least two numbers in a row greater than the current min
positive value
         first 15 elements of np_matrix_w2[i] = first 15 elements of np_matrix_w1[i] - min_row_pos_num[i]
         np_matrix_w1[i][min_row_pos_index[i]] = 0 # make the minimum positive value of the row as 0
         set the negative values in the first 15 elements of np_matrix_w2[i] = 0
         change_flag = 1 #if there is any new training data in this iteration, this flag gets set

   # Next, np_matrix_w3 matrix is created for appending to training data
   sum_of_product_numerator = SUM (elements in first 15 columns of np_matrix_w2 * array of 0 to 14)
   sum_denominator = SUM of elements in first 15 columns of np_matrix_w2
   centroid = sum_of_product_numerator / sum_denominator #elementwise array division
   np_matrix_w3[ 7] = (centroid - 6) # index 7 of all rows of np_matrix_w3 replaced in elementwise array
                                      substitution
   np_matrix10 = CONCATENATE np_matrix10 with np_matrix_w3

   # if there is no change in the iteration, then break the iteration
   IF change_flag == 0: # flag=0 means NO change in this iteration
      BREAK
}

SAVE np_matrix10 to file_path as CSV
```

Algorithm 3.2: Pseudocode to build model from the training data for CCD spectrometer based Interrogation:

```
np_matrix10 = READ_CSV(training data)

np_matrix11 = EMPTY 2D ARRAY
FOR i FROM 0 TO 10:
   np_matrix11 = CONCATENATE np_matrix11 with np_matrix10

np_y11 = LAST COLUMN of np_matrix11
np_X11 = ALL COLUMNS of np_matrix11 EXCEPT the last one

X_train, X_test, y_train, y_test = SPLIT_TRAIN_TEST(np_X11, np_y11)

y_traink = y_train + 77
y_testk = y_test + 77

model = keras.Sequential([ keras.layers.Dense(156, input_shape=(15,), activation='softmax')])
custom_learning_rate = 0.1
custom_optimizer = tf.keras.optimizers.Adam(learning_rate=custom_learning_rate)
model.compile(optimizer=custom_optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_traink, epochs=20000)
SAVE_MODEL()
```

3.3 Bragg Wavelength Prediction using the proposed trained ANN for CCD array-based FBG Interrogator

After training the ANN model, it can be employed to predict the central wavelength of an FBG sensor spectrum in the following manner. When an optical signal from an FBG is incident on a CCD array, a specific set of pixels is illuminated, resulting in voltage buildup on this set. The voltage values of all pixels in this set are normalized by dividing them by the maximum voltage in the set, making the maximum normalized pixel value 1, which is assigned to the pixel with the highest registered voltage.
The wavelength corresponding to this particular pixel is denoted as W_{max} in picometres. The normalized amplitude values are then centered. In this process, the sample with a value of 1 (i.e., the max value) is mapped to index 7 of another array capable of holding 15 elements (index 0 to index 14). This array maps Other sample values accordingly to keep their relative positions unchanged. Empty positions in this array, where no corresponding element from the FBG spectrum, are filled with zero values.

Next, the centroid of this set is computed, and six is subtracted from this centroid value. Then, this value is placed in index 7 of the array, replacing the previous value of 1. These elements of this new array are input into the trained model. The trained model activates one of the 156 output nodes. Depending on which output node is activated, a specific wavelength (denoted as X in pico-meters) is added or subtracted from W_{max} to determine the central wavelength of the FBG spectrum.

Algorithm for determining the central wavelength from the CCD pixel values: The algorithm for predicting the central wavelength of an FBG spectrum is illustrated below with an example.

Say we obtain the following voltages in the CCD pixels. The pixel numbers and respective voltages are shown in the diagram and the table.

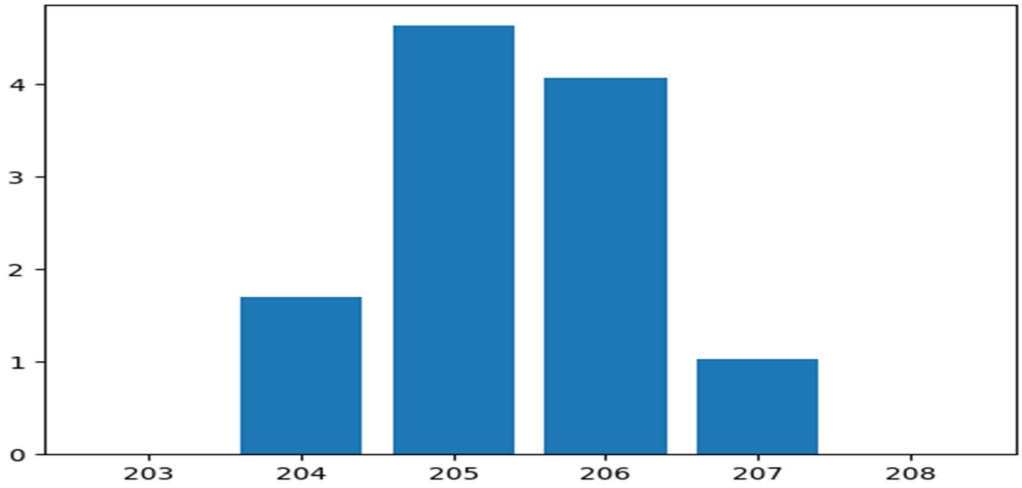


Fig. 3.5. Voltages on CCD pixel array due to optical signal from FBG

Table 3.3. Table showing voltage values on CCD pixels

Pixel No. of CCD array	Voltage Recorded on pixel
204	1.697
205	4.635
206	4.068
207	1.036

These pixel amplitudes are first stored in a 15-element row as follows. Appropriate zero-padding is done for pixels whose amplitude is either zero or less than a threshold.

Table 3.4. Table showing Pixel Amplitude Value

Index No.	Pixel Amplitude Value
0	1.697
1	4.635
2	4.068
3	1.036
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0

13	0
14	0

Then, the values are normalized by dividing all the elements by the maximum element value.

Table 3.5. Table showing Normalized Value

Index No.	Normalized Value
0	0.366
1	1
2	0.878
3	0.223
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0

Then, the values are centered, so the max value 1 is at index seven.

Table 3.6. Table showing Centered Normalized Value

Index No.	Centered Normalized Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0.366
7	1
8	0.878
9	0.223
10	0
11	0
12	0
13	0
14	0

Then, the centroid of this normalized and centered set is computed by the centroid formula:

$$centroid = \frac{\sum_n \{n*max(0,(A_n - A_{mp}))\}}{\sum_n max(0,(A_n - A_{mp}))} \quad , n = 0,1,...14$$

In our example :

centroid = {0x0+1x0+2x0+3x0+4x0+5x0+6x(0.366-0.223) + 7x(1-0.223) + 8x(0.878- 0.223) + 9x(0.223 - 0.223)+10x0+11x0 + 12x0 + 13x0 + 14x0} / {0+0+0+0+0+0 + (0.366-0.223) + (1-0.223) + (0.878- 0.223) + (0.223 - 0.223) +0+0+0+0+0}= 11.537/1.575 = 7.325

Now, finally, the one at index seven is replaced by centroid – 6 = 7.325 -6 = 1.325

So, now the final training set is as follows:

Table 3.7. Final Training set

Index No.	Training Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0.366
7	1.325
8	0.878
9	0.223
10	0
11	0
12	0
13	0
14	0

Now, these values are fed into our trained ANN model, and we get the output as +59. From the calibration data of the CCD array sensor, we can see that the 205th pixel (i.e., the pixel with the highest voltage) corresponds to wavelength 1542.122 nm. So, the central wavelength of the CCD sensor array is 1542.122nm + 59 pm = 1542.181 nm. We have used the classification instead of regression in the proposed model. Therefore, instead of a single output, our model has 156 output nodes each in the 1 pm step. Only one of the output nodes is activated, and the wavelength corresponding to that activated node is considered the output of ANN. This output wavelength is used to compute the central wavelength of the FBG sensor.

The prediction algorithm is described in the pseudocode described in 'Algorithm:3' in which the central wavelength of an FBG peak is predicted using the sample voltages listed in 'lst_sample_vals.'

Algorithm 3.3: Pseudocode to predict Central wavelength of FBG peak from the pixel data for CCD spectrometer based Interrogation:

```
# Normalizing sampled values
max_value = MAXIMUM value in lst_sample_vals
lst_sample_vals = lst_sample_vals / max_value
max_indices = Index of maximum sampled values in lst_sample_vals
max_pixel = Pixel number of maximum sampled values in lst_sample_vals
max_pixel_wl = wavelength corresponding to max_pixel

# Padding sample points and values arrays
padded_lst_sample_vals = [PAD lst_sample_vals with zeros to length 15]

# Centering sample points and values arrays

centered_array_g =  ARRAY of 15 zeros
centered_array_g[(7 - max_indices) to last_index_of_array] = padded_lst_sample_vals [0  to (8 + max_indices)]

centroid_centered_g = COPY of centered_array_g
row = COPY of centered_array_g
smallest_positive = MINIMUM positive value in centroid_centered_g
   IF COUNT of row elements greater than smallest_positive >= 2:
      centered_array_g [condition: element > 0] =  (element-smallest_positive)
   sum_of_product_numerator = SUM(centered_array_g w * array of 0 to 14) #element wise product
   sum_denominator = SUM centered_array_g)
   centroid = sum_of_product_numerator / sum_denominator
   centroid_centered_g[7] = centroid - 6

model = LOAD_MODEL()

