FITNESS FUNCTION DETERMINATION OF UAV ANOMALY DETECTION IN LARGE DATA SET VIA PSO

FATIMAH BINTI DAING JAMIL

BACHELOR OF ELECTRICAL ENGINEERING (ELECTRONICS) WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : FATIMAH BINTI DAING JAMIL

Date of Birth : 11 SEPTEMBER 1996

Title : FITNESS FUNCTION DETERMINATION OF UAV ANOMALY DETECTION IN LARGE DATA SET VIA PSO

Session : SEM I 2021/2022

I declare that this thesis is classified as:

☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*

☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*

☒ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

fatimah	
---------------------------------	---------------------------------
(Student's Signature)	(Supervisor's Signature)
FATIMAH BINTI DAING JAMIL 960911-01-5966	TS. DR. MOHAMMAD FADHIL BIN ABAS
DATE: 7 FEBRUARY 2022	DATE: 7 FEBRUARY 2022

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Electrical Engineering (Electronics) with Honours.

(Supervisor's Signature)

Full Name : TS. DR. MOHAMMAD FADHIL BIN ABAS

Position : SENIOR LECTURER

Date : 7 FEBRUARI 2022

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at University Malaysia Pahang or any other institutions.

fatimah

(Student’s Signature)

Full Name : FATIMAH BINTI DAING JAMIL

ID Number : EA18061

Date : 7 FEBRUARI 2022

FITNESS FUNCTION DETERMINATION OF UAV ANOMALY DETECTION IN LARGE DATA SET VIA PSO

FATIMAH BINTI DAING JAMIL

Thesis submitted in fulfilment of the requirements for the award of the
Bachelor of Electrical Engineering (Electronics) with Honours

College of Engineering
UNIVERSITI MALAYSIA PAHANG

FEBRUARI 2022

ACKNOWLEDGEMENTS

Assalamualaikum w.b.t and greetings. Thanks to God for His grace and permission, I have been able to conduct the study for Engineering Project 2. I am very happy that this project started with good and show the positive progress. It went as smoothly as expected. Special thanks to my supervisor Dr. Mohammad Fadhil Bin Abas for giving me opportunity to conduct this Engineering Project 2 under his guide and for all the advice and guidance, advice and assistance provided. Also to Engineering Project 2 Coordinator for giving a good opportunity for me to encourage me in this project with clear briefing and systematic progress management. I would also like to express my sincere thanks to the lecturers and staff of the University of Malaysia Pahang especially to the Faculty of Electrical and Electronic Engineering who have provided so much guidance in the preparation of this report and provided me with all facilities needed regarding this project. Much appreciation to my family and my fellow comrades who have contributed so much to my moral and encouragement throughout my journey to complete this project. Indeed, I hope this project will become benefit to the others study in the future.

Thank you

ABSTRAK

Projek ini adalah berdasarkan penentuan fungsi kecergasan pengesanan anomali Kenderaan Udara Tanpa Pemandu (UAV) dalam set data yang besar. Fungsi kecergasan adalah penyelesaian kepada isu sebagai input dan output bagaimana "sesuai" atau "cemerlang" jawapan berkenaan dengan masalah yang dibincangkan. Berdasarkan kajian lepas terdapat penggunaan kaedah Pengoptimuman Partikel Swarm (PSO. Dalam projek ini, dengan menggunakan kaedah PSO mentakrifkan kerosakan motor atau bilah dengan mengesannya dengan pecutan, ia adalah ukuran seberapa cepat kelajuan berubah mengikut masa. Ukuran pecutan dinyatakan dalam unit (meter sesaat) sesaat atau meter sesaat kuasa dua (m/s^2). Kaedah PSO bersama dengan pemantauan berasaskan, boleh mengenal pasti di mana sebenarnya kesalahan telah berlaku. Halaju getaran akan meningkat kira-kira dua kali ganda daripada halaju biasa jika kerosakan dikesan. Untuk mengurangkan bahagian kos ujian dan pengesanan kerosakan Unmanned Aerial Vehicle (UAV), data dikumpul dengan menggunakan perisian dalam gelung dengan tiga program seperti mission planner, ardupilot dan flight gear. Melalui simulasi yang telah dilakukan ianya disahkan dengan menggunakan PSO kerosakan yang berlaku pada motor/bilah UAV dapat dikesan dengan nilai keberkesanan sebenar sebanyak 76%.

ABSTRACT

This project is based on fitness function determination of Unmanned Aerial Vehicle (UAV) anomaly detection in large data set. Fitness function is a solution to the issue as input and outputs how "fit" or "excellent" the answer is with regard to the problem under discussion. Based on previous research there are limited used of Particle Swarm Optimization (PSO). In this project, by using the PSO method define the fault of motor or blade by detecting it with acceleration, it is measure of how quickly speed changes with time. The measure of acceleration is expressed in units of (metres per second) per second or metres per second squared (m/s^2). PSO method along with the monitoring based, can identify where exactly the fault has happened. Vibration velocity will be increase about two times from the normal velocity if the fault detected. To reduce the costing part of the Unmanned Aerial Vehicle (UAV) testing and detection of fault, the data is collected by using software in the loop with three program such as mission planner, ardupilot and flight gear. Through the simulation, that has been done it is verified by using PSO the fault occur at the motor/blade of UAV can be detected with a true positive detection rate of 76%.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

An unmanned aerial vehicle (UAV) (or un crewed aerial vehicle, or drone) is a plane that does not have a human pilot on board. Unmanned aerial vehicles (UAVs) are part of an unmanned aircraft system (UAS), which consists of a UAV, a ground-based controller, and a communication system between the two. UAVs can fly with varying degrees of autonomy, such as under remote control by a human operator or autonomously by on board computers known as an autopilot. UAVs were initially used for missions that were too "dull, dirty, or dangerous" for humans, as opposed to crewed aircraft. Although drones were originally designed for military use, they are now being used for a wide range of purposes, including aerial photography, package delivery, agriculture, policing and surveillance, infrastructure inspections, research, smuggling, and drone racing.

Currently, the dataset includes processed data for 47 autonomous flights with 23 sudden full engine failure scenarios and 24 scenarios for seven other types of sudden control surface (actuator) faults, with a total of 66 minutes of flight in normal conditions and 13 minutes of post-fault flight time. It also includes tens of failure situations and many hours of raw data from completely autonomous, autopilot-assisted, and manual flights. The ground truth of the timing and kind of defects is supplied in each scenario to enable assessment of the algorithms utilising the dataset.

Fitness function determination of Unmanned Aerial Vehicle (UAV) anomaly detection in large data set via Particle Swarm Optimization (PSO) is a method for Fault Detection and Identification (FDI) by utilizing a set of rules for its fitness function. Anomaly is an unexpected change within these data patterns, or an event that does not conform to the expected data pattern. The fitness function is formulated to find the anomaly which is used by PSO.

PSO is a computational approach for optimising a problem by iteratively trying to develop a candidate solution in terms of a given quality measure in computational science. It solves a problem by generating a population of candidate solutions, which are referred to as particles, and moving them around in the search space according to a set of rules to simple mathematical formula over the particle's position and velocity. PSO is originally attributed to Kennedy, Eberhart and Shi and was first intended for simulating social behaviour, as a stylized representation of the movement of organisms in a bird flock or fish school.

The algorithm was simplified, and it was discovered to be optimising. Many philosophical issues of PSO and swarm intelligence are discussed in Kennedy and Eberhart's book. An detailed study of PSO applications is done by Poli. Bonyadi and Michalewicz have released a comprehensive survey of theoretical and practical efforts on PSO.

1.2 PROBLEM STATEMENT

Unmanned aerial Vehicle (UAV) is an aircraft without a human pilot on board. Guided autonomously by remote control or both and the carries sensors target designators, offensive ordnance or electronic transmitter. The most common UAV faults include insufficient battery capacity, loss of communication, motors and propellers problems. Among them, the probability of occurring faults in motors and propellers are also other parts when UAV is flying. These fault can be said as anomalies.

These anomalies include blade/motor problem which induce vibration. In order to detect anomaly, large data acquired from UAVs which includes but not limited to roll, yaw pitch and 3 axis acceleration. This proposed can determine UAVs anomaly detection in large data set. Major problem that always happen to the Unmanned Aerial Vehicle (UAV) is motor/blade. The data that has been read with flying UAVs has always have interrupt either with the surrounding or with the UAVs itself. The large data collected are needed in order to create the function that can detect the fault in UAVs.

Based on the literature, minimum optimization problem has not been researched on for FDI especially in monitoring mode.

1.3 OBJECTIVE

The objective for this project are:

i. To detect Unmanned Aerial Vehicle (UAV) anomaly.
ii. To collect the large set of data by using Particle Swarm Optimization (PSO) with reducing the time taken.
iii. To reduce the costing part of the Unmanned Aerial Vehicle (UAV) testing and detection of fault.

1.4 SCOPE OF PROJECT

The scope of this project is as follows:

i. Detected the anomaly of the Unmanned Aerial Vehicle (UAV).
ii. Using the Particle Swarm Optimization (PSO) testing method to test the fault in every data comes from motor/blade at the Unmanned Aerial Vehicle (UAV).
iii. Data collected by using software in the loop with three program such as mission planner, ardupilot and flight gear.
iv. Monitoring and detecting also testing the fitness function of UAVs anomaly in large data set.
v. Collect data of the fault at the UAVs by using Fault Detection and Isolation (FDI) method.
vi. Detected the vibration of the actual fault, not the vibration that produced by turning point.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss literature studies regarding fault in Unmanned Aerial Vehicle (UAV). The discussion including the element involved in the project that can be accessed through various research paper. All of the component in the topic are referred differently in order to obtain more clear understanding regarding this study. The main part of the topic itself is the fault at the motor/blade function. The literature review is going through the project process that already being conducted by other resources which can be adapted into this project. These resource include journals and materials obtained from the internet.Fault detection, isolation, and recovery (FDIR) is a subfield of control engineering which concerns itself with monitoring a system, identifying when a fault has occurred, and pinpointing the type of fault and its location. Two approaches can be distinguished: A direct pattern identification of sensor readings indicating a problem, as well as an examination of the disparity between the sensor readings and predicted values generated from some model. In the latter situation, a problem is considered to be discovered if the disparity or residual exceeds a specified threshold. The goal of fault isolation is then to classify the type of defect and its location in the equipment.

Fault detection and isolation (FDI) approaches are typically divided into two types. Model-based FDI and signal processing-based FDI are examples of these. In model-based FDI approaches, a system model is employed to determine the incidence of a problem. The system model might be mathematical or based on expertise. Observer-based approaches, parity-space approaches, and parameter identification-based methods are examples of model-based FDI procedures. Another type of model-based FDI methodology is known as set-membership methods. Under specific situations, these approaches ensure the discovery of faults. The fundamental distinction is that, rather than identifying the most likely model, these strategies exclude hypotheses that are incompatible with the facts. Some mathematical or statistical procedures are done on the measurements in signal processing-based FDI, or a neural network is trained using measurements to extract information about the defect.

2.2 Literature Overview

The usage of the k-chart is to show the full overview of the study regarding the project. The idea was initiated from review of the research in the literature review. Most of the topic is a using fitness function for determination of Unmanned Aerial Vehicle (UAV) anomaly detection. To detect the anomaly in UAV can use many ways or many methods that already exist such as Faderated Kalman Filter (FKF), Extend Kalman Filter (EKF), Chi-square Test, Double redundancy Test, Proportional and Multiple Integral (PMI), Particle Swarm Optimization (PSO) and GA-BP Neural Network. Over several techniques existed, PSO is the least method have been used in the past 5 years, even though the accuracy of PSO method are more highest than others method. Because of that we wanted to focus on PSO method. This step is important, because the effectiveness of data will be affected by choosing the right testing method.

Then, after defining which technique are proper to used, that technique will be classified into certain based. From the past 5 years of journal there's only two based have been used it is either monitoring based or observer based. Monitoring is more to detect which part have fault, what time, where exactly it is happened and how many time its happen. While, the observer based literally same as monitoring to detect which part have fault but it also came with why that fault is happen. As we know, PSO testing method are detail by taking monitoring based is suitable and easy for PSO to detect fault. Hence, monitoring based is being selected and will be more focus on it.

There have two ways of acquiring the data to be test. First is by using hardware and software together and second is by using software only. By looking at the previous research most of them are using combination method which is use hardware and software that will be add on the cost to do the project. Hence, this project is chosen as it is the best method that can be used to make the objective achieve. Reduce the cost are the important thing in this research, so taken data by using software only being selected as it more easy and can get the accurate data on it.

Several faulty has been define at the UAV, with major problem Motor or Blade/Propeller. Under this two major problem, there are more sub problem such as under the Motor or Blade/Propeller has faulty sensor, steering gear and actuator failure or robustness. By take a close look on previous research, they never take a Motor or Blade/Propeller fault data by using software only. Hence, we choose to take and test the fault data of Motor or Blade/Propeller by using software only. The K-Chart has been show in **Figure 2.1**.