output = PREDICT_MODEL(model, centroid_centered_g)
central_wavelength_of_FBG_peak = max_pixel_wl + output -77
```

3.4 Implementation and Performance of the model

After the training algorithm generates the training samples, the ANN model is built using Tensorflow and the Keras package. It is a two-layer, densely connected model. The computation was carried out on the Google Colab platform. In the ANN model, the Softmax activation function is used. The Adam optimizer and the 'Sparse Categorical Crossentropy' loss function are used to train the model.
While training the model, the low-valued samples were systematically truncated so that higher-valued samples primarily govern the output of the ANN. This will allow us to set a threshold level to truncate low-valued samples that are more affected by noise. The data is trained for 1,30,000 epochs.

The training samples are divided into training and validation sets. The training set is used to train the ANN model. Then, using the trained ANN model, the Bragg wavelength of the FBGs is predicted for validation set samples. The Error in the expected value of FBG wavelength is computed, and the following observations are made.

- The error data showed that the correct wavelength is predicted in 56.56% of the cases.
- The Error is within ±1 pm in 95.84% of predictions.
- Error is limited within ±2 pm in 99.51% of predictions
- Error is limited within ±3 pm in 99.83% of predictions
- The maximum range of predicted Error is ±4 pm, which happens in less than 0.02% of predictions
- The mean absolute error observed is less than 1 pm.

3.5 Comparison with other FBG central wavelength detection algorithms

As discussed earlier, the two most common traditional algorithms for central wavelength detection of an FBG sensor are curve fitting algorithm-based detection and centroid finding algorithm-based detection. Both have advantages and limitations, which are discussed below for our proposed algorithm. A distorted FBG spectrum illustrates the issues.

Curve Fitting Algorithm based detection:
There are several types of curve fitting algorithms, and polynomial curve fitting and Gaussian curve fitting are more popular than others. Curve fitting algorithms assume that the shape of the FBG sensor spectrum profile matches the shape of some family of mathematical functions (like Gaussian or Polynomial). Then, with the available sample points, the algorithm tries to fit a particular function of that family to the sampled points iteratively. Curve-fitting algorithms are accurate when the FBG profile shape matches the assumed family of functions. However, quite often, the spectral shape of the FBG sensor, when captured on an Optical Spectrum Analyzer, is separate from a form of a mathematical function. In cases where sparse sampling has been done, and the actual spectral shape of the FBG sensor is mathematically different from the one assumed by the fitting algorithm, the predicted curve is inaccurate. In such cases, when the FBG sensor spectrum shifts by some value, the central peak of the expected curve shifts by a different value.
In the illustration below, the central value of the FBG spectrum is shifted by 80 pm. However, the Gaussian fit of the sampled plot shifts by 90 pm.

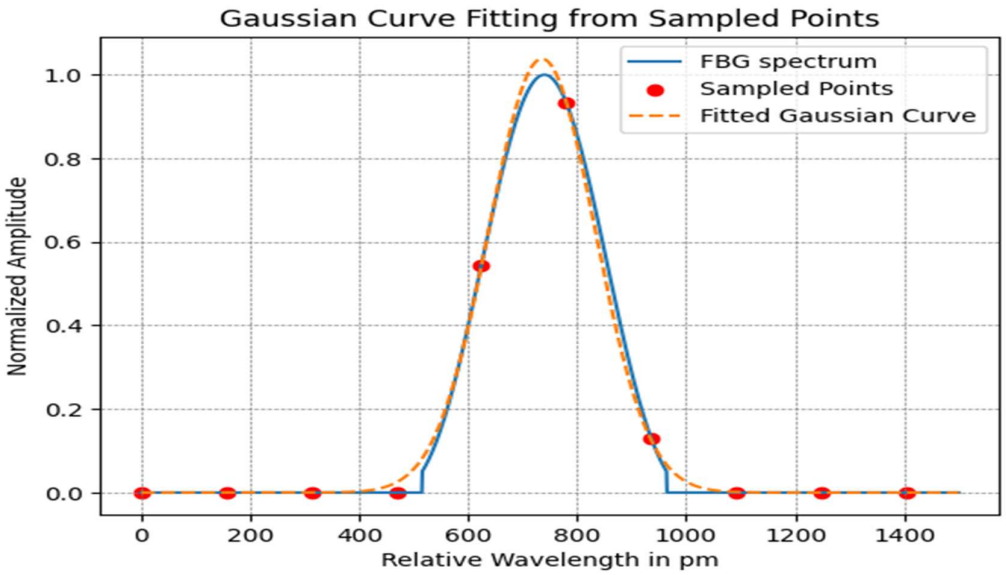


Fig. 3.6.A simulated FBG peak with central wavelength 740 is sampled, and the Gaussian curve is fitted. The fitted curve's central wavelength is 735

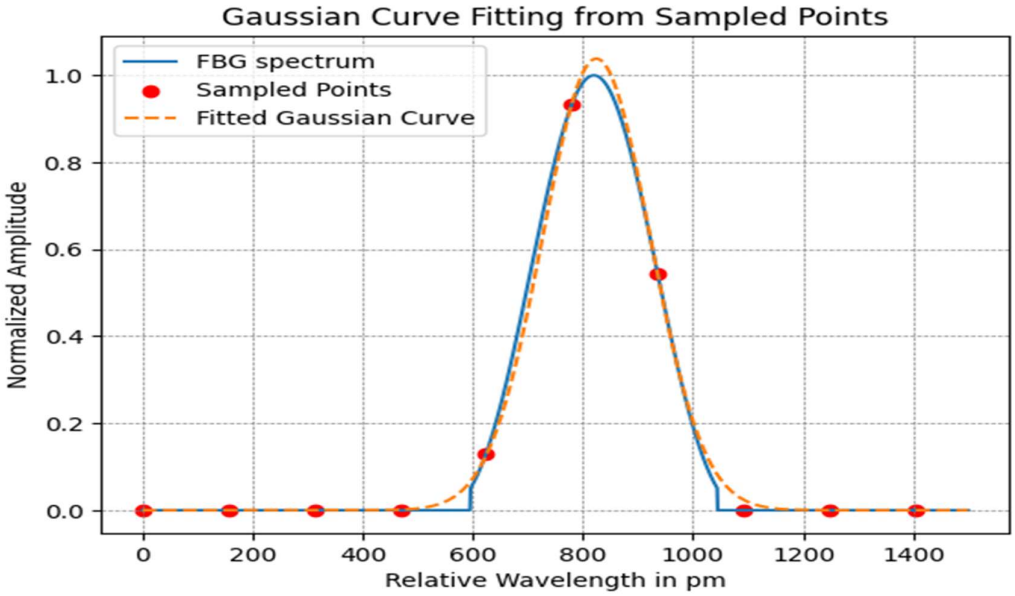


Fig. 3.7.A simulated FBG peak with central wavelength 820 is sampled, and the Gaussian curve is fitted. The fitted curve's central wavelength is 825

However, this curve fitting-based FBG Bragg wavelength detection technique has been reported to be more accurate than centroid algorithm-based Bragg wavelength detection [4].

However, the main limitation of this curve-fitting approach stems from the fact that it is an iterative process [1] and, therefore, is unsuitable for real-time applications with stringent needs for efficient time performance.

In contrast, the proposed ANN-based algorithm for central wavelength detection is iterative only during the model's training. Once a supervised learning technique determines the model parameters, this model can be deployed in the field for prediction by an algorithm that is not iterative. Hence, the proposed ANN-based algorithm overcomes the limitation regarding the real-time processing of the FBG sensor data.

Centroid Finding Algorithm based detection:

The centroid finding algorithm has the advantage that it is not an iterative algorithm. It can be implemented on an FPGA for real-time applications requiring efficient time performance. However, the centroid finding algorithm-based FBG sensor spectrum central wavelength detection suffers from accuracy-related issues when sparse sampling of a distorted spectral peak is done. When we see the centroid of a spectral peak, it is the centroid of the polygon formed by joining the sampled points by straight lines. When the sampling is dense, the shape of the polygon is very similar to that of the spectrum. However, when the sampling is sparse, the shape of the polygon is quite different from the spectrum's shape.

So, in the case of sparse sampling, while doing CCD array-based interrogation, when the FBG spectrum shifts in wavelength, the shape of the polygon made by sampling points on the FBG sensor spectrum changes. As a result, the polygon's centroid may shift by a different value compared to the shift of the FBG sensor spectrum.

In the following section, we have illustrated that when an FBG peak is sampled 156 pm apart, and central wavelength detection is done using a centroid finding algorithm, the Error can reach up ± 8 pm with a mean absolute error of 2.92 pm. The FBG peak used for this computation has a distorted profile because amplitude samples less than 0.1 are ignored.

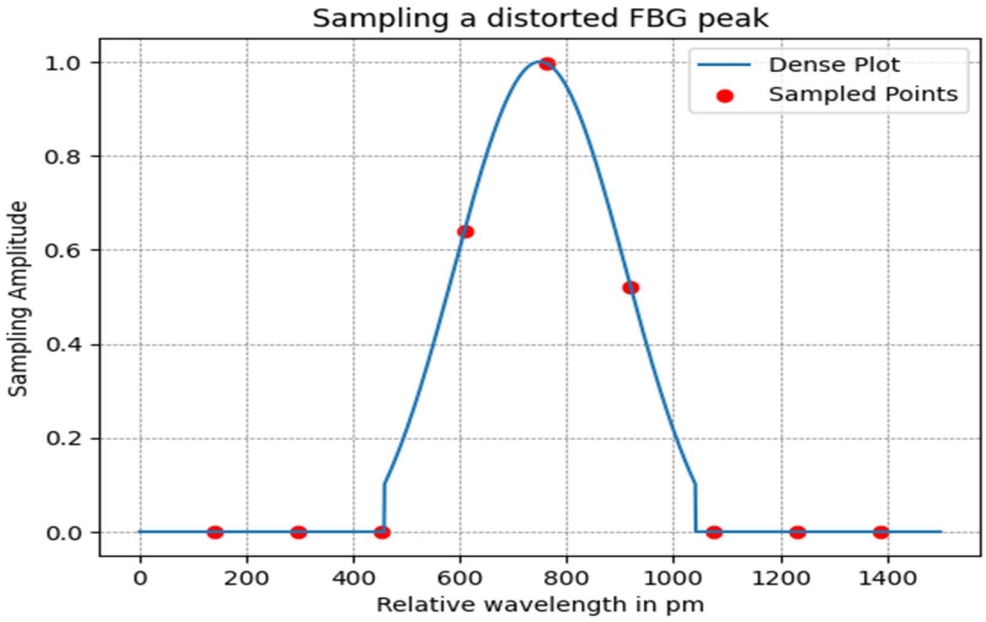


Fig. 3.8. A simulated FBG peak with central wavelength 750 is sampled at intervals of 156 pm.

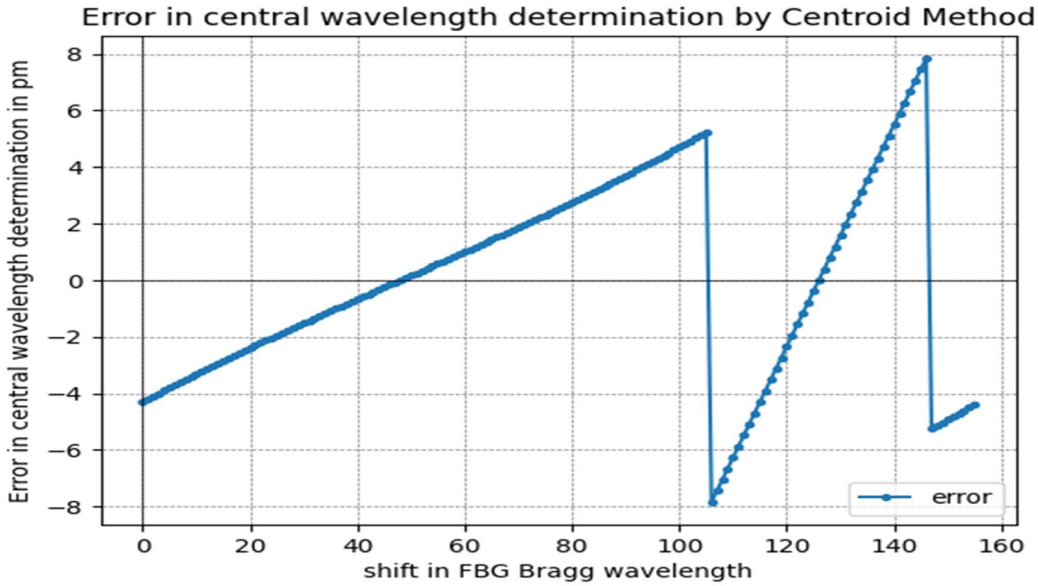


Fig. 3.9. Error in central wavelength computed using centroid algorithm when the peak shifts left in wavelength and sampling is done at intervals of 156 pm. The leftward shift in wavelength in pm is the x-axis.

Our proposed ANN algorithm takes the centroid value as one of the input features. However, to address this Error in central wavelength determination for a given FBG sensor spectrum using the centroid method, the values of all the samples capturing the spectrum are also fed in as inputs in the training data. As a result, the prediction accuracy improves in our ANN-based algorithm, and it is limited to ± 2 pm with a mean absolute error of 0.513 pm for the same distorted profile FBG peak.

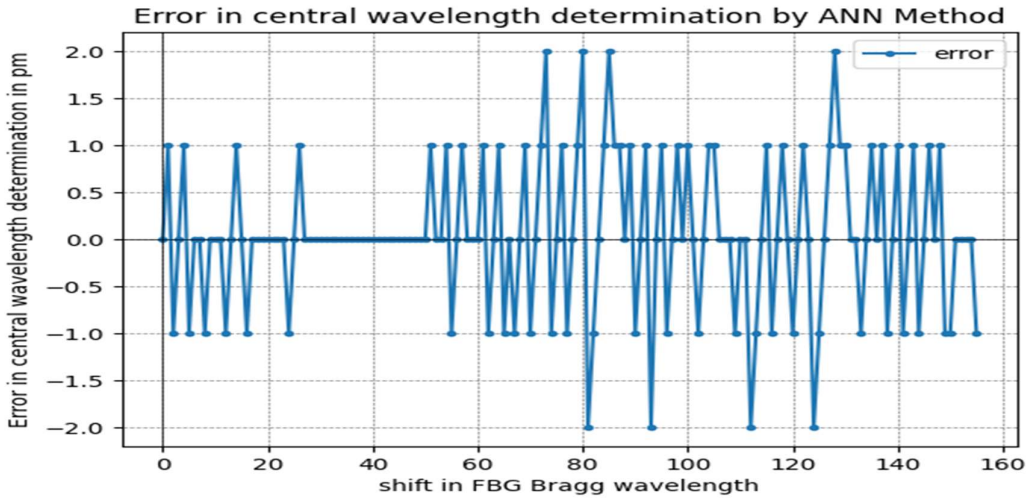


Fig. 3.10. The error in central wavelength is computed using the proposed ANN algorithm when the peak shifts left in wavelength. Sampling is done at intervals of 156 pm. The leftward shift in wavelength in pm is the x-axis.

Chapter-4: Tunable Filter-based FBG Interrogation using ANN algorithm

This chapter proposes a novel ANN-based algorithm for a Tunable Filter-based interrogation. A model using the proposed algorithm has been developed. This developed model takes care of the non-idealities like non-repeatability and non-linearity of tunable filter wavelength vs. voltage characteristics and temporal fluctuations in the intensity of the optical signal applied to the photodetectors. Further, once developed by the proposed algorithm, the model remains usable and valid for predicting the central wavelength of the FBG spectral peak even if it shifts significantly from its nominal position. Moreover, the proposed model is an ANN, which can also be implemented on an FPGA, making it very suitable for real-time application.
In the subsequent sections of this chapter, we first discuss the setup and methodology of FBG interrogation using a Tunable filter algorithm for training the model, followed by a prediction algorithm using the model. Finally, we compare the performance of our proposed model with the existing algorithms, such as curve-fitting and centroid-finding algorithms.

4.1 Set-up & Methodology of FBG Interrogation

This section outlines the configuration and methodology employed in Tunable Filter-based interrogation. This method involves generating a voltage-tunable laser source using a tunable filter alongside a 980 nm pump laser diode, an Erbium-doped active fiber, WDM optical couplers, optical splitters, and isolators, as discussed in Chapter 2, section 2.2. Due to the limitations of the tunable filter, such as poor repeatability and non-linearity, these constraints are inherited by the resulting tunable laser. To mitigate these issues, the laser output is split into two branches in a 90:10 ratio using an optical splitter. The light from the 10% branch is directed to an athermal etalon, and a photodetector captures the transmission spectrum. The light from the 90% branch is directed to both reference FBGs (with known Bragg wavelengths) and sensor FBGs (with unknown Bragg wavelengths) via a circulator, and the reflection spectrum from FBGs is recorded by a photodetector(s). The schematic is depicted in Figure 4.1. The etalon and the reference FBG in this setup are utilized to devise a method to mitigate the issues stemming from the filter's poor repeatability and non-linearity.

In this interrogation setup, the tunable laser sweeps across its entire spectrum by applying a time-varying voltage as a triangular or saw-tooth waveform to the tunable filter. As this occurs, the photodetector outputs voltage samples are recorded over time, resulting in timestamp vs. photodetector-voltage samples. Consequently, the central wavelengths of the etalon peaks (i.e., etalon passbands) and FBG peaks (i.e., FBG reflection spectrum) are expressed in terms of timestamps in this interrogation scheme. All algorithms discussed in this paper for this interrogator, including curve-fitting, centroid, and the proposed neural network-based method, aim to predict the timestamp corresponding to the central wavelengths of FBG and etalon peaks from the recorded timestamp vs. photodetector-voltage samples of the corresponding photodetector. The timestamp of the photodetector-voltage samples represents the sequence number of the sample, whereas the computed timestamp corresponding to the wavelength of

Etalon passband or FBG is a decimal number with a precision of two digits after the decimal point, lying between two integer timestamps.

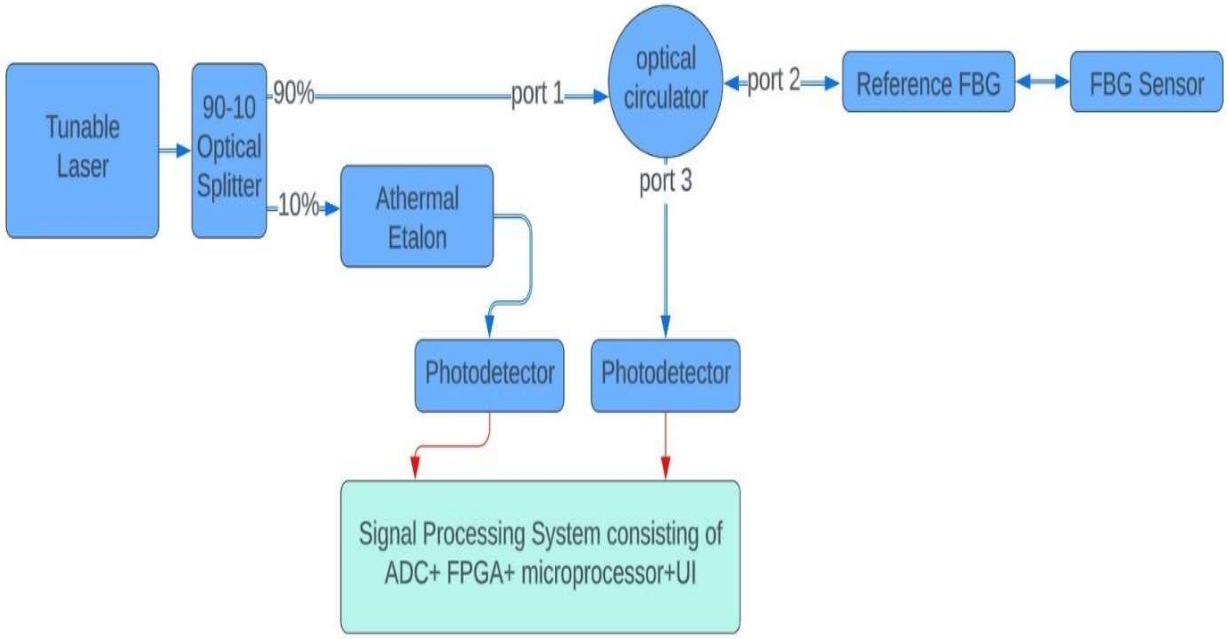


Fig. 4.2. Block diagram of Tunable Laser-based interrogation

4.2 ANN Training Algorithm for Tunable Filter-based Interrogator

Like the CCD array-based interrogation system outlined in the preceding chapter, the Tunable Filter-based interrogation system described here also exhibits variations in the recorded photodetector voltage for a given wavelength of the FBG spectrum. These fluctuations are contingent upon the intensity of the reflected light from the FBG, which is influenced by factors like the intensity of the incident light on the FBG and signal attenuation during its transmission through the optical circuit. Changes in these parameters can result in fluctuations in the photodetector voltage.

However, in a Tunable Filter-based interrogation system, additional issues arise from the poor repeatability of the filter passband characteristics with respect to the applied voltage. The relationship between the 'applied voltage to the tunable filter' and 'the passband wavelength of the Tunable filter' (the same as the wavelength of the tunable laser output) is inconsistent, as mentioned in [2,7]. If we plot this relationship in a graph, we obtain a plot with a different slope for each sweep of the tunable filter. As a result, even if we record photodetector samples at equal time intervals, from one sweep to another, they correspond to different intervals in wavelength.

When we plot timestamp vs amplitude in time-voltage samples, samples with the same timestamp correspond to different wavelengths for different sweeps. Therefore, the neural network training algorithm should be able to address this additional issue as well.

Unlike a CCD array-based interrogator discussed in the previous chapter, in this case, the wavelength correspondence of the samples is not known beforehand, and it changes in each sweep of the tunable LASER. In the context of Tunable Filter-based interrogation, we employ an ANN algorithm to enhance the sampling resolution by a specific factor. In our illustration, we have designed and trained the neural network to increase the sampling resolution by 100 times. This means that if we capture one sample every 100 picometers of the wavelength sweep

of the Tunable LASER, the ANN aims to predict the accuracy of the Bragg wavelength with a precision of 1 picometer. The actual sampling resolution (pm per sampling interval) varies from one sweep to another in Tunable Filter-based interrogation due to its poor repeatability. Even within a given sweep, the sampling resolution of two samples quite far from each other may not be the same because of the Tunable Filter's nonlinearity. However, since the nonlinearity is typically not very pronounced, the sampling resolution of neighboring samples is similar or practically identical.

We can utilize an athermal etalon and a reference Fiber Bragg Grating (FBG) to address this issue, as discussed in [2, 7]. The etalon's 'gain vs wavelength characteristics' exhibit a train of peaks in its spectrum as it has a series of passbands. Each passband in the series corresponds to a spectral peak in the train of peaks and to some specific wavelength in the ITU-T grid. However, which peak corresponds to which wavelength of the ITU-T grid was not known before because of the non-repeatable nature of the Tunable filter characteristics; hence, a reference grating needs to be introduced to solve the issue. Using a reference FBG with a known Bragg wavelength, we can identify the wavelengths of the etalon's passband peaks adjacent to the reference FBG's wavelength. The sub-sample timestamp of the reference FBG will lie between the sub-sample timestamps of these two spectral peaks. Once the corresponding wavelengths of these two etalon spectral peaks are known, the wavelength corresponding to all the spectral peaks can be derived.

Now, the spectral peaks of the etalon passbands are pretty close to each other in wavelength, with the typical wavelength difference between two adjacent peaks being approximately 800 pm. Therefore, once the wavelengths of all the etalon spectral peaks are determined, they divide the whole sweep range of the tunable LASER into small intervals. Hence, even though the voltage-wavelength characteristics of the tunable LASER are non-linear, each small interval between two consecutive wavelengths of the etalon passband peaks may be considered a piecewise linear segment. Hence, the instantaneous wavelength of a sensor FBG in such a given interval can be deduced from its sub-sample timestamp by linearly interpolating corresponding wavelengths of the etalon peaks on both ends of the interval with respect to their sub-sample timestamps. The corresponding wavelengths of the athermal etalon peaks defining such an interval can be obtained from the ITU-T chart. Hence, computing the sub-sample timestamps from the timestamps of time-voltage samples must be done for both etalon and FBG peaks to complete the process. Several algorithms are already being used to achieve this task. In this paper, we propose another algorithm to accomplish the same thing. Our proposed algorithm is an ANN-based algorithm that offers some advantages over the existing algorithms.

In Tunable filter-based interrogation, whenever the filter starts its sweep from one end of the filter spectrum, a counter is also started to count its sequence number (timestamp). Every photodetector-voltage sample has a sequence number, which is also called its timestamp. These timestamps are, therefore, consecutive whole numbers. As the sweep progresses, there are adjacent samples with voltages surpassing a certain threshold recorded on the photodetector that implies some FBG or etalon spectral peak in that region. This peak may be due to passbands of the etalon, sensor FBG, or reference FBG. Each such peak is recorded by several consecutive time-voltage samples (i.e., photodetector-voltages with timestamp). In this interrogation process, the function of our proposed Artificial Neural Network (ANN) model is to determine, for each peak (associated with the etalon and FBG), a timestamp represented as a decimal number with a precision of two digits after the decimal point. This decimal fraction timestamp (sub-sample timestamp) is computed from the photodetector-voltage sample values and timestamps that capture an etalon or FBG peak. The sub-sample timestamp thus calculated for each peak (either etalon peak or FBG peak) corresponds to the central wavelength of that peak. (For an FBG peak, the central wavelength means its Bragg wavelength).In the case of reference grating, this sub-sample timestamp is mapped to the known Bragg wavelength of the

reference FBG. This sub-sample timestamp is mapped to a wavelength in the etalon wavelength chart (ITU-T chart) for an etalon peak. For sensor FBG, this sub-sample timestamp's corresponding wavelength is computed by linear interpolating the adjacent etalon peaks' corresponding wavelengths.