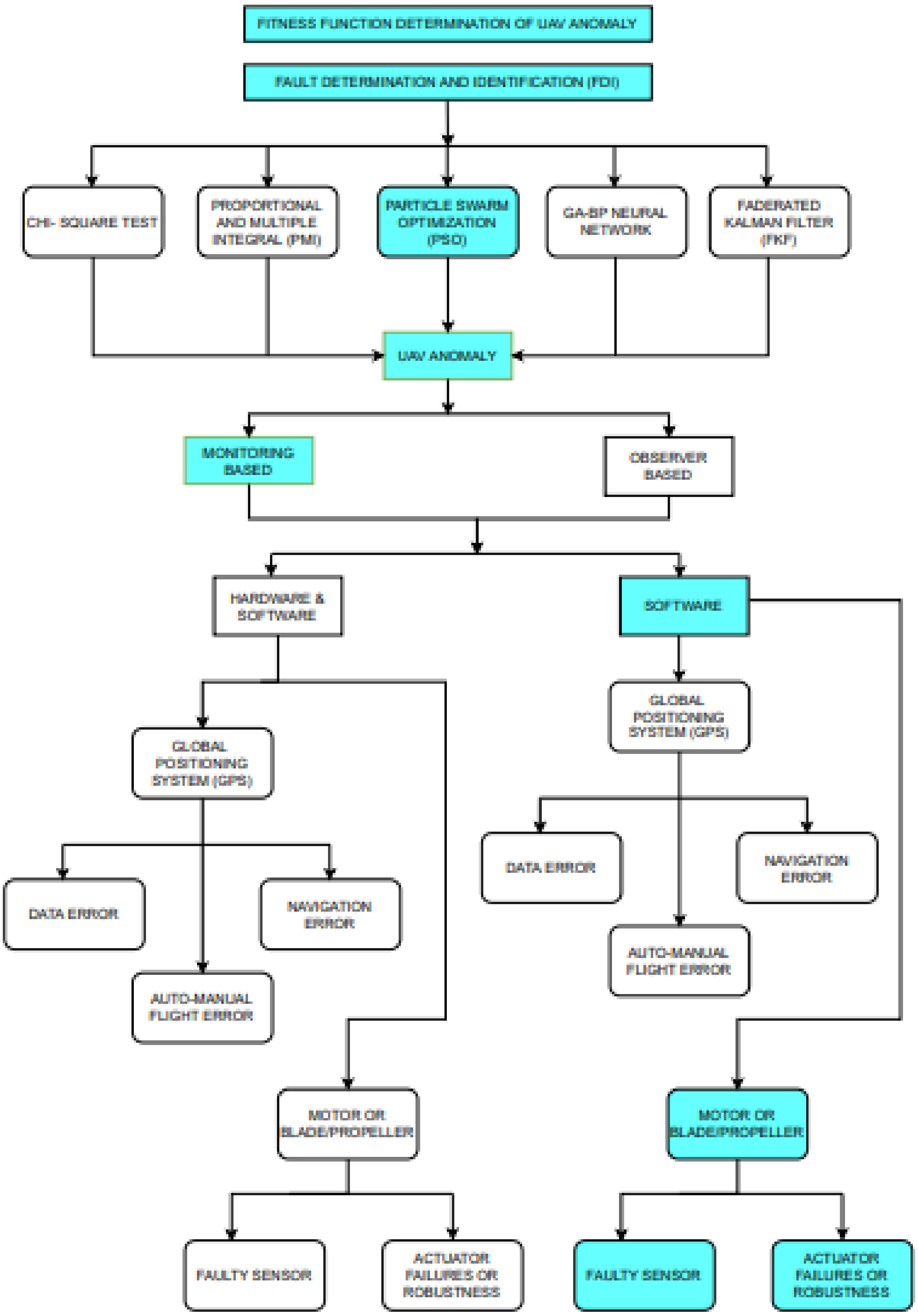


Figure 2.1 K-Chart of the project

2.3 ALFA: A Dataset for UAV Fault and Anomaly Detection

From this paper, it has proposed a dataset of numerous defect types in the Carbon-Z T-28 fixed-wing UAV platform equipped with an on board computer and additional modules for the dataset collection as shown in **Figure 2.2**. The collection contains processed data from 47 autonomous flights with 23 sudden full engine failure scenarios and 24 scenarios for seven additional types of abrupt control surface (actuator) faults, totalling 66 minutes of normal flight time and 13 minutes of post-fault flying time [1]. It also includes tens of failure situations and many hours of raw data from completely autonomous, autopilot-assisted, and manual flights. In each scenario, the ground truth of the timing and kind of defects is supplied to allow evaluation of the algorithms using the dataset [1]. Also have a supplied helpful tools in numerous programming languages to load and manipulate the data, as well as to aid in the assessment of a detection technique utilising the dataset. A collection of measures is proposed to aid in the comparison of various approaches utilising the dataset. The majority of existing fault detection systems are assessed in simulation, and as far this dataset is the only one that provides real flight data with defects in this capacity. The expectation is it will contribute to furthering the state-of-the-art in Anomaly Detection or FDI research for Autonomous Aerial Vehicles and mobile robots in order to improve the safety of autonomous and remote flying operations [1].

The Air Lab Fault and Anomaly (ALFA) Dataset is presented in this paper, which currently includes processed data for 47 autonomous flights with scenarios for eight different types of sudden control surface faults, including engine, rudder, aileron(s), and elevator faults, with 23 of the scenarios focusing on full engine failures. The processed data includes 66 minutes of normal flight time and 13 minutes of post-fault flight time. Several hours of raw autonomous, autopilot-assisted, and manual flight data with tens of various failure scenarios are also included in the collection. To aid in the evaluation of the new methodologies, the processed data offers the ground truth for the exact time and kind of defect in each case. Keipour et al. (2019) [2] utilised a tiny fraction of this dataset to evaluate a real-time anomaly detection algorithm.

Figure 2.2 The Carbon-Z T-28 fixed-wing UAV platform equipped with an on board computer and additional modules for our dataset collection [1]

2.4 Anomaly Detection and Condition Monitoring of UAV Motors and Propellers

For unmanned aerial vehicles, early identification of faulty components is critical (UAVs). The purpose of this study is to create a monitoring system that can detect potential defects in UAV motors and propellers early on. The motor current signature analysis (MCSA) method is used to examine stator current signals under various situations. Then, an unsupervised learning strategy called fuzzy adaptive resonance (Fuzzy ART) neural network (NN) is used to determine if the motors are running normally or not. In addition, the UAV propellers are monitored using the vibration signature analysis (VSA) approach. To decrease computing time, a Q-learning-based Fuzzy ARTMAP NN is used to learn extracted statistical features, and the Genetic algorithm (GA) is utilised to choose an optimal subset of features off-line. The experimental findings supported the proposed model's efficacy in identifying flaws in UAV motors and propellers when compared to CART, KNN, NB, and SVM [3]. Because of their adaptability, high mobility, simple construction, vertical take-off and landing, low cost, and ease of maintenance, unmanned aerial vehicles (UAVs) are rapidly being employed in a variety of applications [4]. Inadequate battery capacity, lack of communication, and motor and propeller issues are the most prevalent UAV failures [5].

2.4.1 The Condition Monitoring System

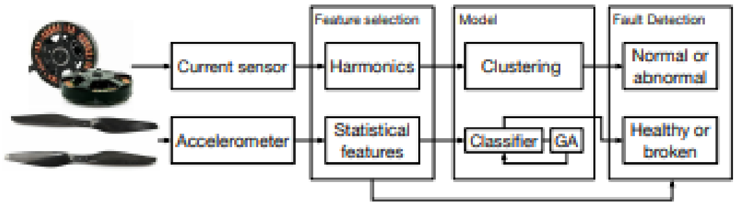


Figure 2.3 Fault detection system diagram [5]

This section outlines the proposed method for monitoring the status of UAV motors and propellers. It is made up of two pieces, as depicted in **Figure 2.3** motors and propellers. To begin, three current sensors (one for each phase of a three-phase motor) and a three-axis accelerometer are used to acquire current and vibration measurements for each UAV motor and propeller [5]. These sensors are linked to an Arduino as a data collecting device, and the signals collected are saved in the computer through a network connection. Fast Fourier transform (FFT) is used to extract the first, third, fifth, seventh, and ninth harmonics of observed current signals. Simultaneously, nine times domain statistical characteristics are derived from those vibration signals, as reported in [6]. The fuzzy ART NN [7] (**Figure 2.4**) is then used to classify extracted harmonics into a number of clusters, as shown below: To begin, fuzzy ART calculates the similarity degree of the current input's complemented-coded.

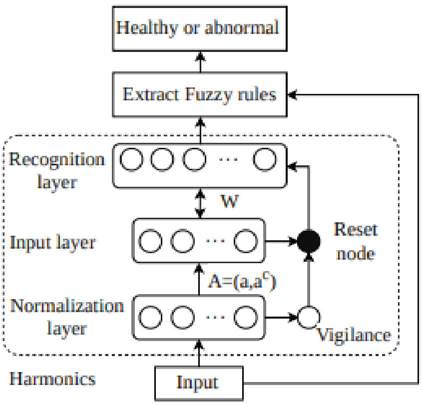


Figure 2.4 The proposed UAV motors monitoring and fault detection system [7]

2.5 Fault detection of a VTOL UAV using Acceleration Measurements

Based on acceleration signals generated by a high-rate inertial measurement unit, this work offers an actuator failure detection method for vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV). The suggested approach is based on time and frequency-domain analysis of acceleration signals, as well as feature selection techniques, to determine the most significant frequencies of the spectrum and time-domain characteristics involved in the fault's occurrence. The algorithm is being tested for damage to a propeller blade and the impact of such damage during vertical take-off and landing, where the propellers create all of the lift [8]. The technique is validated using real-world data collected by the Songbird VTOL UAV [9]. The experimental results show that acceleration signals may be used for diagnostic reasons on VTOL UAVs [8].

Unmanned aerial vehicles are quickly becoming a valuable tool in a variety of applications, including search and rescue, surveillance, inspection, and precision farming. The integration of unmanned aerial vehicles (UAVs) in urban situations necessitates an increase in dependability and fault prediction capabilities, particularly when these unmanned vehicles must be approved to operate in inhabited regions. UAV components, such as sensors and actuators, will undoubtedly fail at some point. When this situation emerges, the error should be discovered as soon as possible so that relevant steps may be taken. Methodologies for defect detection are classified into three types: model-based, signal-based, and knowledge-based [9].

2.5.1 The Songbird VTOL UAV

The Songbird UAV, seen in **Figure 2.5**, is a VTOL (Vertical Take-Off and Landing) UAV that combines the benefits of a typical fixed-wing aircraft with the versatility of a multi-rotor UAV. The major benefit of this type of vehicle is the ability to take off and land practically anyplace, as well as the extended flying time and distance. Because of its vertical take-off and landing capabilities, no runway is required for take-off or landing while retaining all of the benefits of a fixed-wing aircraft. The Songbird is outfitted with a sophisticated autopilot that can manage the UAV during take-off/landing, forward flight, and the transition from quadrotor to fixed-wing mode. We define quadrotor mode as the flight mode in which the four motors are vertical and the four propellers provide all of the lift, and fixed-wing mode as the flight mode in which the motors are inclined. In fixed-wing mode, the motor axes are parallel to the UAV's longitudinal axis, and the wings produce the majority of the lift. **Table 1** [9] summarises the major technical features of the Songbird VTOL UAV.

Table 1 Songbird Technical Specifications [9]

Wing Span:	3.10 m
Maximum Payload:	2 kg
Max Take-off Weight:	10 kg
Working Speed:	Approx. 16 – 18 m/s
Max Speed:	45 m/s (160 km/h)
Flight time:	> 60 minutes (standard)

Figure 2.5 German drones Songbird VTOL UAV [9]

2.6 Fault Diagnosis and Condition Monitoring of UAV Rotor using Signal Processing

A technique for detecting physical deterioration of UAV rotor blades is provided in this study. Actuators in multirotor UAV (Unmanned Aerial Vehicle) systems are typical subjects for fault detection methods, which are an integral component of an active fault-tolerant control scheme. Defects in the aerial vehicle's propulsion system cause rotors to lose thrust, causing thrust balance to be disrupted, greater power consumption, and additional deterioration, perhaps leading to the vehicle's collapse. In this research, they have proposed a three-stage approach based on signal processing and machine learning to identify rotor faults and assess their magnitude and nature. As imbalanced rotating components typically produce vibrations in mechanical systems [10], the approach is based on acceleration readings from the on board IMU (Inertial Measurement Unit) sensor.

The acceleration signal is recorded in a cyclic buffer and then analysed by basic feature extraction methods to generate a broken condition signature. This article examines three distinct techniques of feature extraction, as well as the effects of varied buffer length. The Support Vector Machine (SVM) classifier is then utilised to determine the presence and nature of the rotor malfunction. The provided solution was validated by a number of tests that demonstrated its efficacy. Furthermore, such a signal processing-based technique is exceedingly adaptable and simple to implement in any arbitrary flight controller [10].

Concerns about the safety of unmanned aerial vehicles (UAVs) are growing in tandem with their increasing acceptance in military and civil applications [11]. As a result, the subject of how to offer safe flight control algorithms for unmanned aerial vehicles emerges. Recent hurdles in UAV development include, among other things, the issue of fault-tolerant control (FTC). FTC approaches are designed to maintain the system's minimum necessary performance and to keep it operational in the event of hardware or software component failure [12].

2.6.1 Fault Detection Method

A. General concept

In this research, has shown in **Figure 2.6** they proposed a three-stage approach for defect identification in UAV propulsion. Because accelerometer data is a fundamental component of state estimation techniques in multirotor UAVs [13], it is appropriate for use in problem identification and diagnostics. As a result, the initial step in the suggested technique is to store measurements of accelerations collected during flight in the cyclic buffer. The data in the buffer is then analysed by the features extraction method to get the vibration signal's distinctive signature.

The findings of several signal analysis methods are reported in this study. For this goal, three well-known and straightforward methods were chosen: The Fast Fourier Transform (FFT), Wavelet Packet Decomposition (WPD), and measuring signal power in linearly spaced frequency bands (BP, Band Power) [14]. In the following stage, features are categorised using the Support Vector Machine (SVM) to identify the incidence, scale, and type of the defect.

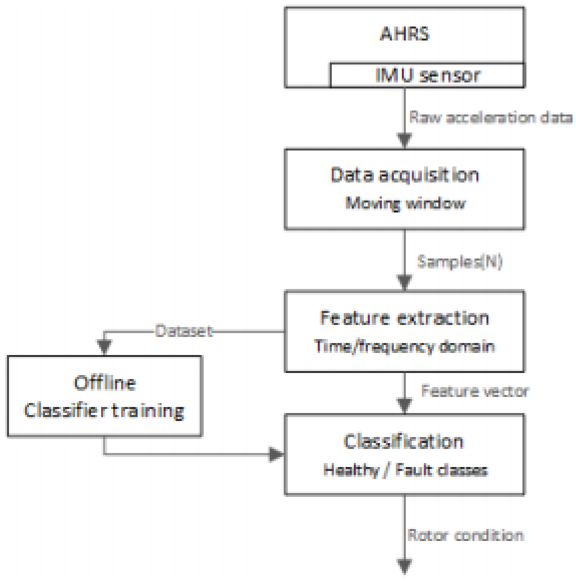


Figure 2.6 General concept of the fault detection method [14]

B. Data acquisition

The initial stage is to acquire raw (unfiltered) acceleration data in two axes of a plane parallel to the rotor discs. The data on the vertical axis was deleted since no meaningful information about vibrations was found there. The signal is first pre-processed with the standard Hanning window. The length of the buffer is a characteristic that influences the speed and accuracy of fault detection. It is regarded as a parameter for experimental assessment, as discussed in the next section [13] [14].

They believe that a lengthier data packet of acceleration signal may carry more information regarding the fault's distinctive signature. As a result, extended data gathering may enhance problem detection performance while increasing signal processing time [14].

C. Extraction of signal features

To fine-tune and evaluate the algorithm's performance, three distinct approaches of feature extraction from single-dimensional signals were applied. The first technique uses the traditional Fast Fourier transform (FFT) algorithm to generate a single-sided amplitude spectrum. To match the length of the feature vector, the spectrum is shortened.

The second method employs discrete signal decomposition into wavelet packets. In the given technique, a three-level wavelet packet decomposition (WPD) tree was utilised, and the feature vector consists of standard deviations of wavelet components at the lowest level of the WPD tree [13]. A series of trials resulted in the selection of the db3 wavelet, which provided the greatest performance in the end result.

The third technique relies on calculating average signal power across frequency bands. The signal is filtered into a number of linearly spaced sub bands, and the computed signal strength in each band is used to construct the vector of features [13].

D. Fault diagnostic

The third stage of the presented technique is based on machine learning and employs the Support Vector Machine with the Gaussian kernel to categorise signals and identify the rotor's state. The SVM was chosen because of its great speed and adaptability in basic feature classification tasks. Because the SVM is a binary classifier, the one vs. all technique was chosen to differentiate fault types.

The entire method is made up of three distinct classifiers [14]. First, the fault occurrence (healthy/damaged rotor class) is identified. The second SVM then determines the severity of the injury (light/severe). Finally, the final classifier predicts the kind of defect (damaged edge / deformed propeller blade tip) [14].

2.9 Summary

From the several journal or research above, all of them mostly at the monitoring based under fault of Motor/Blade. By taking the closed look, for the Motor/Blade are mostly fault in propellers and actuator. This several research can determine which problem need to be solved and improved. The first obvious thing, is the cost of testing and the second is the time taken for testing also there are several method and several way of testing but for the past 5 years, none of them using the Particle Swarm Optimization (PSO) testing method to test any fault on Motor/Blade. The interesting things is, most of the journal are using the Fault Detection and Isolation algorithm method to collect data and to solve it.

The accuracy of data, for the past 5 years are not over than 93% [1][2]. To improves, this several gap the proposed and the objective of this paper will try to solved and achieved the target which is reducing the time taken of data and reducing cost to test and get the exact accurate data. With all the observation to every journal, Fault Detection and Isolation (FDI) method with the Anomaly Detection (AD) method are the effective way to collect data but the way of testing method still on the progress to get the right one.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describe on the methodology used during the study of the project. It provides information concerning the method that was used in undertaking this research as well as a justification for the use of this method. The Chapter also describes the various stages of the research, which includes the data collection process and the process of simulation. This project explored the challenges faced by simulation. The project will go through a 2 phase in total which is collect the Unmanned Aerial Vehicle (UAV) data and test the fitness function on it. This is in order to obtain the full view on how the project will be conducted. The phases are being discussed in this chapter.

3.2 Flowchart

The project starts with develop the coding. It has two coding that have to develop, first is Fault Detection and Identification (FDI) which is the coding for detect the problem and that will be the main objective that we must achieve in this project. Second step is developing Particle Swarm Optimization (PSO) coding and will be the Fitness Function, the testing technique that have been chosen because of the accuracy to collect large dataset which is PSO can do it with better performance [15]. Then after develop that two major coding, we proceed to choose Motor/Blade data and from that import any of that two data into the MATLAB to be test.

Begin the monitoring based which is the fitness function part, we have to create the formula first and make sure the formula is match with the based. After the formula has been created, start to add the frame. This frame to make the program easy to detect the fault, the number of frame depends on how accurate we want the PSO to detect the fault data. Set the frame to be 100 data each and using only two frame on this monitoring based. For the first try, we test with the fault data which is import the Motor/Blade data to MATLAB.

Next, we come out with a standard deviation method combine with tolerance calculation where we take the different between normal peak data with fault peak data and make it as a range which is the peak are chosen between fault peak that do not same level with the turning point or the normal data. Import the fault data to MATLAB run the PSO coding together with fitness function and to make sure it has a consistent or has the accurate detection we should do multiple run with different iteration. After detect the fault we will proceed to next step which is the last step of this project which is record every testing put into chaos matrix to monitor the percent of accuracy. Hence, the objective achieved.

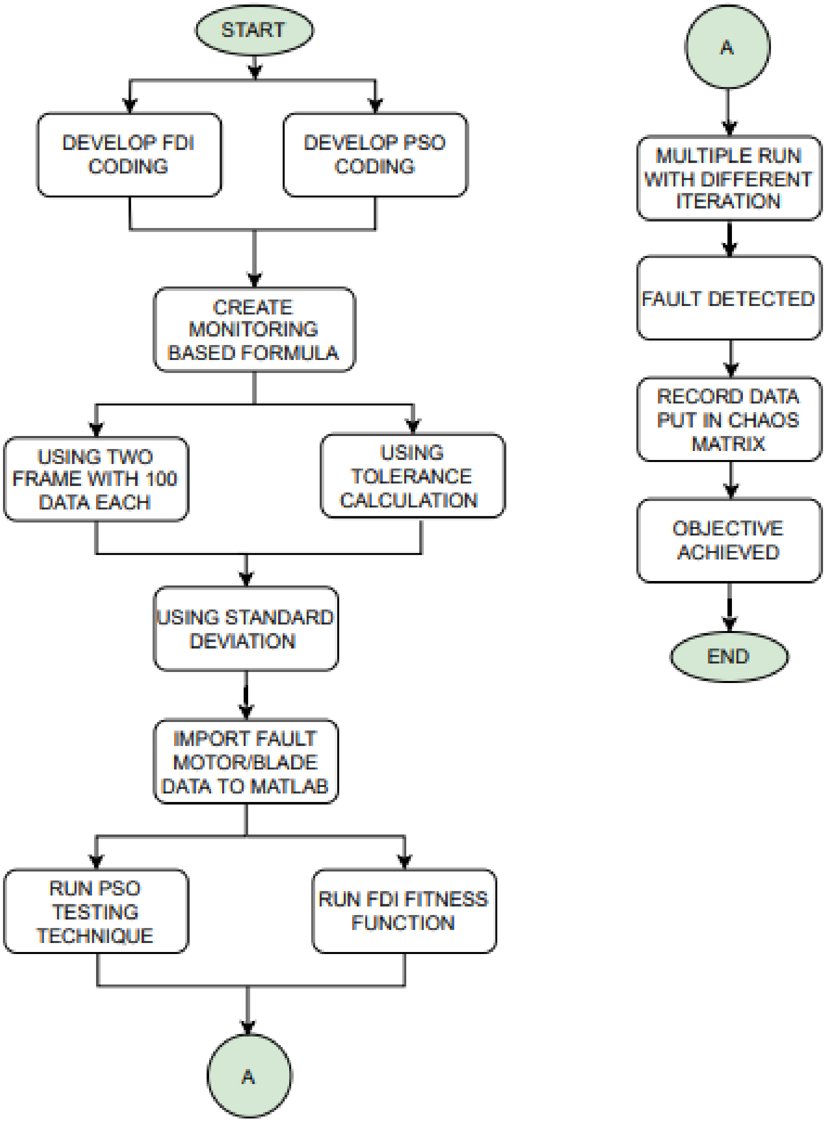


Figure 3.1 Flowchart

3.2.1 Fault Detection and Isolation

The primary purpose of FDI is to detect problems and reliably isolate them to a failing component in the lowest amount of time feasible. This functionality reduces diagnostic time or downtime in general, resulting in better system availability.

3.2.2 Particle Swarm Optimization

PSO is out of another testing method that can be used to test the fitness of data. PSO can test in large data set without hesitation, some of testing method only can collect a several data. The key benefits of the PSO algorithm are described as follows: simple principle, simple implementation, control parameter robustness, and computational efficiency when compared to other heuristic optimization approaches and mathematical algorithms.

3.3 Basic PSO Working

The PSO algorithm can be described as below [16]:

$$\boldsymbol{v_{ij}^{k} = w_{ij}^{k-1} + c_1 r_1\left(pb_{ij}^{k-1} - x_{ij}^{k-1}\right) + c_2 r_2\left(gb_{j}^{k-1} - x_{ij}^{k-1}\right)} \qquad (1)$$

$$\boldsymbol{x_{ij}^{k} = x_{ij}^{k-1} + v_{ij}^{k}} \qquad (2)$$

where 1 c and 2 c are positive constants, called acceleration constants (1 c is the self-confidence (cognitive) factor and 2 c is the swarm confidence (social) factor), usually 1 c and 2 c are in the range from 1.5 to 2.5, and 1 r and 2 r are two random functions uniformly distributed in the range [0, 1]. w is the inertia weight factor that takes linearly decreasing values downward from 1 to 0 [17]. The size of swarm population is ascertained by i = 1, 2, …, N and j = 1,2, …, D, where N is the size of swarm population and k = ,1 2,..., determines the iteration number. Equations (1) and (2) describe the flight trajectory of a population of particles. The velocity is dynamically updated as described by Eq. (1) and the position update of the flying particles is determined by Eq. (2). The effect of the particle inertia, the particle memory influence, and the swarm (society) influence represent the 1st term, 2nd term, and 3rd term in Eq. (1), respectively [16]. The flowchart of the procedure is shown in **Figure 3.2**.

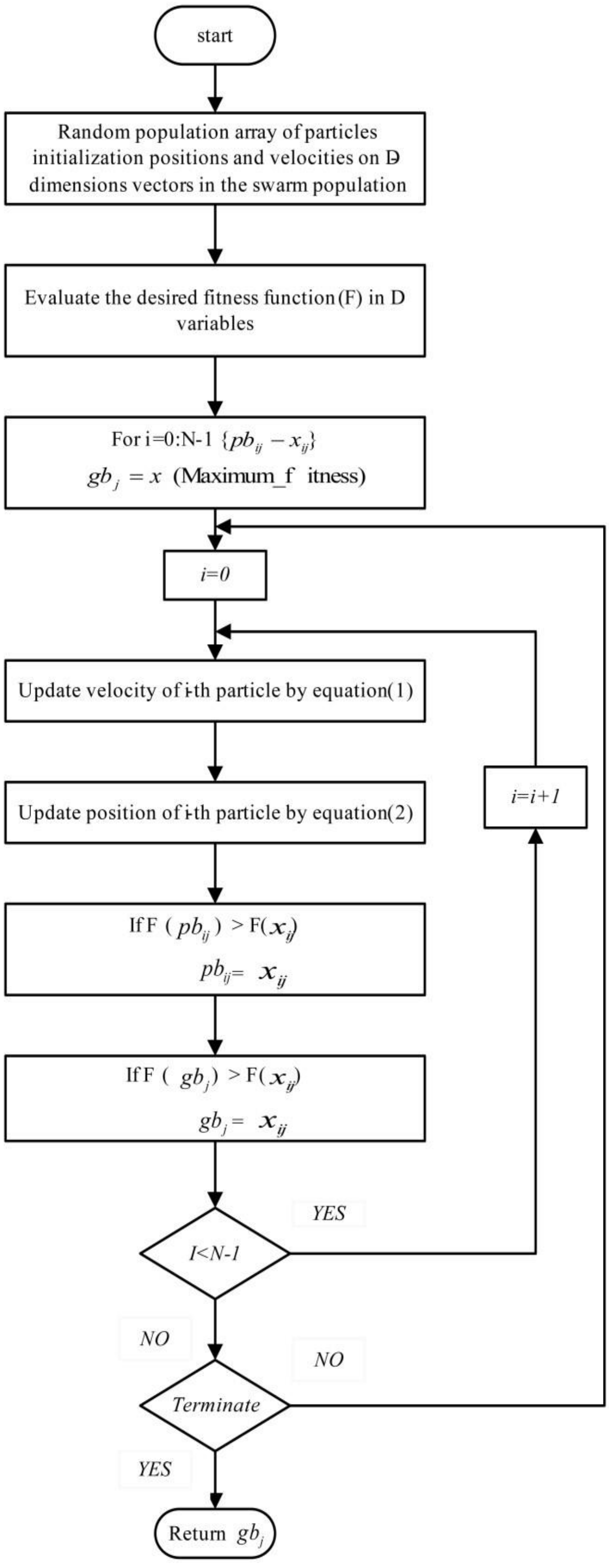


Figure 3.2 PSO algorithm flowchart [16]

PSO with a local neighbourhood and PSO with a global neighbourhood are developed as two variants of the PSO algorithm. Each particle is updated by following two (best) values during each iteration process. Each particle (solution) moves towards its best previous position and towards the best particle in the whole swarm according to the global neighbourhood, This best position is set as the current global best and is called gbest. On the other hand, the position vector of the best solution (fitness) this particle can be achieved (called lbest) when each particle (solution) moves towards its previous best position and towards the best particle in its restricted local neighbourhood [16].

3.4 Vibration Fault Value in UAV

The detection of fault value in Unmanned Aerial Vehicle (UAV) is depends on value of accelerometer. In this part, we take the fault of Motor/Blade on UAV and test the vibration data. By doing some research, if a motor or a blade are having a problem the vibration velocity will increase about two times from the normal value.

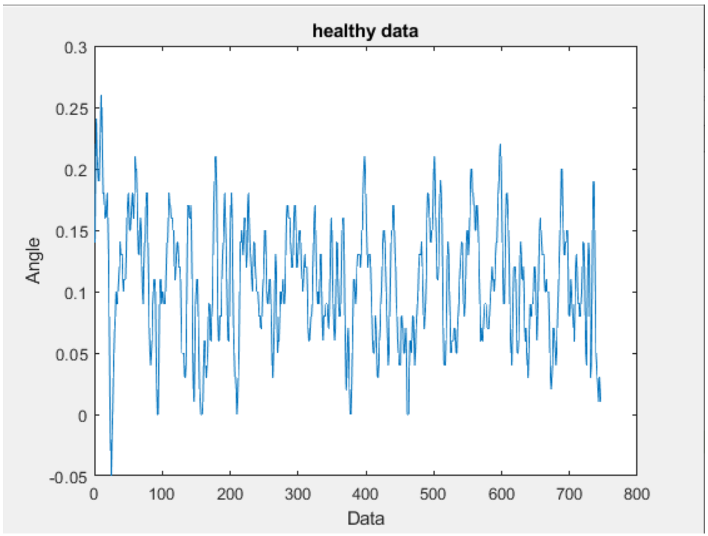


Figure 3.3 Data without fault

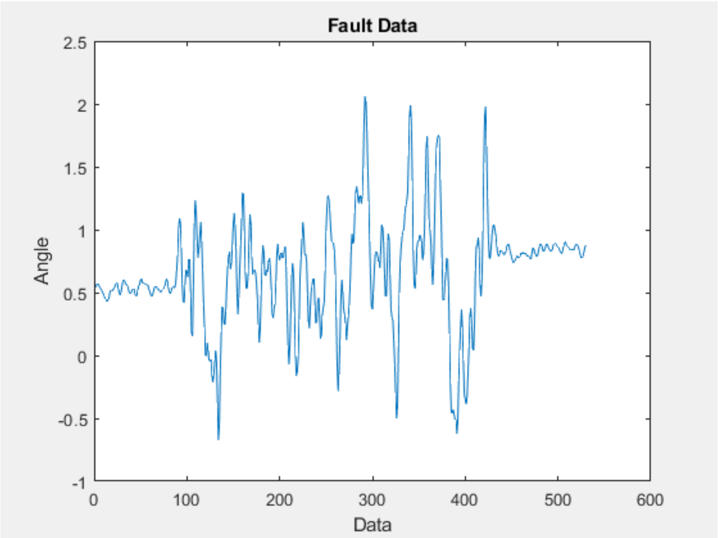


Figure 3.4 Data with fault

This part is to test either vibration is really can be used to detect a fault or not. From the figure above it is proven that if the fault been detected, the value of vibration velocity will increase two times from normal.