Similar to the CCD spectrometer-based interrogation discussed in the previous chapter, this work proposes a two-layer ANN for Tunable filter-based interrogation and for predicting the sub-sample timestamp from the time-voltage samples. In this case, the problem of predicting the sub-sample timestamp is also framed as a classification problem. The input layer here has 15 nodes, as it is assumed that an etalon or FBG peak can span over a maximum of 15 samples. The output layer has 100 nodes as the sub-sample resolution (i.e., the resolution of the sub-sample timestamp) is targeted as 100 times the sampling resolution (i.e., the resolution of a captured time-voltage sample timestamp, i.e., the corresponding wavelength difference between two consecutive time-voltage samples). Each output node corresponds to one integer number between -40 to +50. When the amplitudes of 15 time-voltage samples of a spectral peak are fed as input to the ANN after normalization (i.e., by scaling the amplitude as such that the highest amplitude is 1) and centering (i.e., by positioning the samples in a manner that the highest amplitude sample position corresponds to the middle input node), it outputs an integer between -49 and +50. Now, if the sample's maximum amplitude timestamp is T_{max} and the output of the ANN is Y, then the predicted sub-sample timestamp of the spectral peak is '$T_{max}+. \frac{Y}{100}$'.

In this paper, the procedure of training such an ANN, which can compute a sub-sample timestamp from the timestamps of the samples, has been illustrated below with respect to Fig 4.2. Let the spectral peak, which has to be detected from its time-voltage samples, be a 200 pm wide symmetric Gaussian peak with an FWHM of 50 pm. Now, we are considering the normalized version of this peak for training. We do the training in the following steps.

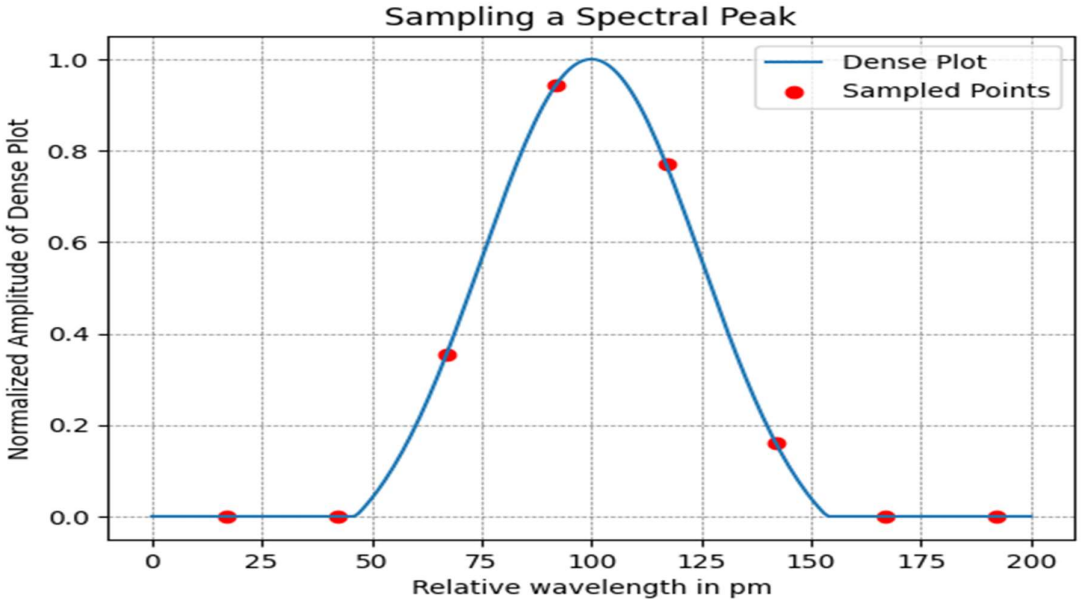


Fig. 4.2. The spectral peak dense plot (blue line) is sampled at discrete points (red dot) (target value +32)

Steps of Training Algorithm: Similar to CCD array-based interrogation discussed in the previous chapter, the training data is a matrix in which each row is a training sample containing the pattern set and a target value**.** In the following steps the first five steps describe how the pattern set of the training data is formed. The 6th step describes how the corresponding target value for each pattern set is computed. The following steps generate the training data for a given FBG sensor spectrum:

1. First, the normalized profile of the FBG sensor spectrum is plotted. This is done in two steps. First, the reflection intensity value of the sensor is densely tabulated in a row for every wavelength in steps of 1 pm. Then, this FBG spectrum is normalized by dividing all the values in the row by the maximum value. So, after normalization, the peak value of the FBG spectrum is 1. This spectrum is plotted with the x-axis representing the wavelength axis of the spectrum. This wavelength, denoted by the x-axis, is not the actual wavelength of the spectrum but a relative wavelength with respect to the lowest wavelength in the FBG sensor spectrum. The unit of measurement on the x-axis is picometers (pm). In the example provided in Figures 4.2 and 4.3, a single row comprises 200 values, as the FBG spectrum is plotted for a range of 200pm.
2. After the FBG spectrum profile is plotted, next in our illustrated case, sparse sampling of the spectrum at different points by varying sampling intervals and initial sampling point is done. In contrast to the CCD array-based interrogation case, where the consecutive samples always had a fixed number of picometers apart in wavelength, the wavelength difference between consecutive time samples in one sweep may differ. Here, we considered that the sampling interval can vary from 20 to 40pm. In our illustrated case, when the training samples are 8 pm apart, pattern sets are generated by sampling the spectrum at x= [0, 20, 40, 60, 80, 100, 120, 140, 160, 180], x= [1, 21, 41, 61, 81, 101, 121, 141, 161, 181], x= [2, 22, 42, 62, 82, 102, 122, 142, 162, 182] and so on till x= [19, 39, 59, 79, 99, 119, 139, 159, 179, 199]. After that, we get the same pattern sets again. Again when the training samples are 21 pm apart, pattern sets are generated by sampling the spectrum at x= [0, 21, 42, 63, 84, 105, 126, 147, 168, 189], x= [1, 22, 43, 64, 85, 106, 127, 148, 169, 190], x= [2, 23, 44, 65, 86, 107, 128, 149, 170, 191] and so on till x= [20, 41, 62, 83, 104, 125, 146, 167, 188]. After that, we get the same pattern sets again. Similarly, when the training samples are 40 pm apart, pattern sets are generated by sampling the spectrum at x= [0, 40, 80, 120, 160], x= [1, 41, 81, 121, 161], x= [2, 42, 82, 122, 162] and so on till x= [39, 79, 119, 159, 199]. After that, we get the same pattern sets again.
3. All the amplitude values in each such pattern set are then divided by the maximum value of the sample amplitude in the set. Therefore, the maximum amplitude value in the pattern set becomes 1, and the rest of the amplitude values in that set become ≤ 1. This step is named ‘pattern-normalization’ in this document.
4. Now, we center the sample points so that the highest sample is in the middle of the row. In our case, we will take a 15-point row. So, the highest amplitude value (1 after pattern normalization) is kept at index 7 (indexing starts at 0). All other sample values are mapped accordingly. Where there is no value to map, it is padded with zero. This step is named 'pattern centering' in this document.

In our illustrated example (Fig: 4.2), after normalization and centering, we get the following:

Table 4.1. Table showing Normalized and Sample Amplitude

Timestamp relative to the peak	Normalized and Centered Amplitude
0	0
1	0
2	0
3	0
4	0
5	0
6	0.375
7	1
8	0.816
9	0.169
10	0
11	0
12	0
13	0
14	0

5. Then, the centroid of this normalized and centered set is computed using the centroid formula mentioned in the previous chapter.

$$centroid = \frac{\sum_{n}\{n*max(0,(A_n - A_{mp}))\}}{\sum_{n} max(0,(A_n - A_{mp}))} \quad , n = 0, 1, ...14$$

Where n is the index number, A_n is the normalized and centered pattern set Amplitude, and A_{mp} is the minimum positive amplitude in the pattern set.

Now, finally, the one at index seven is replaced by centroid –6 = 7.262 -6 = 1.262

So, now the final pattern set for (Fig: 4.2) is given in Table 4.2

Table 4.2. Table showing Normalized and Centered time-sample Amplitude with centroid

Input Node Number	Values applied at Input Node
0	0
1	0
2	0
3	0
4	0
5	0
6	0.375
7	1.262
8	0.816
9	0.169
10	0
11	0
12	0
13	0
14	0

6. The target value for a particular sample set is derived using the following steps. Subtract the x-coordinate of the sample with the highest amplitude in the sample set from the x-coordinate of the central wavelength of the dense plot of the spectral peak. Then, divide the difference by the 'wavelength interval (in pm) between two consecutive samples' in the sample set. Then multiply the quotient by 100 and round to the nearest integer. This is illustrated as follows.

In the case illustrated in Fig: 4.2, the central wavelength of the spectral peak lies at x=100. The spectral peak is sampled at x = [17, 42, 67, 92, 117, 142, 167, 192], i.e. 25 pm apart starting from 5 pm offset. The maximum valued sample is captured at x=92. So, we subtract 92 from 100 and divide it by 25. We get 0.32 multiplied by 100 and rounded to the nearest integer. So, we get the target value for this sample set as +32.

Another training sample is illustrated in the following figure (Fig 4.3). Here, the central wavelength of the spectral peak lies at x=100. The spectral peak is sampled at x = [15, 38, 61, 84, 107, 130, 153, 176, 199], i.e., 11 pm apart starting from the 15 pm offset. The maximum valued sample is captured at x=107. So, we subtract 107 from 100 and divide it by 23. We get -0.304347…. which is multiplied by 100 and rounded to the nearest integer. Therefore, we get the target value for this sample set as -30.

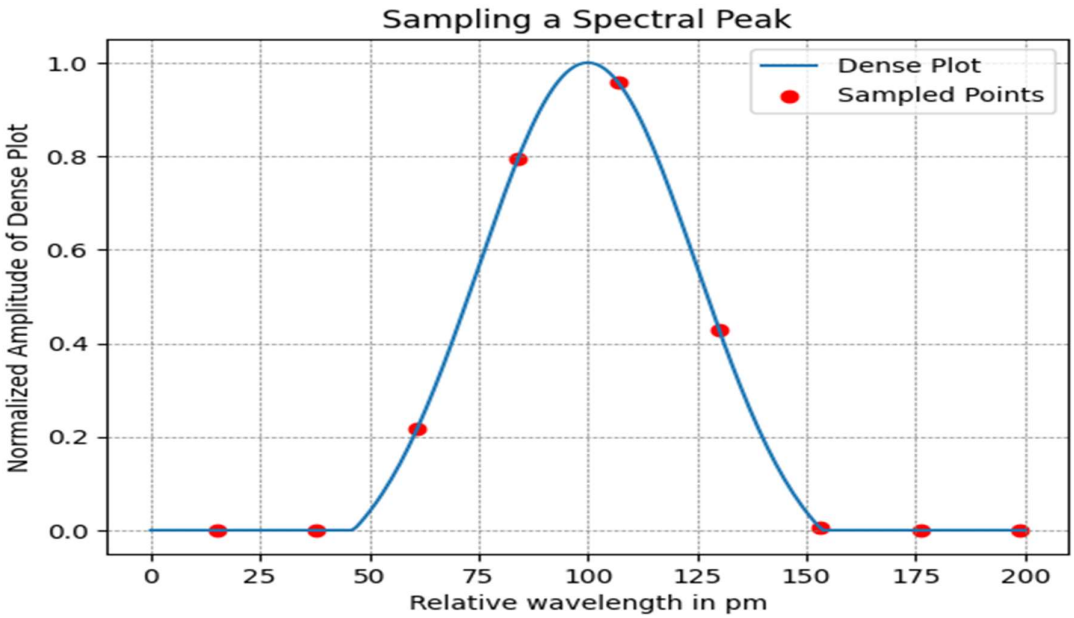


Fig.4.3. The spectral peak dense plot (blue) is sampled at discrete pts (red dot) (target value: -30)

She proposed an Artificial Neural Network. The ANN for Tunable Filter-based interrogation also has two layers. The input layer comprises 15 nodes corresponding to the 15 elements of normalized, centered, and padded rows, with the central element (index 7) replaced with the value 'centroid-6'. The input layer is densely connected to the nodes of the output layer. The output layer has 100 nodes. The Adam optimizer was used with the softmax activation function to train this neural network, and the learning rate was 0.1.

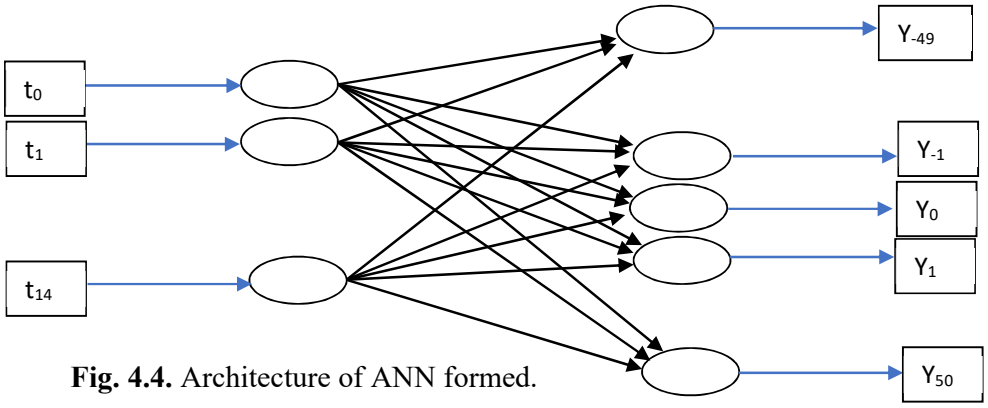


Fig. 4.4. Architecture of ANN formed.

The algorithm to generate and save the training data in a file is described in the following 'ALGORITHM: 4.1', and the algorithm to build a model from this training data is described in 'ALGORITHM: 4.2' respectively.

Algorithm 4.1: Pseudocode to generate and save the training data for Tunable Filter based Interrogation:

```
# Function for Gaussian curve
FUNCTION gaussian_waveform(x, amplitude, mean, stddev)
   RETURN amplitude * EXP(-(x - mean)^2 / (2 * stddev^2))

# Parameters
num_points = 201 # Number of points (pico-meter) in the FBG spectrum
amplitude = 1 # maximum value of the peak, i.e., normalized plot
mean = 100 # this peak has center at position  index 750 pm (relative)

x_values = ARRAY of length num_points initialized with values from 0 to num_points-1 # making the x-axis of
Gaussian waveform simulating dense plot of FBG spectral peak

# Generate the distorted Gaussian waveform
gaussian_wave = CALCULATE [(gaussian_waveform(x_values, amplitude, mean, stddev=25) – 0.1)/0.9]
#making the distorted Gaussian wave, which simulates a dense plot of FBG spectral peak
threshold = 0.0 # minimum value of the waveform is zero, i.e. no negative amplitude
FOR each element in gaussian_wave:
      IF element < threshold:
         set element to 0
```

```
# Here we sample the Gaussian waveform at intervals and make a training data matrix
# Initialize lists for sampled points and values
lst_sample_pts = Empty  List of arrays  # list of arrays (rows) of sampled wavelengths
lst_sample_vals = Empty  List of arrays # list of arrays (rows) of sampled values

# Sample the Gaussian waveform at variable intervals
min_num_samples = 20
max_num_samples = 31
FOR num_samples from min_num_samples to max_num_samples - 1:
   FOR starting_pt from 0 to num_samples - 1:
      sampling_points = array of integers in the range from zero upto num_points sampled at values starting
                        from starting_pt with step size num_samples
      APPEND sampling_points to lst_sample_pts
      APPEND 'gaussian_wave sampled at indices which are elements of array sampling_points' to lst_sample_vals

# Here, the sampled values are normalized by dividing each row by its max element
FOR i IN RANGE(number of rows in lst_sample_vals):
   max_value = MAXIMUM value in lst_sample_vals[i] # Find the maximum element in the current row
   # Divide each element of the row by the maximum element of the row
   each element of lst_sample_vals[i] = (each element of lst_sample_vals[i]) / max_value

# max_indices contains the indices of maximum sampled values(i.e. 1) of each row before centering
max_indices = ARRAY of indices containing the index of the maximum element in each row of lst_sample_vals


# In this step, the sampling wavelength and sampled value arrays are padded with a calculated number of
zeros
  such that the number of elements in each array is 15. This step is required before centering arrays
For each row in lst_sample_pts:
   padded_lst_sample_pts  = row padded with [15- len(row)] number of zeros to make its length 15
   APPEND row to training_matrix1
   # training_matrix1 contains the suitably padded array of sampling wavelengths

For each row in lst_sample_vals:
   padded_lst_sample_vals  = row padded with [15- len(row)] number of zeros to make its length 15
   APPEND row to training_matrix1g
  # training_matrix1g contains the suitably padded array of sampled values



# Next, the sampled values and sampling points arrays are properly centered so that the element at index
position 7 has the highest sampled value and its corresponding sampling wavelength, respectively.
# The centered sample value array (centered_array_g) will be used to make a pattern set (i.e., X of training
data)
# centered sampling wavelength (centered_array) will be used to compute the 'wavelength difference
between FBG central wavelength and corresponding wavelength of max sampled value' (target of training
data)
centered_array = 2D ARRAY of zeros with dimensions same as training_matrix1 # initialized with zero
FOR i IN RANGE(length of training_matrix1):
   centered_array[i][(7 - max_indices[i]) to end of array] = training_matrix1[i][ 0 to (8+max_indices[i])]

centered_array_g = 2D ARRAY of zeros with dimensions same as training_matrix1g # initialized with zero
FOR i IN RANGE(length of training_matrix1g):
   centered_array_g[i][(7 - max_indices[i]) to end of array] = training_matrix1g[i][0 to (8+max_indices[i])]
# The corresponding wavelength of the maximum sampled value is stored in max_sampled_val_wl
# This wavelength can be found as it will be in index 7 of each array of centered_array due to centering
max_sampled_val_wl = ARRAY containing wavelengths at index 7 of each row of centered_array matrix
```

```
# shift_wl is the wavelength difference between FBG central wavelength and the corresponding wavelength of max sampled value
# This will target training data (i.e., y of training data).
shift_wl_abs = 750 - max_sampled_val_wl
shift_wl = round_to_nearest_integer [(shift_wl_abs/(training_matrix1[column_1 ] - training_matrix1[ column_0]*100)

# Here, the X and y of the  training data are joined (normalized centered pattern data is joined with target value)
result_2d_array = CONCATENATE centered_array_g with shift_wl

# Next, from the training matrix result_2d_array, systematically lower amplitudes in the pattern set are dropped, and the highest amplitude (which is always one due to normalization) is replaced with the centroid value of the retained amplitudes to make a new training matrix np_matrix10. While dropping the lower amplitudes, it is ensured that at least two amplitudes in the pattern set of each training sample, i.e., the highest two amplitudes in each pattern set, are always retained. While computing the centroid, instead of considering the normalized amplitude directly, we believe the differential value of normalized amplitude with respect to the smallest retained normalized amplitude in the pattern set.
np_matrix = result_2d_array
np_matrix10 = EMPTY 2D ARRAY with zero rows and 16 columns
# np_matrix10 will be  used to store the complete training data, including patterns in which lower amplitudes have been discarded
max_iterations = 15
# Since there are 15 elements in each row of the pattern set, no more than 15 iterations are required to form a pattern set by discarding lower-valued samples
np_matrix_w1 = COPY of np_matrix # making a working copy of the training set

FOR iteration_count IN RANGE(1, max_iterations + 1): #Iterate for generating training data
{
   change_flag = 0 # flag is set if there is any change with respect to the previous iteration
   min_row_pos_num = Empty  List # minimum positive element in each row will be stored in this list
   gt_min_row_num = Empty  List # count of amplitudes greater than minimum positive element in each row
                                  will be stored in this list
   min_row_pos_index = Empty  List # indices of the minimum positive element in each row will be stored here