3.5 Develop Monitoring Coding

Monitoring coding develop with four stages which is frame data, different slope check, tolerance calculation and last standard deviation (std).

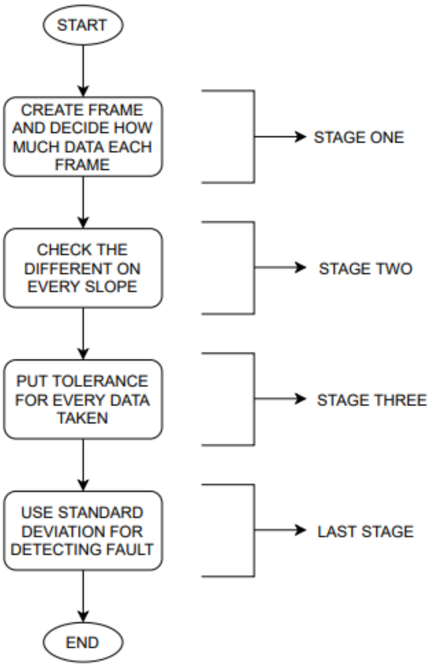


Figure 3.5 Four stages of monitoring based

3.6 Standard Deviation Method

We use standard deviation method where we have to find a different between the fault peak which is the range that still in the fault value.

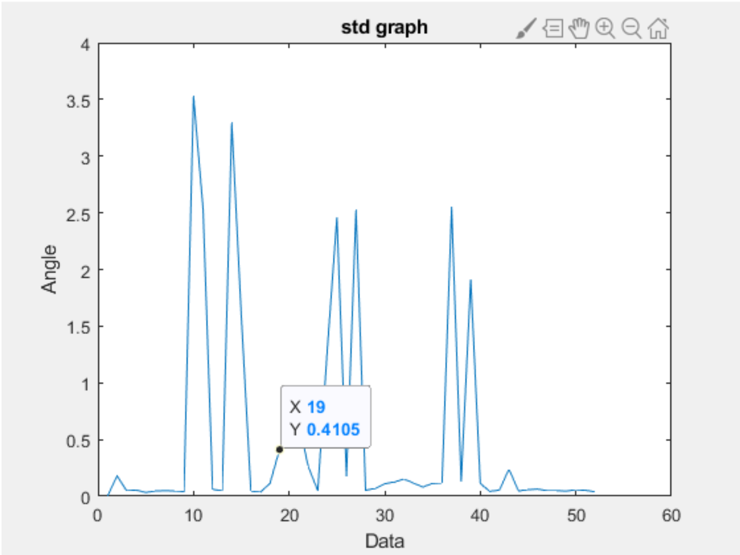


Figure 3.6 Minimum range of std

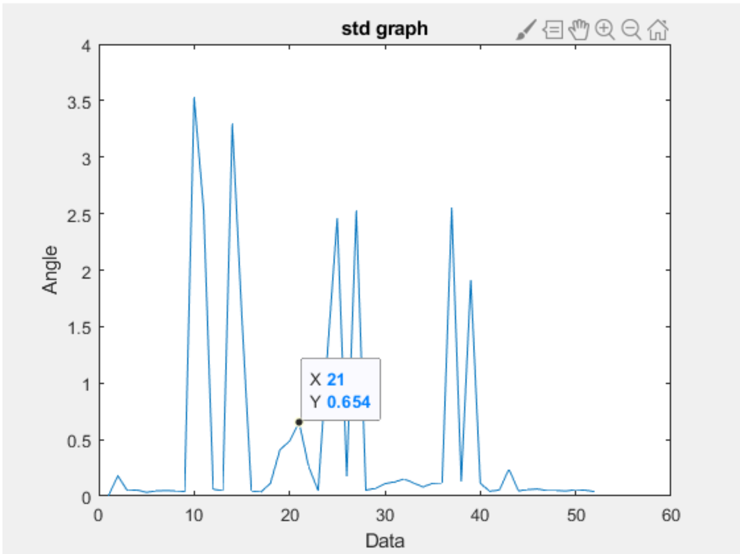


Figure 3.7 Maximum range std

As shown in **Figure 3.6** and **Figure 3.7** there are the range of the fault data. By detecting the lowest fault data which is the lowest peak that not same level with the turning point or the normal data and same with the highest fault data which is literally not same level with the highest turning peak. With this value we decided the best range of the standard deviation value.

```
Emma=datafile(x+8)-datafile(x+2)/6;
Emmal=datafile(x+18)-datafile(x+12)/6;
            if (Emmal<Emma +0.000002) & (Emmal>Emma -0.000002) % tolerance
                Y=100;
            else
                if (std(Emal)>0.5&&std(Emal)<0.7)

                 Y=100-Y;

                else
                    Y=100;
                end
            end
```

Figure 3.8 Standard deviation at OPT

The best range standard deviation (std) to get the accurate data is in between as it set in OPT that have shown in **Figure 3.8.** The combination of standard deviation with other stages make the monitoring based more stable to detect the fault data by using Particle Swarm Optimization (PSO) testing technique. The full code can be seen in **APPENDIX A** for main, **APPENDIX B** for PSO, **APPENDIX C** for Fitness Function and **APPENDIX D** for finding value of standard deviation.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter will briefly explain the preliminary result from the develop coding that have been studied. Which is by using Fault Detection and Identification (FDI) with Particle Swarm Optimization (PSO) testing technique combine with monitoring based method can determine the Unmanned Aerial Vehicle (UAV) anomaly in large data set and the expected outcome were being discussed.

4.2 Accelerometer Value Testing

For this testing, we start put the value from the lowest to the highest and testing with multiple run at MATLAB with different iteration because sometimes number of maximum iteration also give an effect to the accuracy of testing data. As mention above that, the value of fault Blade/Motor accelerometer must two times bigger from the normal value. Hence, we test from $2m/s^2$ until $9m/s^2$.

1. The first data is injected with $2m/s^2$ vibration. The fault was at 1321 but using Fitness Function the fault is detected at 552.

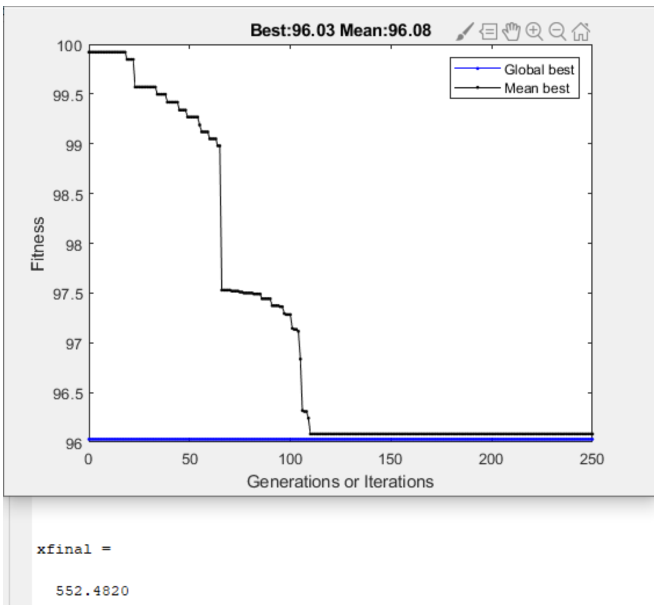


Figure 4.1 Error on 1321

2. The second data is injected with $4 m/s^2$ vibration. The fault was at 1168 but using Fitness Function the fault is detected at 3601.

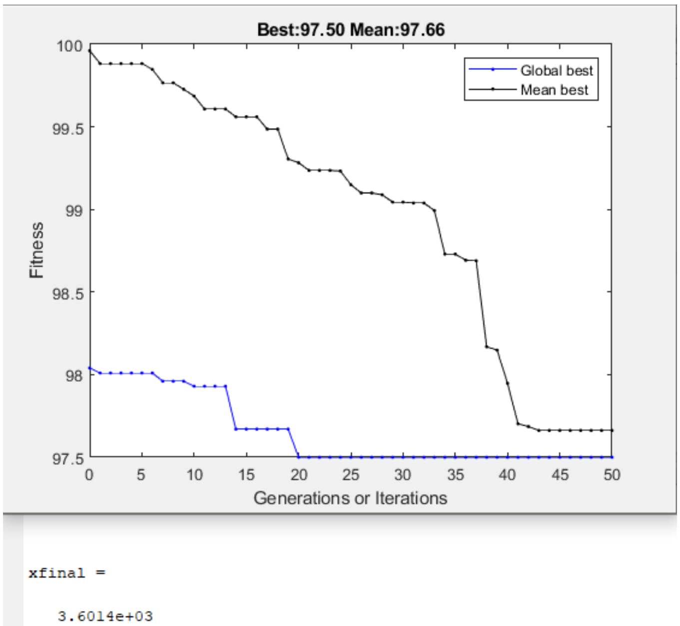


Figure 4.2 Error on 1168

3. The third data is injected with $5 m/s^2$ vibration. The fault was at 1281 but using Fitness Function the fault is detected at 1137.

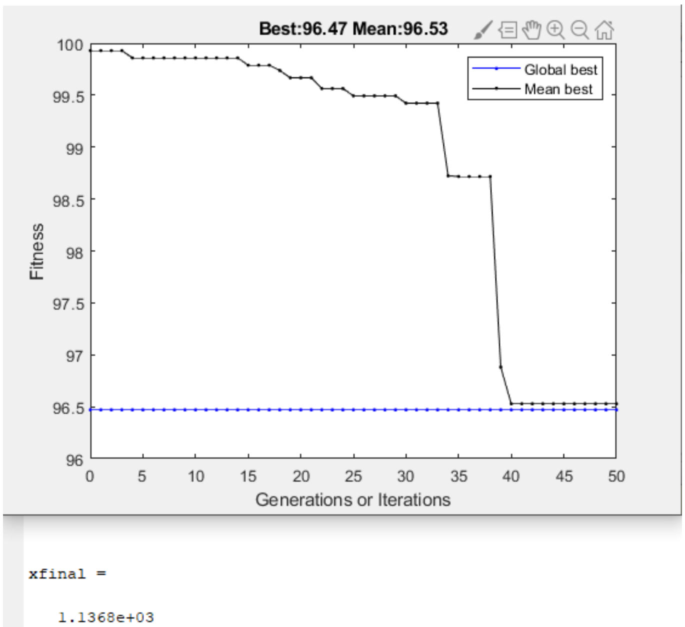


Figure 4.3 Error on 1281

4. The fourth data is injected with 6m/s^2 vibration. The fault was at 1263 but using Fitness Function the fault is detected at 1139.

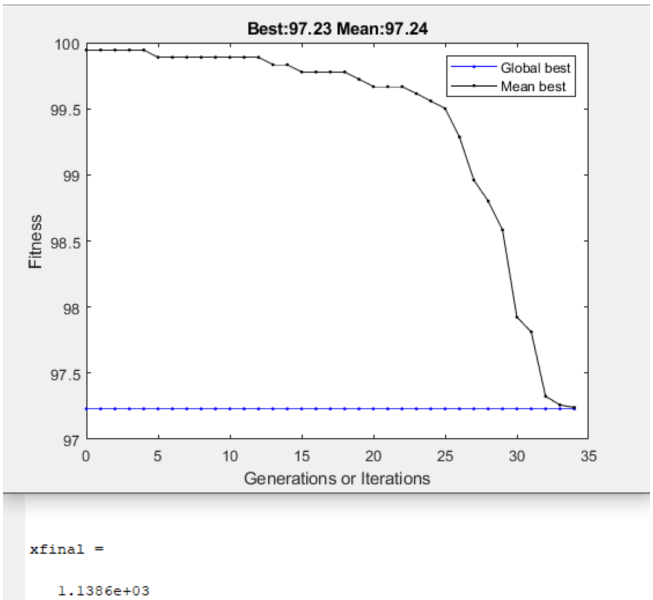


Figure 4.4 Error on 1263

5. The fifth data is injected with 7m/s^2 vibration. The fault was at 1201 but using Fitness Function the fault is detected at 1047.

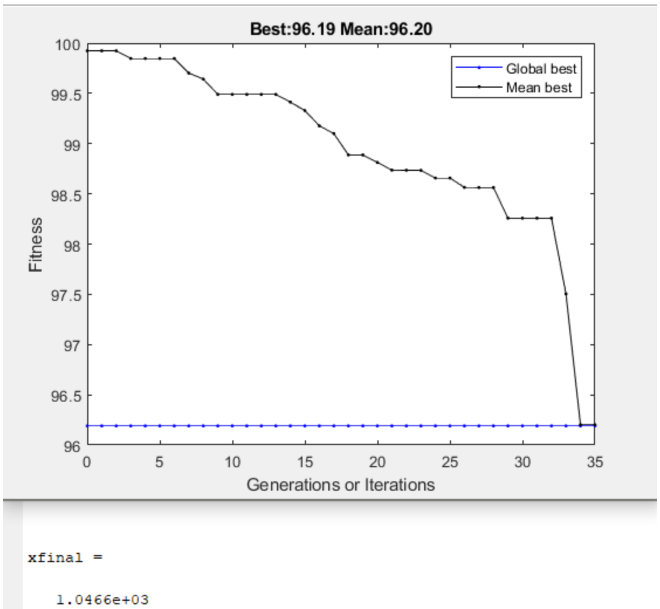


Figure 4.5 Error on 1201

6. The sixth data is injected with 8m/s^2 vibration. The fault was at 1180 but using Fitness Function the fault is detected at 1004.

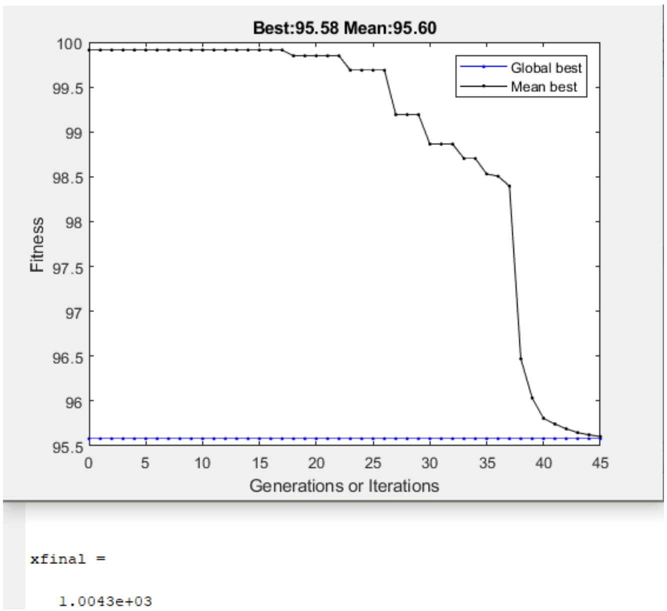


Figure 4.6 Error 0n 1180

7. The seventh data is injected with 9m/s^2 vibration. The fault was at 1142 but using Fitness Function the fault is detected at 717.

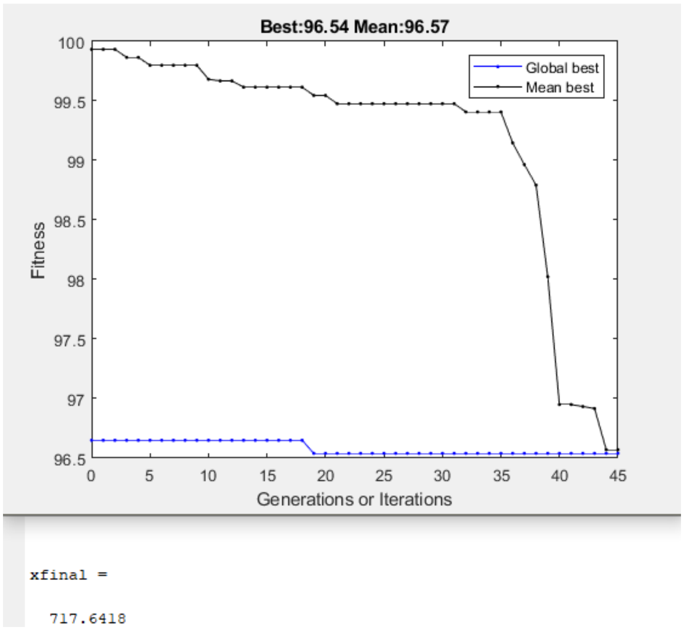


Figure 4.7 Error on 1142

Table 2 Accelerometer Value Testing

m/s^2	Actual	Detect
2	Fault at 1321	552
4	Fault at 1168	3601
5	Fault at 1281	1137
6	Fault at 1263	1139
7	Fault at 1201	1047
8	Fault at 1180	1004
9	Fault at 1142	717

From all the seventh accelerometer that has been test, we have to choose the best value that can make the percent of accuracy of detected the fault is increase. Hence, the best value is $7m/s^2$ and $8m/s^2$ and each of the value must be add with the frame that we have created. For this project we make two frame with one hundred data each. After add the frame with the value that have been tested we decided that $8m/s^2$ is the best value to be used, to detect any fault and it has fulfilled the criteria where the value of accelerometer must be two times bigger than the normal data.

4.3 Fault Detection with $8m/s^2$ Accelerometer Value

The best value of accelerometer to be test the fault data has been chosen. Hence, we show every result taken by only using the same iteration and the same accelerometer value but with multiple run and also we show the error of detecting. In this result we not only show the fault data testing but also the normal data that have no fault to be detected to make sure PSO only can detect the vibration of the fault data not include the vibration of turning point. Every fault data that PSO detect must be add with the value of data in frame.

1. Fault on 1180

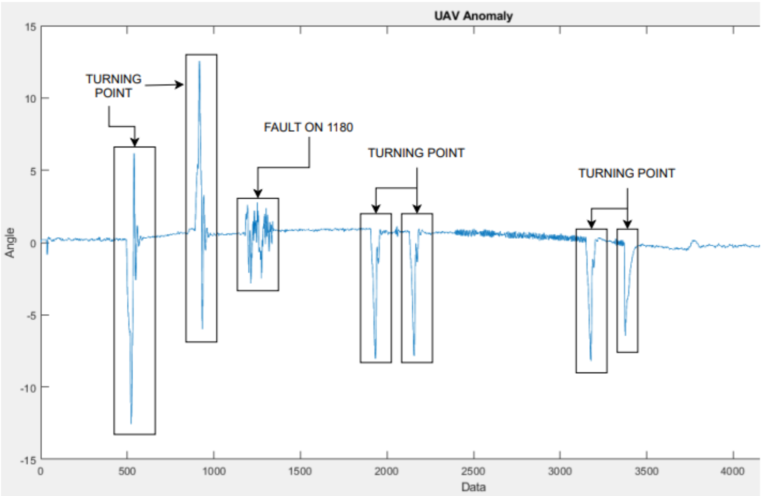


Figure 4.8 Vibration Fault

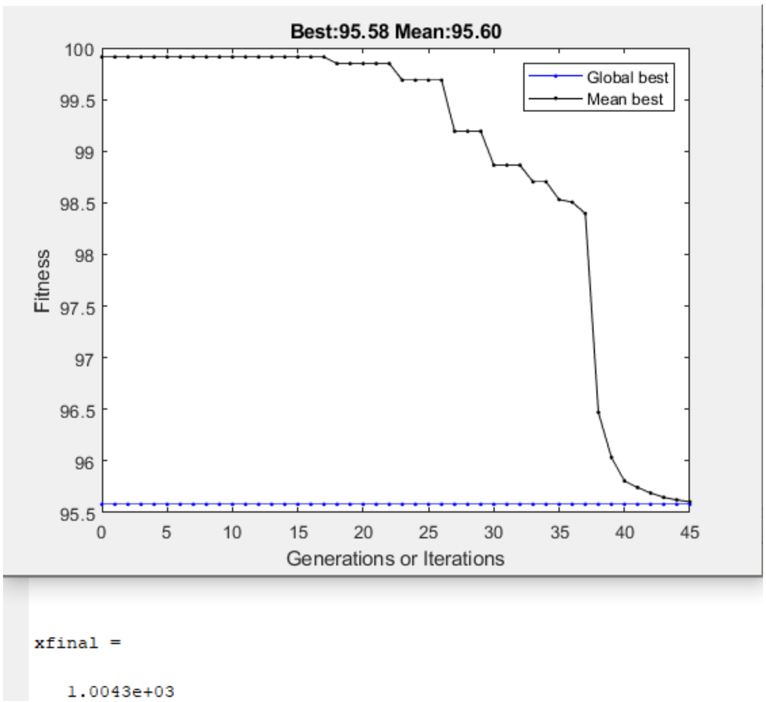


Figure 4.9 Error detect_1004

There have two figure shown above the first one **Figure 4.9** where the figure shows that is the place where vibration of fault has happened. It supposed to be detected at the 1180 value but PSO has detect at the 1004 value and to be accurate we must add the frame data value which is 200 more data. After we plus the frame data PSO has detected on 1204 which over data around 24 which is it is still in the fault just not detected the beginning but also not to late detected. We proceed to the next detection to be more clear.