   # in the following section, we check some status of np_matrix_w1 matrix related to its minimum positive
     element, and number of values above the minimum positive element in each row
   # while performing row-wise iterations, the last column containing target values of training data is dropped
   FOR row IN np_matrix_w1 with target column dropped:
      positive_numbers = POSITIVE numbers in row
      IF length of positive_numbers > 0:
         min_positive = MINIMUM value in positive_numbers in row
         gt_min_pos = COUNT of numbers in positive_numbers GREATER THAN min_positive in row
         min_index = INDICES of row where min_positive is present
         APPEND min_positive to min_row_pos_num
         APPEND gt_min_pos to gt_min_row_num
         APPEND min_index to min_row_pos_index
      ELSE:
         APPEND 0 to min_row_pos_num
         APPEND 0 to gt_min_row_num
         APPEND -1 to min_row_pos_index

# next np_matrix_w2 and np_matrix_w3  matrix is created from np_matrix_w1.
   np_matrix_w2 = COPY of np_matrix_w1 # np_matrix_w2 will be used for computing centroid
   np_matrix_w3 = COPY of np_matrix_w1 # np_matrix_w3 is for appending to training data
```

```
   FOR i IN RANGE(number of rows of np_matrix_w1):
      IF gt_min_row_num[i] > 2: #if there are at least three numbers in a row greater than the current min
positive value
         first 15 elements of np_matrix_w2[i] = first 15 elements of np_matrix_w1[i] - min_row_pos_num[i]
         np_matrix_w1[i][min_row_pos_index[i]] = 0 # make the minimum positive value of the row as 0
         set the negative values in the first 15 elements of np_matrix_w2[i] = 0
         change_flag = 1 #if there is any new training data in this iteration, this flag gets set

   # Next, np_matrix_w3 matrix is created for appending to training data
   sum_of_product_numerator = SUM (elements in first 15 columns of np_matrix_w2 * array of 0 to 14)
   sum_denominator = SUM of elements in first 15 columns of np_matrix_w2
   centroid = sum_of_product_numerator / sum_denominator #elementwise array division
   np_matrix_w3[ 7] = (centroid - 6) # index 7 of all rows of np_matrix_w3 replaced in elementwise array
                                          substitution
   np_matrix10 = CONCATENATE np_matrix10 with np_matrix_w3

   # If there is no change in the iteration, then break the iteration
   IF change_flag == 0: # flag=0 means NO change in this iteration
      BREAK
}



# Save training data to a CSV file
SAVE np_matrix10 to file_path as CSV
```

Algorithm 4.2: Pseudocode to build model from the training data for Tunable Filter based Interrogation:

```
np_matrix10 = READ_CSV(training data)
np_matrix11 = EMPTY 2D ARRAY

FOR i FROM 0 TO 10:
   np_matrix11 = CONCATENATE np_matrix11 with np_matrix10

np_y11 = LAST COLUMN of np_matrix11
np_X11 = ALL COLUMNS of np_matrix11 EXCEPT the last one
X_train, X_test, y_train, y_test = SPLIT_TRAIN_TEST(np_X11, np_y11)

y_traink = y_train + 48
y_testk = y_test + 48

model = keras.Sequential([keras.layers.Dense(99, input_shape=(15,), activation='softmax')])
custom_learning_rate = 0.1
custom_optimizer = tf.keras.optimizers.Adam(learning_rate=custom_learning_rate)
model.compile(optimizer=custom_optimizer, loss='sparse_categorical_crossentropy',  metrics=['accuracy'])
model.fit(X_train, y_traink, epochs=15000)
SAVE_MODEL()
```

4.3 Bragg Wavelength Prediction using the proposed trained ANN for Tunable Filter-based FBG Interrogator

Prediction of sensor FBG Bragg wavelength from the time-voltage samples acquired on the photodetector in the Tunable filter-based interrogation is a little more complicated than the CCD array-based interrogation described in the previous chapter. This is because, unlike CCD array-based interrogation, where there is a direct correspondence between pixel number and wavelength, in the case of Tunable filter-based interrogators, there is no such fixed correspondence between the sample timestamps and wavelength. So, for every sweep and every spectral peak, the sub-sample timestamp of the peak needs to be determined first by our proposed ANN algorithm, and then its correspondence with the wavelength has to be established by using the known Bragg wavelength of reference FBG and known passband wavelengths of athermal etalon. This method of prediction is described below.
When the Tunable laser is driven by a triangular or saw-tooth shaped voltage waveform, it sweeps its wavelength range. We record the timestamp and voltage values whenever we detect a few consecutive samples on the photodetector greater than a threshold. Then, we first normalized the samples by dividing all the voltage values in the sample set by the maximum voltage value. Next, we do centering by putting the maximum amplitude sample at index 7 of a 15-element array. All other samples are re-positioned accordingly to maintain the relative order of samples—empty positions in this array, where no corresponding element is filled with zero values. Then, the centroid of the array elements is computed. Six is subtracted from the calculated centroid value, and the resultant value is placed in index 7 of the array. Then, these 15 elements of this array are fed into the trained ANN model, which outputs an integer in the range [-49 to +50]. This output is divided by 100 and added to the timestamp if the sample has the maximum amplitude to determine the timestamp of the spectral peak.

The following example illustrates the prediction method:

Say we get five consecutive samples greater than the threshold voltage of 0.5 V on a photodetector. The voltage values and their timestamps are given below:

Voltage values in V	1.612	4.3	3.509	0.727
Timestamps	101	102	103	104

As 4.3 is the highest value here, all the voltage values are divided by 4.3 to normalize the values. So, after normalization, we get the following.

Voltage values in V	0.375	1	0.816	0.169
Timestamps	101	102	103	104

Now, these values are copied in a 15-element array such that the value one at timestamp 102 is mapped to index 7. This is called centering, and we get the following after centering:

Table 4.3. Table showing Centered Normalized and Sample Amplitude

Array Index	Centered and Normalized Sample Value
0	0
1	0

2	0
3	0
4	0
5	0
6	0.375
7	1
8	0.816
9	0.169
10	0
11	0
12	0
13	0
14	0

Now, we calculate the centroid value of this array by the centroid formula as mentioned in section 3.1
And then we get centroid as 7.262.
We subtract six from the centroid to get 1.262
So, we replace one at index 7 with this value 1.262, so we get the following table as input values to ANN.

Table 4.4. Table showing the input to ANN nodes

Node no.	Input Values to ANN
0	0
1	0
2	0
3	0
4	0
5	0
6	0.375
7	1.262
8	0.816
9	0.169
10	0
11	0
12	0
13	0
14	0

For these input values, the trained model of ANN will give an output value of 32. This value of 32 is divided by 100, and the result is added to the timestamp of the highest sampled value. We need to add 0.32 to the timestamp of the sample 4.3V. This timestamp is 102. Hence, the computed sub-sample timestamp for this spectral peak is 102+0.32 = 102.32.
In this way, the sub-sample timestamps of all the spectral peaks (Etalon passband, sensor, and reference FBG) are computed for each sweep of the tunable laser.
The Bragg wavelength of the reference FBG and its position among the FBGs are known in advance. For instance, if the reference FBG has a Bragg wavelength of W_{ref} nm and holds position N when sequenced in ascending order with other FBGs, then this position N can be designed to remain unaffected throughout the experiment. As its relative position N is known, its corresponding timestamp T_{ref} is mapped to its Bragg wavelength W_{ref}.

From the chart of Etalon passband wavelengths, we can see that Wref falls between the two Etalon passbands (commonly called Etalon channels). So, the channel wavelengths of Etalon passbands adjacent to the W_{ref} are mapped to the sub-sample timestamps of Etalon passbands adjacent to T_{ref}. As wavelength vs sub-sample timestamp mapping of two etalon channels is known, then wavelength vs. sub-sample timestamp mapping of all etalon channels can be derived accordingly.
Next, the Bragg wavelength of sensor FBG can be computed from their sub-sample timestamps. It has to be found from the sub-sample timestamps of the etalon passbands that the sensor FBG timestamp lies between the two etalon passband sub-sample timestamps. Then, by doing linear interpolation of the wavelengths of these two adjacent etalon peaks, the Bragg wavelength of the sensor FBG is computed.

Algorithm 4.3: Pseudocode to predict sub-sample timestamp of the spectral peak from the time-voltage samples:

```
# Normalizing sampled values
max_value = MAXIMUM value in lst_sample_vals
lst_sample_vals = lst_sample_vals / max_value
max_indices = Index of maximum sampled values in lst_sample_vals
max_sample_ts = time stamp of maximum sampled values in lst_sample_vals


# Padding sample points and values arrays
padded_lst_sample_vals = [PAD lst_sample_vals with zeros to length 15]

# Centering sample points and values arrays

centered_array_g =  ARRAY of 15 zeros
centered_array_g[(7 - max_indices) to last_index_of_array] = padded_lst_sample_vals [0  to (8 + max_indices)]

centroid_centered_g = COPY of centered_array_g
row = COPY of centered_array_g
smallest_positive = MINIMUM positive value in centroid_centered_g
   IF COUNT of row elements greater than smallest_positive >= 2:
      centered_array_g [condition: element > 0] =  (element-smallest_positive)
   sum_of_product_numerator = SUM(centered_array_g w * array of 0 to 14) #element wise product
   sum_denominator = SUM centered_array_g)
   centroid = sum_of_product_numerator / sum_denominator
   centroid_centered_g[7] = centroid - 6

model = LOAD_MODEL()