2. Fault on 1416

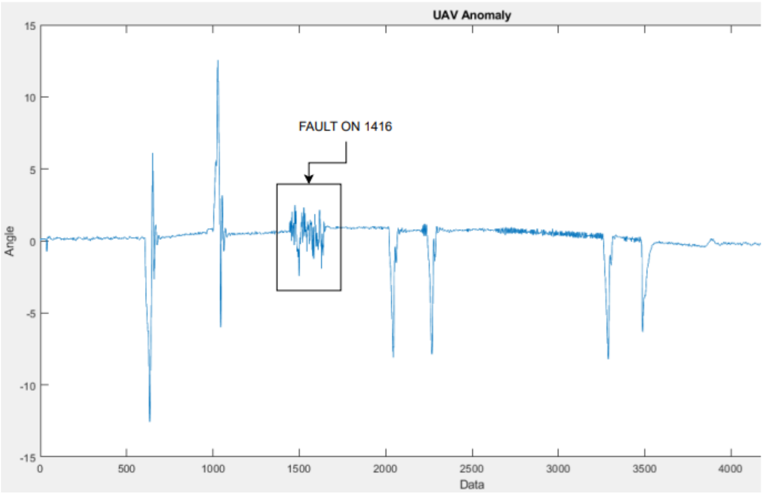


Figure 4.10 Vibration Fault 1

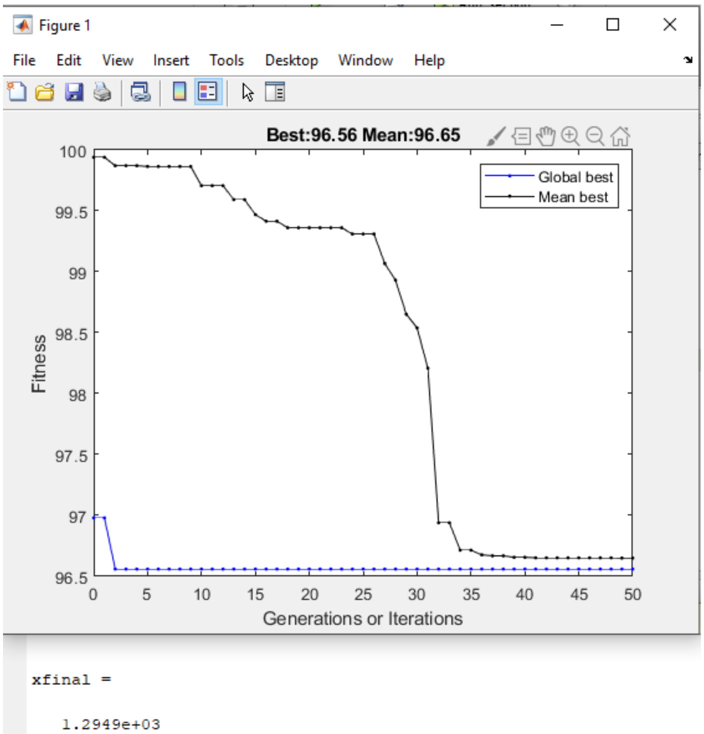


Figure 4.11 Error detect 1295

3. Fault on 2488

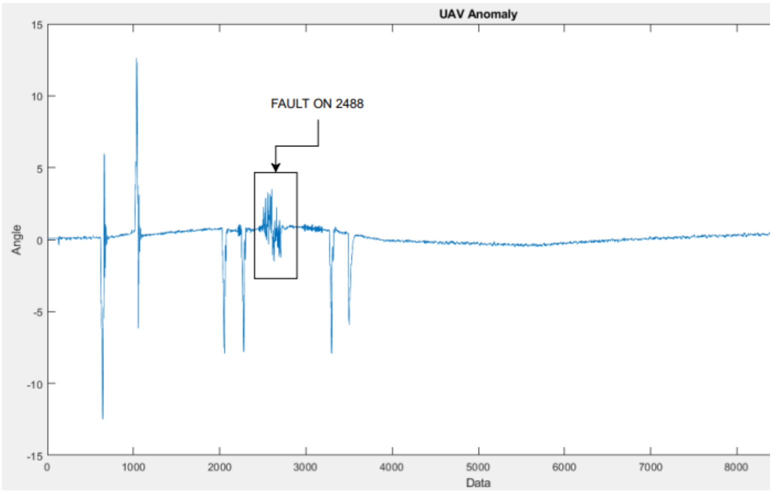


Figure 4.12 Vibration Fault 2

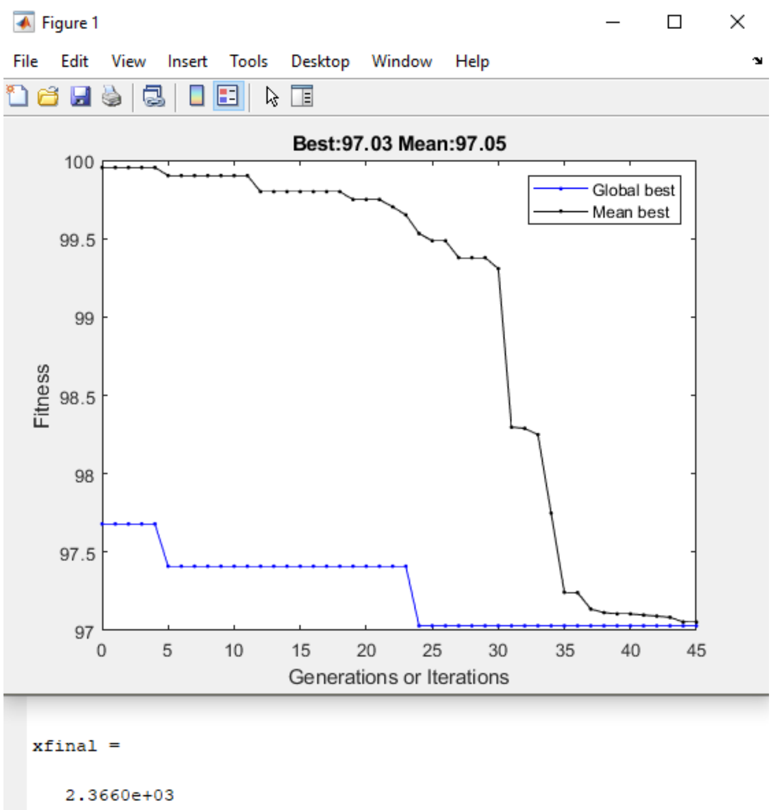


Figure 4.13 Error detect 2366

4. Fault on 1157

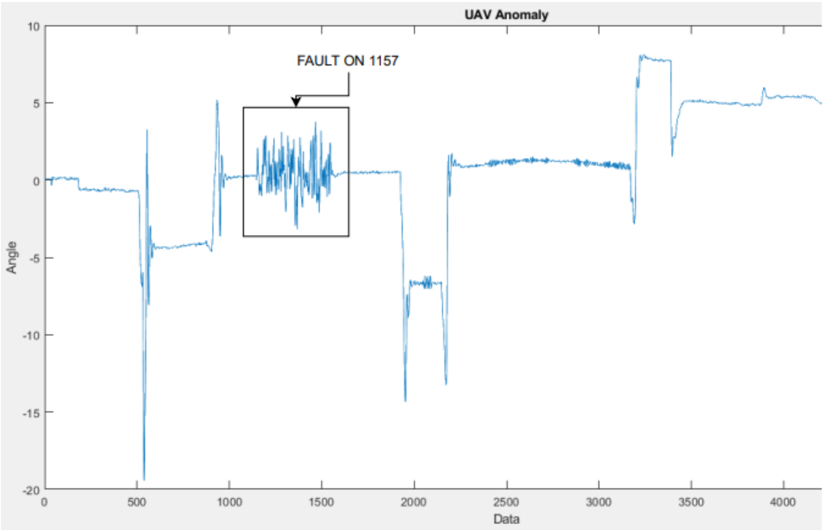


Figure 4.14 Vibration Fault 3

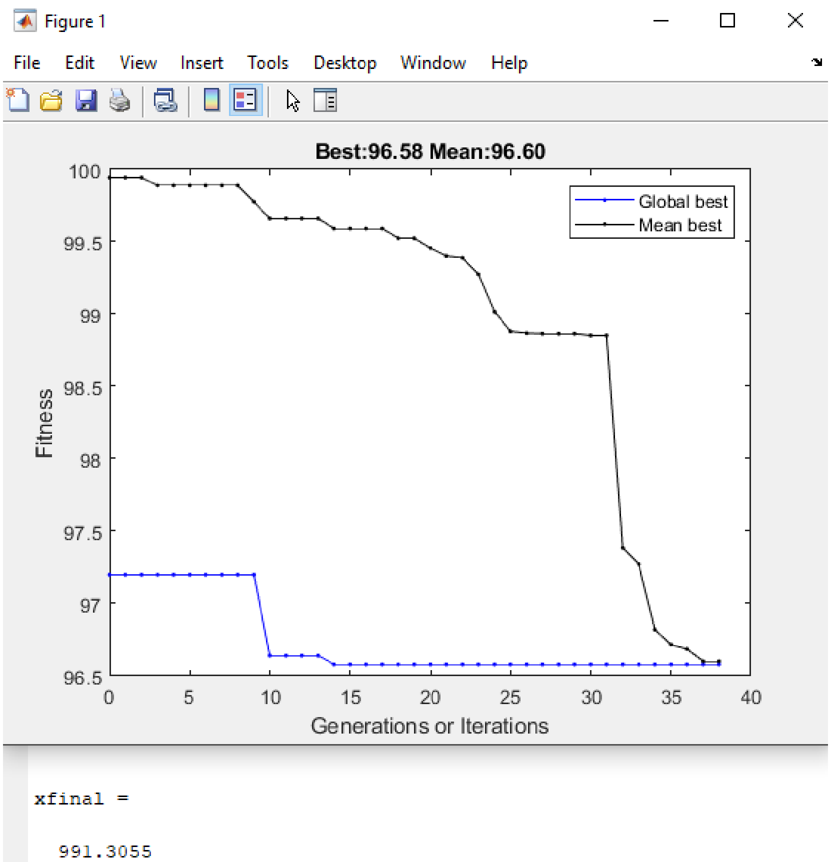


Figure 4.15 Error detect 991

5. Fault on 2090

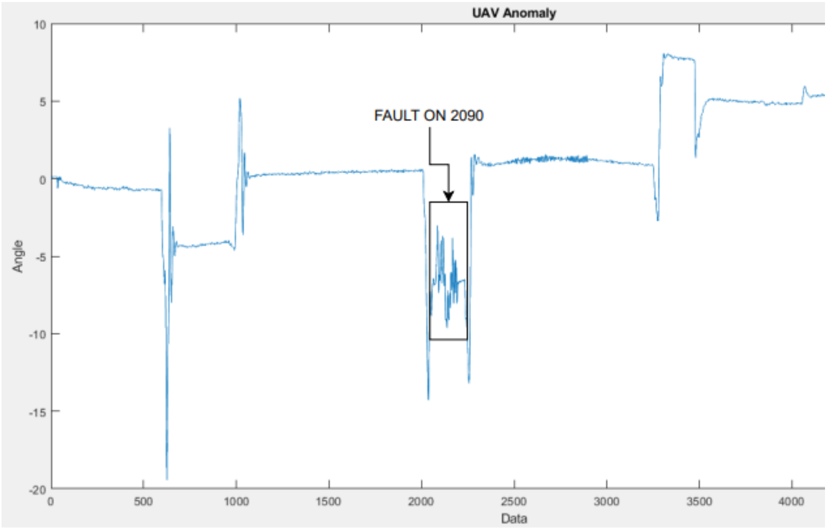


Figure 4.16 Vibration Fault 4

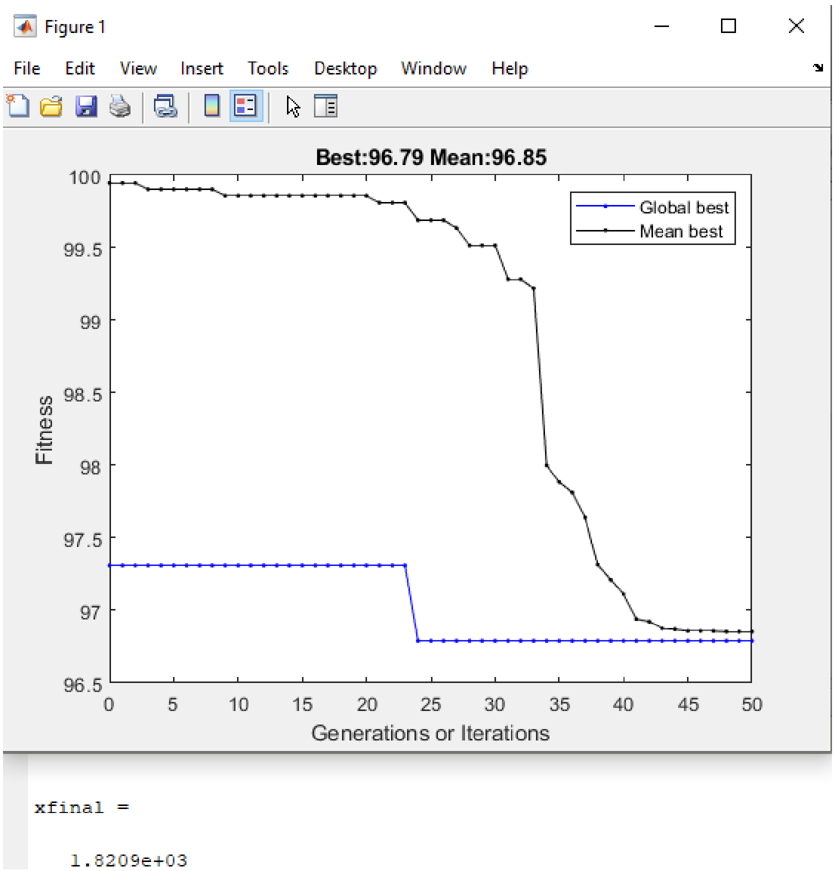


Figure 4.17 Error detect 1821

6. Fault on 2425

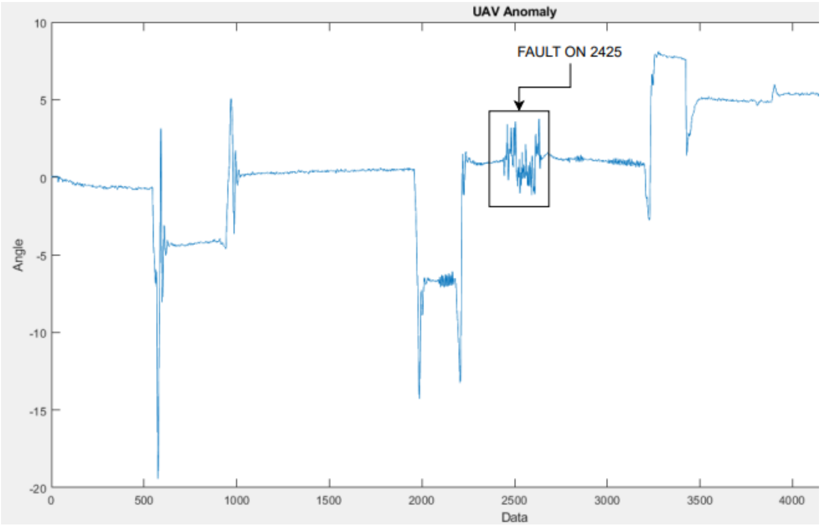
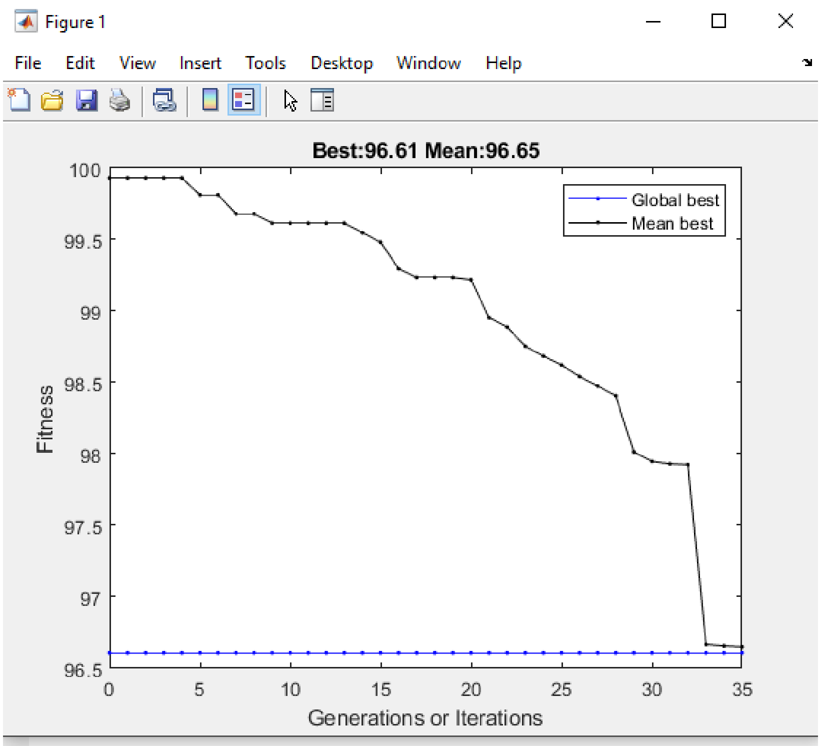


Figure 4.18 Vibration Fault 5




```
xfinal =

   2.2825e+03
```

Figure 4.19 Error detect 2283

7. Fault on 1191

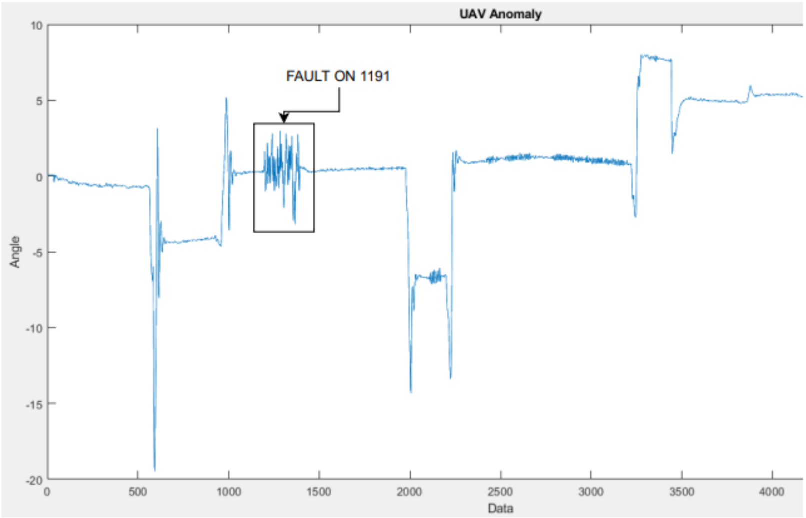


Figure 4.20 Vibration Fault 6

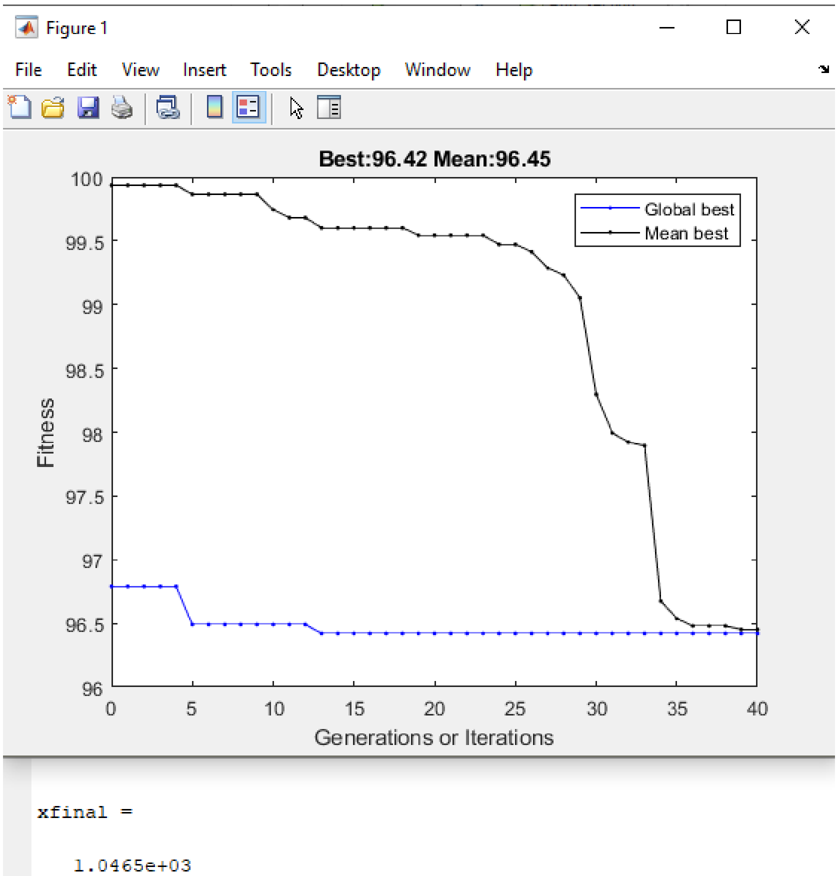


Figure 4.21 Error detect 1047

8. Fault on 1433

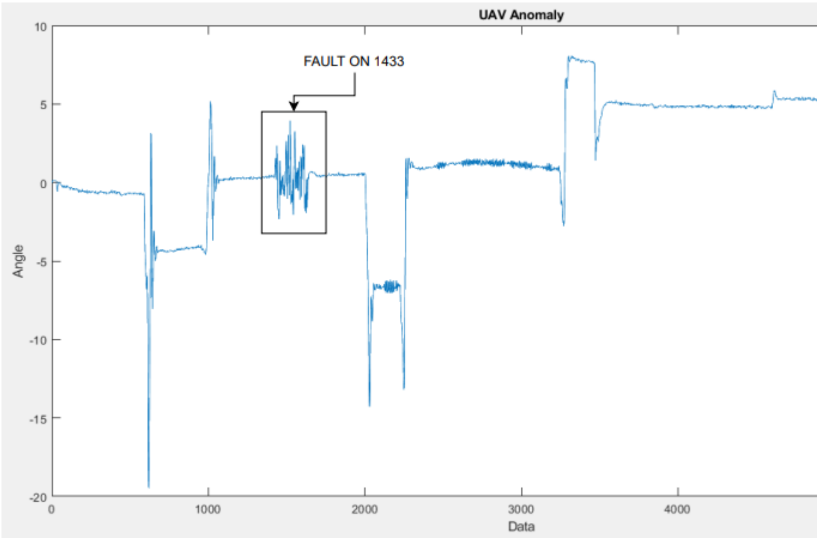


Figure 4.22 Vibration Fault 7

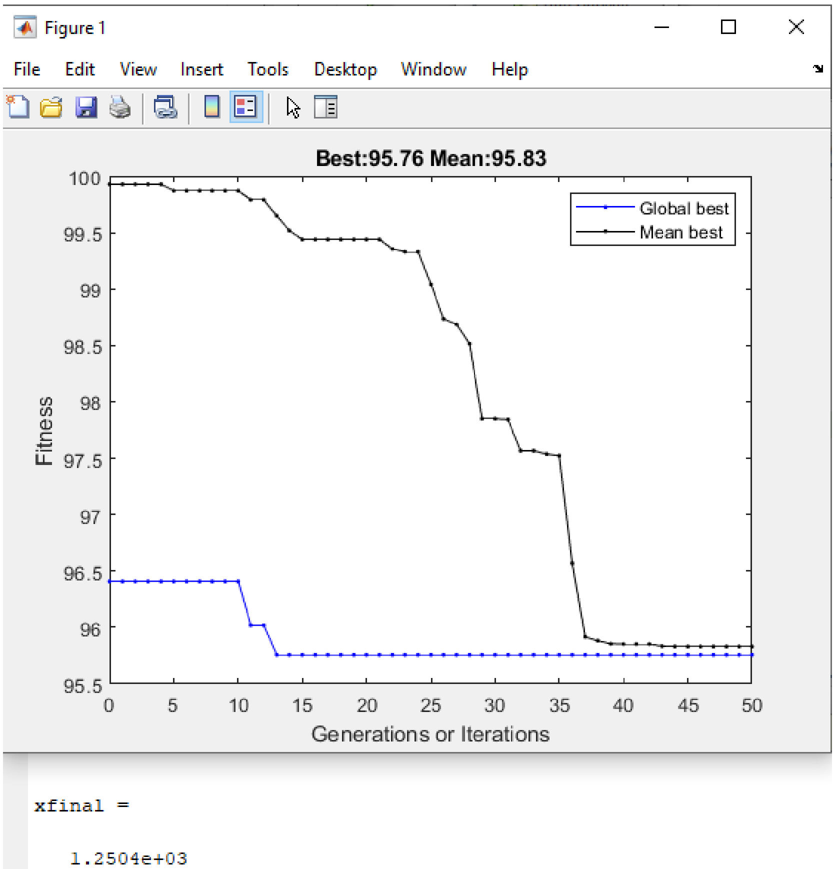


Figure 4.23 Error detect 1250

9. Fault on 1602

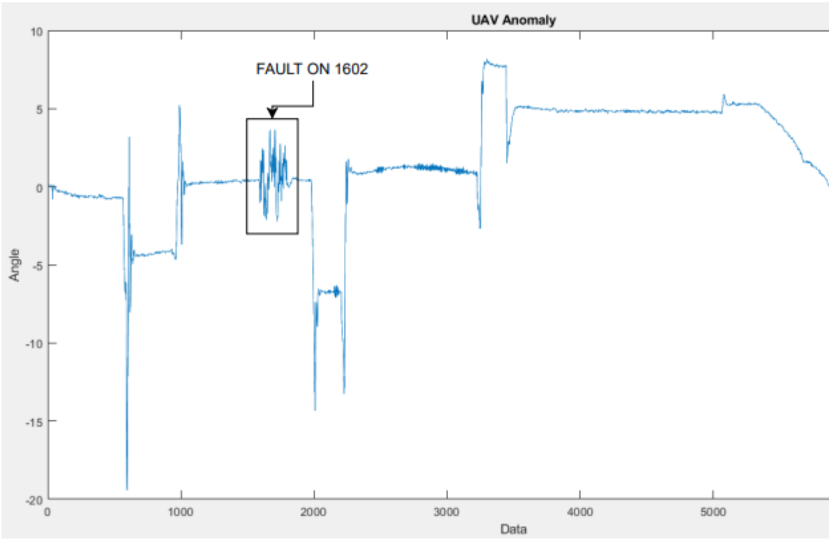


Figure 4.24 Vibration Fault 8

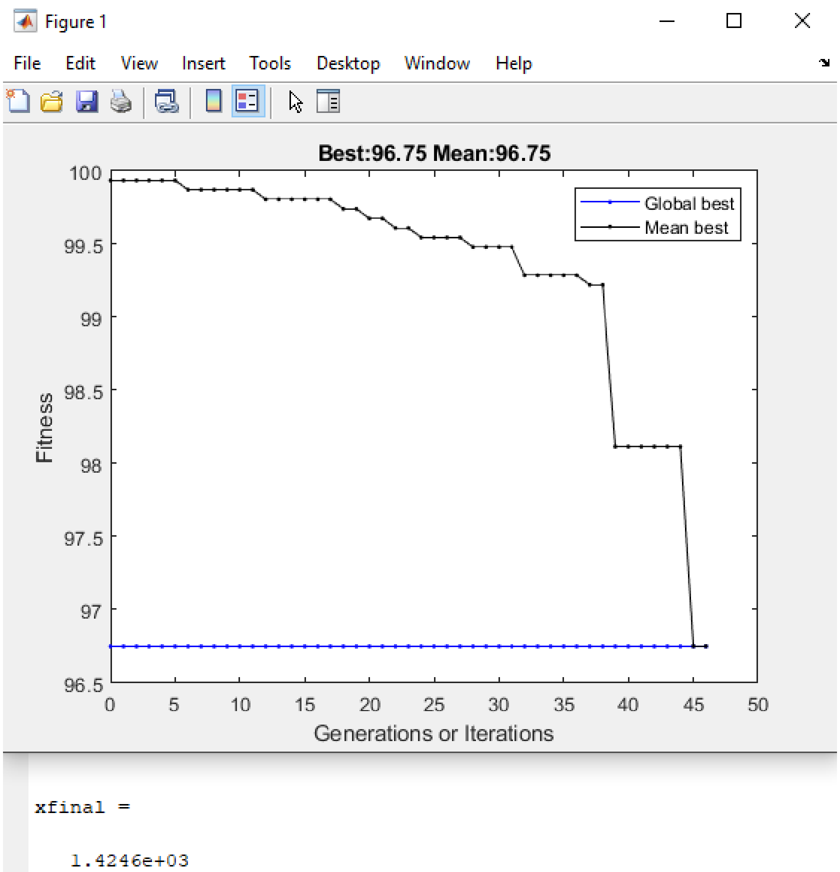


Figure 4.25 Error detect 1424

10. Fault on 1815

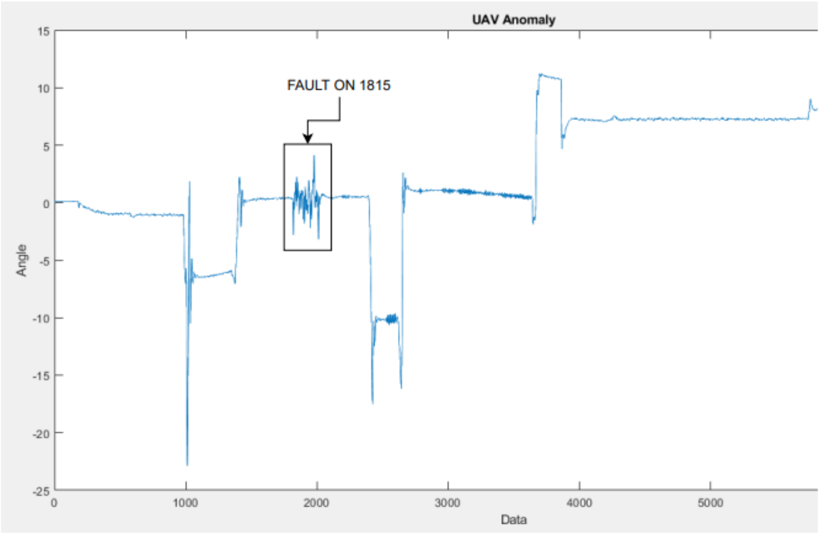


Figure 4.26 Vibration Fault 9

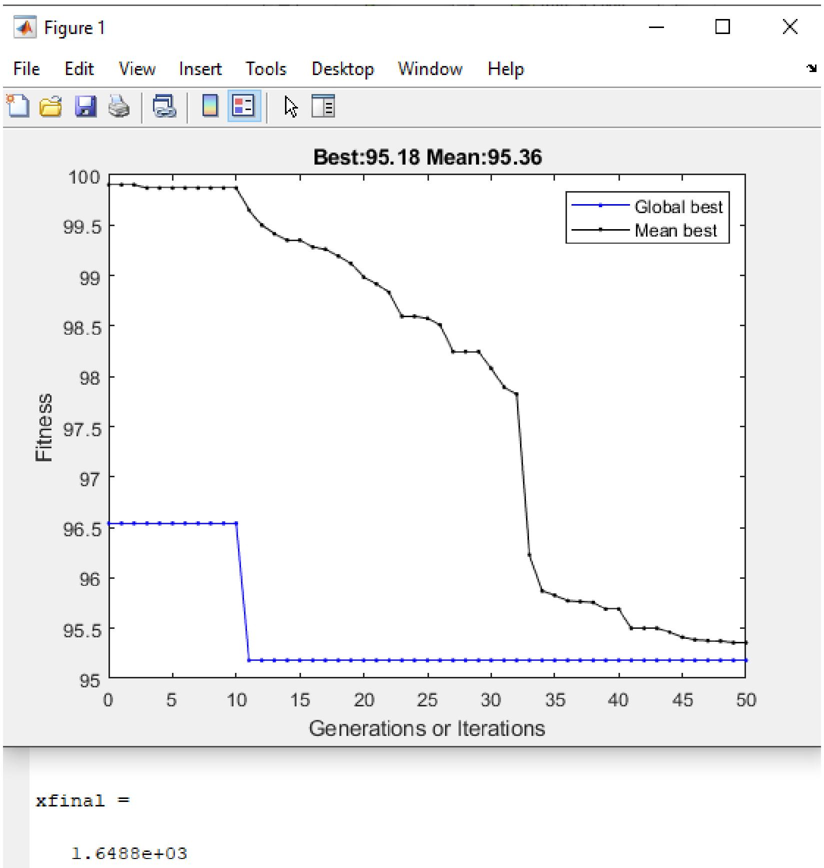


Figure 4.27 Error detect 1649

11. Fault on 5609

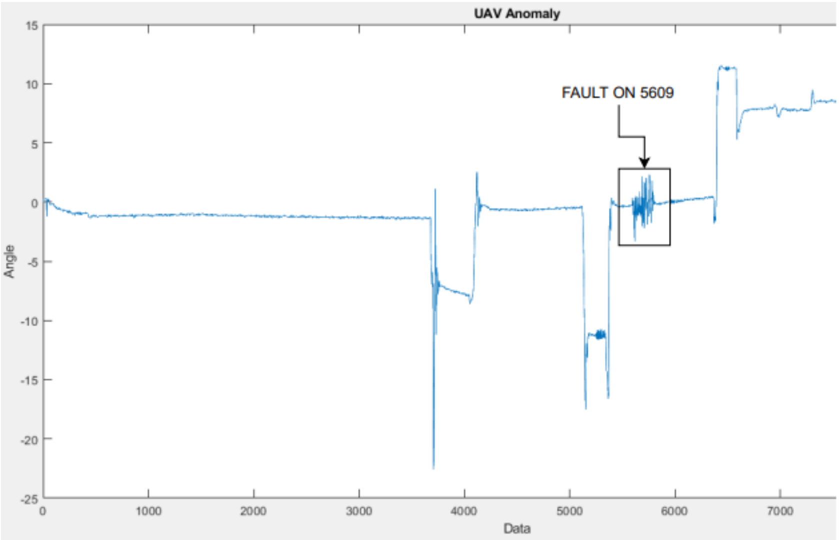


Figure 4.28 Vibration Fault 10

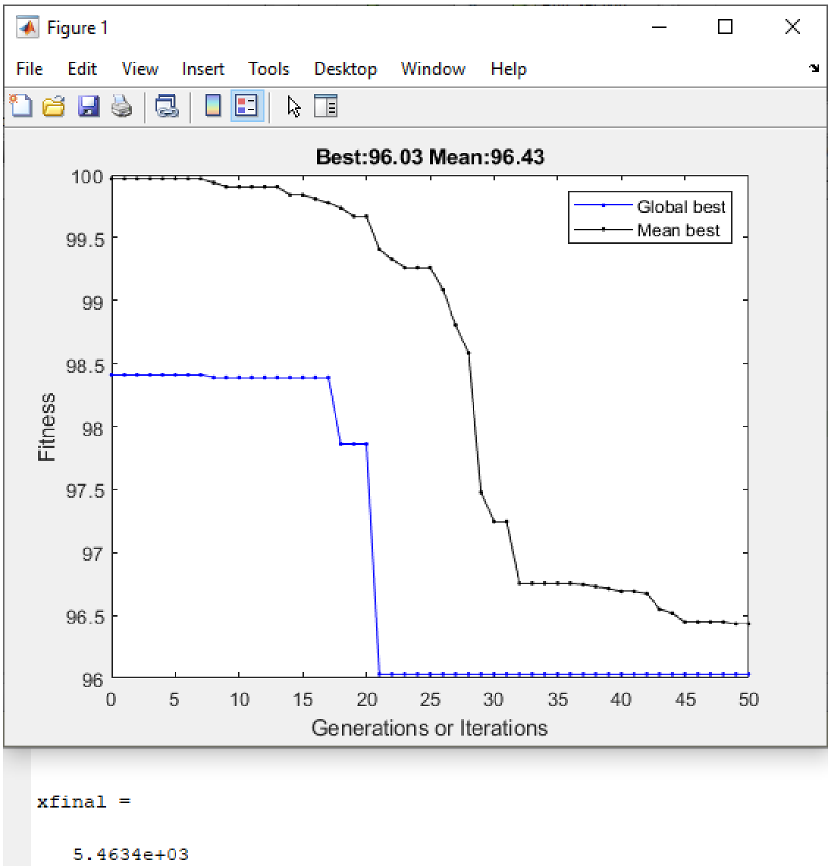


Figure 4.29 Error detect 5463

12. Fault on 2606

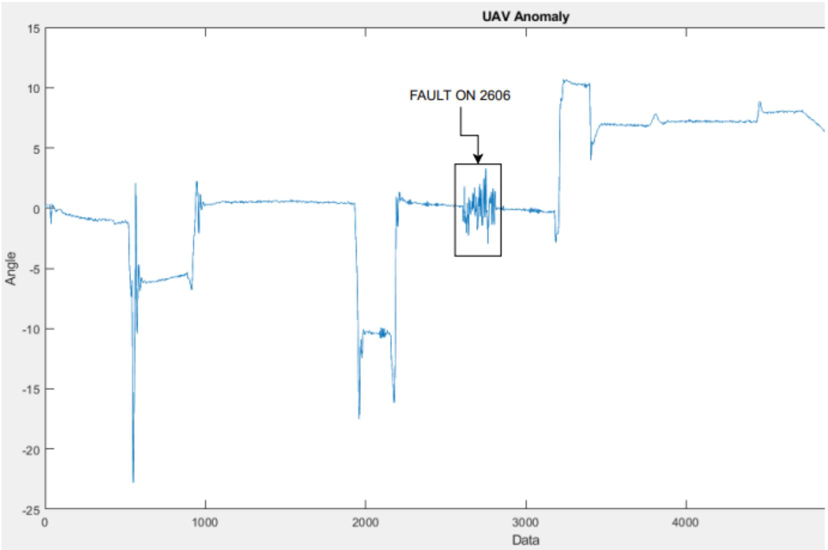


Figure 4.30 Vibration Fault 11

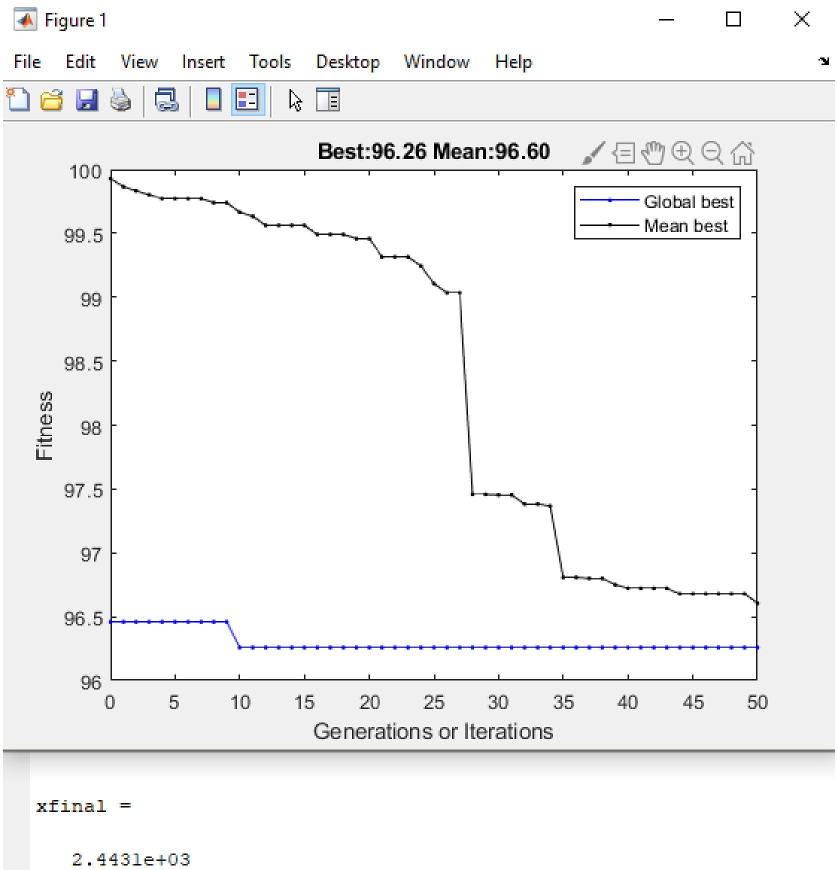


Figure 4.31 Error detect 2443

4.4 Healthy Data Testing

Healthy data also being tested, to make sure that if UAV not making any fault so that PSO would not detect any fault. In this healthy data, we have set the swarm size is 50 and the maximum iteration is 40. To clarify that PSO not detect any fault at the healthy data, we can see at the iteration where we set maximum iteration is 40. The fitness function not detect any fault until reach the maximum iteration.

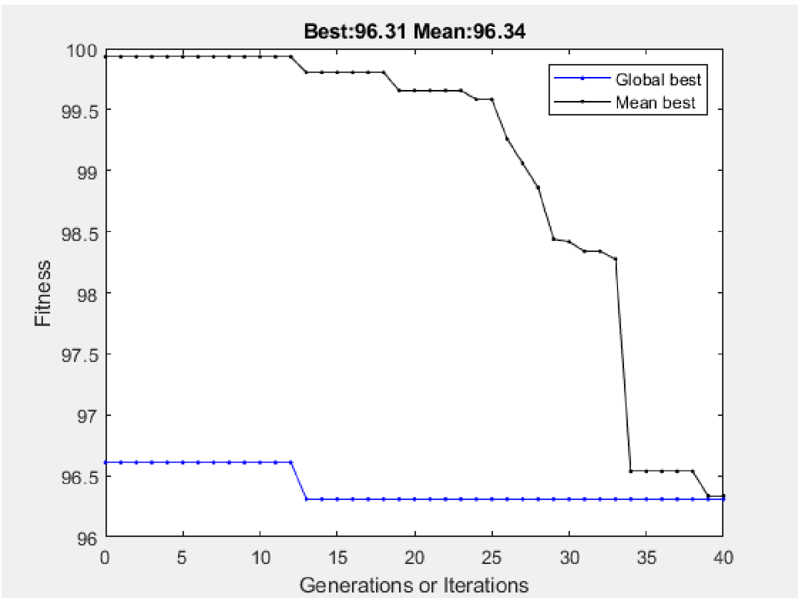


Figure 4.32 Fitness function with max iteration

The result is same with another 3 healthy data, which is it reach a maximum iteration that we set and not detect any fault on it.

4.5 Chaos Matrix

Table 3 Chaos Matrix Data

	FAULT DATA	HEALTHY DATA
DETECT	- Fault at 1180 - Fault at 1191 - Fault at 1416 - Fault at 1433 - Fault at 2488 - Fault at 1602 - Fault at 1157 - Fault at 1815 - Fault at 2090 - Fault at 5609 - Fault at 2280 - Fault at 2606	- No Fault 1 - No Fault 2 - No Fault 3 - No Fault 4
NO DETECT	- Fault at 2447 - Fault at 3553 - Fault at 2667 - Fault at 2950 - Fault at 1952	

There 21 data have been tested, 16 from that are detect. From this chaos matrix we have 76% successful test.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter will summarize the conclusion on the finding and analysis of fitness function determination of Unmanned Aerial Vehicle (UAV) anomaly detection in large data set via Particle Swarm Optimization (PSO).

5.2 Conclusion

In summarize, we succeed to detect fault on Unmanned Aerial Vehicle (UAV) which is the fault that produce by motor/blade and it's the one that can induce vibration. Second achievement is we succeed to collect the large set of data by using PSO with reducing the time taken. Furthermore, by using its vibration we monitoring the fault until we only detect the vibration that produce by fault not by any turning point that UAV made. The effectiveness and also the detection rate being monitor and the result are good with minimum set of time and large dataset. We overcome natural interrupt to UAV during fly with only test using software in loop (SIL). We succeed in reduce the costing part of the UAV testing and detection of fault. In the future, it can be improved on collecting data not just with one anomaly and also it can be improve by testing in real time fault of UAV.

REFERENCE

[1] Keipour, Azarakhsh & Mousaei, Mohammadreza & Scherer, Sebastian. ALFA: A dataset for UAV fault and anomaly detection. The International Journal of Robotics Research. 40. 027836492096664. 10.1177/0278364920966642.

[2] Keipour A, Mousaei M and Scherer S (2019) Automatic real time anomaly detection for autonomous aerial vehicles. In: 2019 IEEE International Conference on Robotics and Automation (ICRA). pp. 5679–5685. DOI:10.1109/ICRA.2019. 8794286. URL https://ieeexplore.ieee.org/ document/8794286.

[3] F. Pourpanah, B. Zhang, R. Ma and Q. Hao, "Anomaly Detection and Condition Monitoring of UAV Motors and Propellers," 2018 IEEE SENSORS, 2018, pp. 1-4, doi: 10.1109/ICSENS.2018.8589572.

[4] S. Islam, P. X. Liu, and A. El Saddik, “Robust control of four-rotor un manned aerial vehicle with disturbance uncertainty,” IEEE Transactions on Industrial Electronics, vol. 62, no. 3, pp. 1563–1571, 2015.

[5] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, “Motor anomaly detection for unmanned aerial vehicles using reinforcement learning,” IEEE Internet of Things Journal, 2017.