output = PREDICT_MODEL(model, centroid_centered_g)
subsample_time_stamp_of _spectral_peak = max_sample_ts + output -48
```

4.4 Implementation and Performance of the model

For a Tunable filter-interrogation system, instead of directly predicting the central wavelength of the FBG spectral peak, the ANN model is used to increase the resolution of timestamp so

that the spectral peak wavelength can be determined better as the time interval between two consecutive time-voltage samples acquired on the photodetector is divided into 100 parts. Hence, a zero error in the prediction of a sub-sample time stamp means an increase of precision by 100 times.

If the target value of a sample set is 'A' and the ANN model predicts 'B,' then the prediction error is A-B. So, the mistake in the sub-sample timestamp is by (A-B)/100. If the maximum interval of training sample wavelength is R pm, then this corresponds to the maximum possible error in wavelength by $\frac{(A-B)*R}{100}$ pm.

After generating the training samples using the training algorithm, the ANN model is constructed using the Tensorflow and Keras packages. A two-layer densely connected model is created, with computations performed on the Google Colab platform. The ANN model utilizes the Softmax activation function, and for training, the Adam optimizer is employed alongside the 'Sparse Categorical Cross entropy loss function.

During the training process, lower-valued samples are systematically truncated, ensuring that higher-valued samples primarily influence the output of the ANN during training. This enables establishing a threshold level to truncate lower-valued samples that are more susceptible to noise interference. The data undergoes training for 80,000 epochs.

The training samples are split into training and validation sets. The training set is utilized to train the ANN model. Then, the trained ANN model predicts the sub-sample timestamps for validation set samples. In our case, the sub-sample timestamp resolution is 100 times the resolution of the time-voltage samples captured on the photodetector. That means the interval between two consecutive time-voltage samples is divided into 100 steps to determine the sub-sample timestamp. When there is an error of K steps in the actual and predicted value of the sub-sample timestamp for a spectral peak, it means there is an error of K% of the sampling interval. Then, the error in the prediction of the sub-sample timestamp is tabulated below.

- The data shows that the sub-sample timestamp with zero error is predicted in 56.18% of the predictions.
- The error in sub-sample timestamp prediction is within ±1 steps in 88.73% of the cases.
- The error in sub-sample timestamp prediction is within ±2 steps in 97.27% of the cases.
- The error in sub-sample timestamp prediction is within ±3 steps in 99.54% of the cases.
- The error in sub-sample timestamp prediction is within ±4 steps in 99.90% of the cases.
- The error in sub-sample timestamp prediction is within ±5 steps in 88.73% of the cases.
- The maximum range of error in sub-sample timestamp prediction is within ±6 steps.

- The mean absolute error in sub-sample timestamp prediction is within 1 step.

4.5 Comparison with other FBG central wavelength detection algorithms

As previously mentioned, the two primary traditional algorithms for detecting the central wavelength of an FBG sensor are the curve fitting algorithm and the centroid finding algorithm. Both methods have advantages and limitations, which will be explored below compared to our proposed algorithm. These will be further discussed in the context of a distorted spectral peak.

Curve Fitting Algorithm based detection:
Similar to the case of CCD spectrometer-based interrogation, in the case of tunable filter-based interrogation, curve fitting algorithms can also be used for central wavelength detection of a spectral peak. As discussed in the previous chapter, the curve fitting algorithms also suffer from the same limitations, the most prominent of them being their iterative nature and unsuitability for real-time applications.
In contrast, the proposed ANN-based algorithm for central wavelength detection is iterative only during the model's training. Once a supervised learning technique determines the model parameters, this model can be deployed in the field for prediction by an algorithm that is not iterative. Hence, the proposed ANN-based algorithm overcomes the limitation regarding the real-time processing of time-voltage samples of spectral peak.

Centroid Finding Algorithm based detection:
The centroid finding algorithm has the advantage that it is not an iterative algorithm. It can be implemented on an FPGA for real-time applications requiring efficient time performance. However, the centroid finding algorithm-based FBG sensor spectrum central wavelength detection suffers from accuracy-related issues when sparse sampling of a distorted spectral peak is done. (Distortions arise due to the minimum threshold below which values are ignored). When we find the centroid of a spectral peak, it is the centroid of the polygon formed by joining the sampled points by straight lines. When the sampling is dense, the shape of the polygon is very similar to that of the spectrum. However, when the sampling is sparse, the shape of the polygon is quite different from the spectrum's shape.
On the other hand, when the central wavelength of a spectral peak is found using our proposed ANN model, the accuracy improves considerably. Our proposed ANN algorithm takes in the centroid value of the spectral peak as one of the input features. However, here in the ANN algorithm, along with the centroid value, all the sample amplitudes capturing the spectrum are also fed in as inputs to the model after pattern normalization and centering. As a result, prediction accuracy improves in our ANN-based algorithm compared to the centroid algorithm, especially in cases where sparse sampling is done for a distorted spectral peak. This has been illustrated for a distorted spectral peak with FWHM 50 pm sampled every 26 pm, as shown in Fig 4.5. The distortion of this spectral peak results because the normalized value below 0.2 is discarded for reasons explained before.

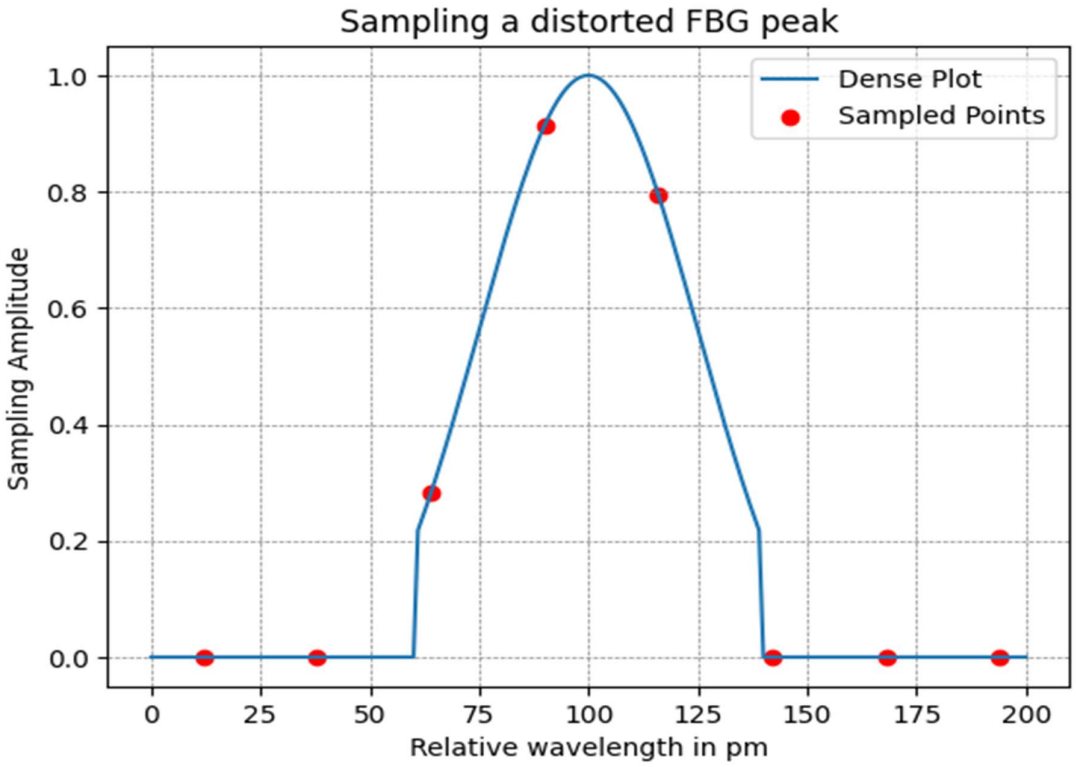


Fig. 4.5. A simulated FBG peak with central wavelength 100 is sampled at intervals of 26 pm.

We find the sub-sample timestamp for each spectral peak in the Tunable Filter-based interrogation method. The accuracy of central wavelength determination depends on how accurately we can compute the sub-sample timestamps. In the following figures (Fig: 4.6 and Fig:4.7), the error in the sub-sample timestamp has been plotted when the spectral peak is shifted leftward (i.e., towards lower wavelength) in steps of 1 pm. In Fig:4.6 and Fig:4.7, the sub-sample timestamp is computed from the time-voltage samples using the centroid and ANN algorithms, respectively. The error is measured in steps of sub-sample timestamp. Each step in sub-sample time stamp is 0.01 fraction of the sampling interval.

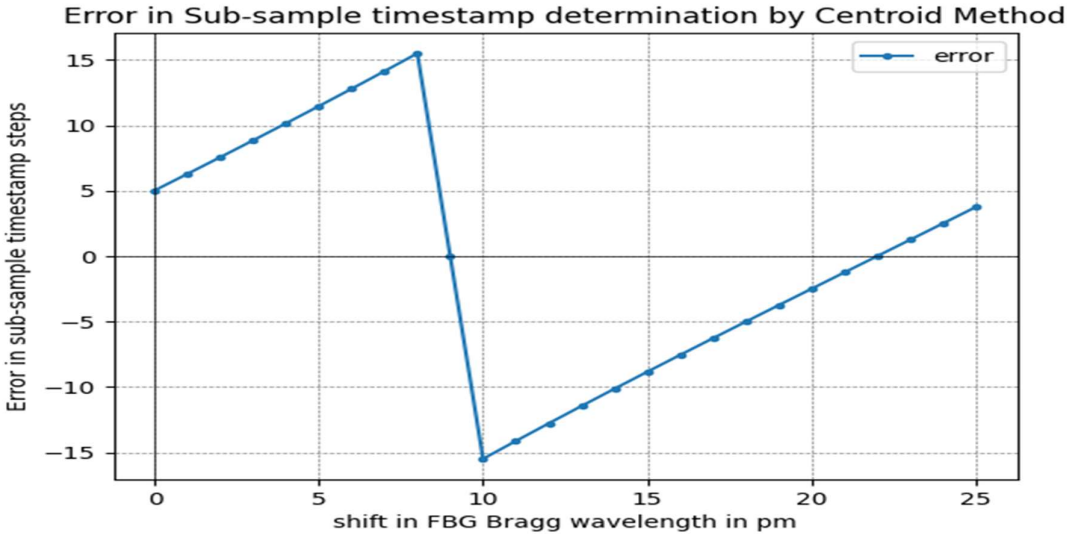


Fig. 4.6. Leftward shift of spectral peak from its nominal position vs. error in sub-sample timestamp when the sub-sample timestamp is computed using centroid algorithm

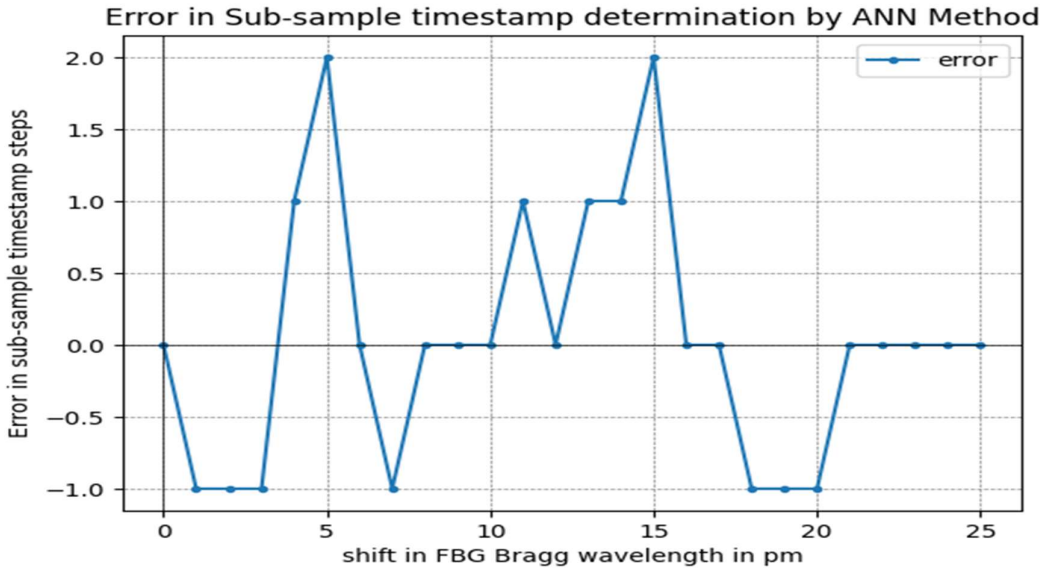


Fig. 4.7. Leftward shift of spectral peak from its nominal position vs. error in sub-sample timestamp when the sub-sample timestamp is computed using the proposed ANN algorithm

The plots show that when the centroid algorithm is used for sub-sample timestamp computation of the spectral peak, the error can range anywhere between +15 to -15 steps of 0.01 fraction of the sampling interval. That means the error varies from +15% to -15% of the sampling interval as the sub-sample timestamp target resolution is 100 times the sampling resolution. The mean absolute error in this case is more than five steps, i.e., 5% of the sampling interval. However, when we compute the sub-sample timestamp, the error is limited from +2 to -1 steps, i.e., +2% to -1% of the sampling interval with a mean absolute error less than 1% (i.e., 1 step). Now, to calculate the error in the pm scale, that can be done using the following formula:

$$\text{Error in pm} = \frac{error\ in\ steps * sampling\ interval\ in\ pm}{100}.$$

Chapter 5: Conclusion and Future Scope of Work

Referring to the results discussed in sections 3.5 and 4.5 of Chapters 3 and 4, respectively, it is evident that the proposed ANN-based algorithm Bragg wavelength detection has considerable advantages compared to other traditional algorithms currently in use.

For CCD array spectrometer-based FBG interrogation, when we use the proposed ANN algorithm for central wavelength detection of the FBG instead of the centroid algorithm, the improvement in error range improves by four times, and the improvement in the mean absolute error is more than five times. This is primarily because the proposed ANN algorithm considers the centroid value of the peak as an input feature of the ANN and other features, like the relative amplitude of the samples in the peak, as additional input features. In this work, we have also tweaked the centroid computing formula to enhance accuracy, especially for distorted spectral peaks.

Similarly, for Tunable Filter-based interrogation, where we first compute the subsample timestamps from the sample timestamps, we see the accuracy of this sub-sample determination improves significantly when the proposed ANN-based algorithm is used instead of the centroid algorithm. The improvement in accuracy is ten times when we consider the range of error, and more than five times the mean absolute error is regarded for our illustrated example. This improvement in sub-sample timestamp computation leads to accuracy in central wavelength computation for the spectral peaks (etalon and FBG) in the set-up. As the spectral peaks can now be determined with greater accuracy, the FBG sensor peak's Bragg wavelength determination also becomes more accurate.

The other algorithms usually used for Bragg wavelength determination are curve-fitting algorithms. Famous curve fitting algorithms include Gaussian fit and polynomial fit algorithms. These curve-fitting algorithms determine the central wavelength for an FBG peak spectrum in the case of CCD array spectrometer-based interrogation and sub-sample timestamp for tunable filter-based interrogation. The curve fitting algorithms are generally entirely accurate when the spectral peak profile is similar to the assumed mathematical function. However, in cases where the spectral peak profile cannot be modeled as the assumed mathematical function, then the accuracy of the curve fitting algorithm drops considerably. This limitation of the curve-fitting algorithm has been illustrated in section 3.5, Fig 3.6, and Fig 3.7.

On the other hand, since our proposed ANN-based algorithm derives the input features from the actual sampling of the dense plot of the spectrum instead of any mathematical approximated model, its accuracy does not reduce when the peak shape is distorted or quite different from a mathematical form. Hence, the ANN algorithm for peak detection is at least as accurate as curve fitting algorithms. In cases where the spectral peak shape differs considerably from the fitting function, the proposed ANN-based algorithm outperforms the curve fitting algorithms.

Apart from accuracy, the proposed ANN algorithm has another advantage over curve-fitting algorithms. Curve-fitting algorithms have an inherent limitation in that they are iterative while doing the prediction. In contrast, all the iterations of ANN are completed during the training phase. Thus, in the prediction phase, there are no iterations to be done. This makes ANN-based algorithms more suitable for deployment in real-time scenarios with stringent time constraints. Another advantage our proposed ANN-based algorithms offer is that, like centroid algorithms, these are also implementable on FPGA hardware[3]. This makes them very suitable for fast computation when high-time performance is required. Thus, our proposed ANN-based algorithms combine the advantages of both techniques (curve fitting and centroid finding). These are non-iterative and FPGA implementable, like the centroid finding algorithm, and they are reasonably accurate, like curve fitting algorithms.

The training data in our proposed ANN algorithms are generated based on the spectrum shape. Hence, the training data remains valid even if the FBG sensor spectral peak undergoes a wavelength translation, keeping the same shape. Such translation/shift in FBG Bragg wavelength may be caused by pre-strain variation and changes in ambient nominal conditions, and the proposed ANN-based algorithm adequately takes care of this.

In this thesis, we have simulated an FBG spectral peak for FBG spanning 1500 pm in chapter 3 for CCD array spectrometer-based interrogation and a spectral peak spanning 200 pm in chapter 4 for Tunable filter-based interrogation. In this thesis, the superiority of the proposed algorithm compared to the other existing algorithms was established mathematically by sampling from the simulated peaks. However, the spectral peaks of FBG and etalons come in varying shapes and sizes. Some are broad, and some are narrow. The profile shapes also differ depending on optical fiber characteristics, grating writing strength, and grating length. Therefore, a more comprehensive study may be done using acquired spectral peaks. Effects of different types of optical and electrical noise may also be studied in such acquired peaks. In this thesis, the actual implementation of the ANN models has not been done on FPGA. Therefore, in future work, an FPGA implementation of these ANN models may be done to make a prototype interrogator device.

References

1. J.Alves, J.L.Santos, A.Carvalho, and A.LagE: Fiber Bragg, Sensor Interrogation System, Based on a CCD Spectrometer. IEEE Xplore (2003)

2. Yi Jiang: High-resolution interrogation technique for fiber optic extrinsic Fabry–Perot interferometric sensors by the peak-to-peak method. Optical Society of America (2008)

3. A.Muthuramalingam, S.Himavathi, E. Srinivasan: Neural Network Implementation Using FPGA: Issues and Application. World Academy of Science, Engineering and Technology (2008)

4. Thomas Bodendorfer, Mathias S. Muller, Florian Hirth, Alexander W. Koch: Comparision of Different Peak Detection Algorithm with Regards to Spectrometric Fiber Bragg Grating Interrogation Systems. IEEE Xplore (2009)

5. Lucas Negri, Ademir Nied, Hypolito Kalinowski, Aleksander Paterno: Benchmark for Peak Detection Algorithms in Fiber Bragg Grating Interrogation and a New Neural Network for its Performance Improvement. MDPI Sensors (2011)

6. Daniele Tosi: Review and Analysis of Peak Tracking Techniques for Fiber Bragg Grating Sensors. MDPI Sensors (2017)

7. Zengbiao Chen, Chao Zhang: Analysis of Tunable F-P filter and Peak-Detection Algorithm in FBG Demodulation System. International Conference on Networking Systems of AI (2022)

8. Timur Agliullin et al.: Application of Neural Network Algorithms for Central Wavelength Determination of Fiber Optic Sensors. MDPI Applied Sciences (2023)

9. Othonos, A.; Kalli, K. Fiber Bragg Gratings: Fundamentals and Applications; Artech House: Boston, MA, USA, 1999.

10. Kersey, A.D.; Davis, M.A.; Patrick, H.J.; LeBlanc, M.; Koo, K.P.; Askins, C.G.; Putnam, M.A.; Friebele, E.J. Fiber grating sensors. J. Lightwave Technol. 1997, 15, 1442–1463.

11. Udd, E.; Spillman, W.B. Fiber Optic Sensors: An Introduction for Engineers and Scientists, 2nd ed.; Wiley: Hoboken, NJ, USA, 2011.

12. Jiang, Y.; Ding, W. Recent developments in fiber optic spectral white-light interferometry. Photonic Sens. 2011, 1, 62–71.

13. Bayspec FBGA. Available online: http://www.bayspec.com/telecom-fiber-sensing/fbga-systems/ (accessed on 04 May 2024).

14. FBGS International. FBGG-Scan Interrogators. Available online: https://fbgs.com/components/#measurement-devices (accessed on 04 May 2024).

15. Ibsen Photonics. Interrogation Monitors. Available online: https://ibsen.com/products/interrogation-monitors/ (accessed on 04 May 2024).

16. Kwon, Y.S.; Ko, M.O.; Jung, M.S.; Park, I.G.; Kim, N.; Han, S.P.; Ryu, H.C.; Park, K.H.; Jeon, M.Y. Dynamic sensor interrogation using wavelength-swept laser with a polygon-scanner-based wavelength filter. Sensors 2013, 13, 9669–9678.

17. Yun, S.H.; Richardson, D.J.; Kim, B.Y. Interrogation of fiber grating sensor arrays with a wavelength-swept fiber laser. Opt. Lett. 1998, 23, 843–845.

18. Mohanty, L.; Koh, L.M.; Tjin, S.C. Fiber Bragg grating microphone system. Appl. Phys. Lett. 2006, 89, 161109.

19. Bezombes, F.A.; Lalor, M.J.; Burton, D.R. Contact microphone using optical fibre Bragg grating technology. J. Phys. Conf. Ser. 2007, 76, 012017.

20. Wilson, A.; James, S.W.; Tatam, R.P. Time-division-multiplexed interrogation of fibre Bragg grating sensors using laser diodes. Meas. Sci. Technol. 2001, 12, 181–187.

21. Tosi, D.; Olivero, M.; Perrone, G. Low-cost fiber Bragg grating vibroacoustic sensor for voice and heartbeat detection. Appl. Opt. 2008, 47, 5123–5129.

22. Micron Optics. Interrogators. Available online: https://lunainc.com/ (accessed on 04 May 2024).

23. HBK is available online: https://www.hbm.com/en/2322/optical-interrogators/?product_type_no=Optical%20Interrogator (accessed on 04 May 2024).

24. Fiber Bragg grating. Available online: https://en.wikipedia.org/wiki/Fiber_Bragg_grating (accessed on 04 May 2024)

25. FBGS. Available online: https://fbgs.com/technology/fbg-principle/ (accessed on 04 May 2024).

26. Anees Mohammed, Sinisa Djurovic: A Study of Distributed Embedded Thermal Monitoring in Electric Coils Based on FBG Sensor Multiplexing.Microprocessors and Microsystems(2018)

27. smartfibres. Available online: https://www.smartfibres.com/technology (accessed on 04 May 2024).

28. Taha Ali Alian et al.: Performance Evaluation of Fiber Bragg grating temperature sensor. Lambert Academic Publishing (2014)

29. fiberoptics4sale. Available online: https://www.fiberoptics4sale.com/blogs/archive-posts/95046406-what-is-fiber-bragg-grating (accessed on 04 May 2024).

30. Thorbjoern Buck, Mathias S.Müller, A.W. Koch: Theoretical Assessment of an All-Optical Temporal Low-Pass Filter for Dynamic Fiber Bragg Grating Signals.Hindawi Publishing Corporation Journal of Sensors(2011)

31. Rp-photonics. Available online: https://www.rp-photonics.com/bg/buy_bragg_gratings.html (accessed on 04 May 2024).

32. Emerson. Available online: https://www.emerson.com/documents/automation/brochure-pantograph-controls-aventics-en-6854624.pdf (accessed on 04 May 2024).

33. Wikipedia. Available online: https://en.m.wikipedia.org/wiki/File:HAZMAT_Class_1_Explosives.png (accessed on 04 May 2024).

34. Available online: https://phys.org/news/2017-05-fiber-based-sensor-quickly-problems-bridges.html (accessed on 04 May 2024).

35. laserfocusworld. Available online: https://www.laserfocusworld.com/fiber-optics/article/16560884/fiber-optic-sensing-distributed-fiber-optic-sensing-solves-real-world-problems (accessed on 04 May 2024).