[6] Z. Xu, J. Xuan, T. Shi, B. Wu, and Y. Hu, “A novel fault diagnosis method of bearing based on improved fuzzy artmap and modified distance discriminant technique,” Expert Systems with Applications, vol. 36, no. 9, pp. 11 801–11 807, 2009.

[7] G. A. Carpenter, S. Grossberg, and D. B. Rosen, “Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” Neural Networks, vol. 4, no. 6, pp. 759 – 771, 1991.

[8] A. Benini, F. Ferracuti, A. Monteriù and S. Radensleben, "Fault detection of a VTOL UAV using Acceleration Measurements," 2019 18th European Control Conference (ECC), 2019, pp. 3990-3995, doi: 10.23919/ECC.2019.8796198.

[9] P. Freeman, R. Pandita, N. Srivastava, and G. J. Balas, “Model based and data-driven fault detection performance for a small uav,” IEEE/ASME Trans. Mechatron., vol. 18, no. 4, pp. 1300–1309, 2013.

[10] A. Bondyra, P. Gasior, S. Gardecki and A. Kasiński, "Fault diagnosis and condition monitoring of UAV rotor using signal processing," 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2017, pp. 233-238, doi: 10.23919/SPA.2017.8166870.

[11] R. Loh, Y. Bian and T. Roe: Safety Requirements for Unmanned Aerial Vehicles (UAV) in Future Civil Airspace, 2006 IEEE/ION Position, Location, And Navigation Symposium, Coronado, CA, 2006, pp. 1151- 1163.

[12] M. Witczak, M. Pazera: Fault Tolerant-Control: Solutions and Challenges, Pomiary Automatyka Robotyka, Vol. 20, No. 1/2016, pp. 5-16

[13] S. Gardecki, W. Giernacki, J. Goslinski, A. Kasinski: An adequate mathematical model of four-rotor flying robot in the context of control simulations, Journal of Automation Mobile Robotics and Intelligent Systems, Vol. 8, No. 2, pp 9-17, 2014

[14] T. Uktveris, V. Jusas: Comparison of Feature Extraction Methods for EEG BCI Classification. Information and Software Technologies, Com munications in Computer and Information Science, vol 538. Springer, 2016, pp. 81-92

[15] F D Wihartiko et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 332 012020

[16] Alasadi, Hamid & Al-Mansoori, Mohammed & Hitam, Salasiah & Saripan, M Iqbal & Mahdi, Mohd Adzir. (2011). Particle swarm optimization on threshold exponential gain of stimulated Brillouin scattering in single mode fibers. Optics express. 19. 1842-53. 10.1364/OE.19.001842.

[17] M. Donelli, and A. Massa, “Computational approach based on a particle swarm optimizer for microwave imaging of two-dimensional dielectric scatterers,” IEEE Trans. Microw. Theory Tech. 53(5), 1761–1776 (2005).

APPENDIX A

CODING: MAIN

main.m | PSO_fixed.m | OPT.m

```
global iter_out;
global datafile;
% LOAD FLIGHT DATA TO SEARCH FOR TURNING
load noerror2_8
datafile=noerror2.ATT;

% FINDING HOW BIG IS THE DATA
%[m,n]=size(time);
x=10;
cf=@const;
fitnessfcn=@(x) OPT(x);
nvars=1;                    % bilangan variable

[m,n]=size(datafile);
LB = 10;                          % lower bound BASED ON HOW BIG THE DATA
UB = m-200;                      % upper bound BASED ON HOW BIG THE DATA

iter_out=0;
while (iter_out<3)
x=PSO_fixed(fitnessfcn,nvars,LB,UB); % swarm = 200, iteration = 2000
end

[m,n]=size(x)
xfinal=x(m,2)
```

APPENDIX B

CODING: PSO

main.m | PSO_fixed.m | OPT.m | +

```
function Out=PSO_fixed(objfun,nvar,lb,ub,options)          % function starts
%%
global iter_out
iter_out=0;
tic
if nargin == 4                                      % if no input for options
options.swrmsize=50;                                % swarm or population size
options.itermax=40;                                % maximum iterations
options.c=[1.05 1.05]; options.wt=[1 0.3];          % inertia wieght parameters
options.plt=1;                                      % plotting command 0
elseif nargin < 4  && nargin > 5                    % .
disp('Inputs mismatch')                             % else condition
end                                                 % end if
%%
signal=0;
D=nvar;                                             % D: dimentionality
N=options.swrmsize;                                 % Option parameters
itermax=options.itermax;                            % .
c1=options.c(1); c2=options.c(2);                   % .
wmax=options.wt(1); wmin=options.wt(2);             % .
plt=options.plt;                                    % .
w=linspace(wmax,wmin,itermax);                      % Inertia vector
a=repmat(lb,N,1);                                   % lb for all swarms
b=repmat(ub,N,1);                                   % ub for all swarms
d=(b-a);                                            % parameter space
q=(b-a)/1;                                          % 1/1 of parameter space
```

main.m | PSO_fixed.m | OPT.m | +

```
x=a+d.*rand(N,D);                                   % initial position
v=q.*rand(N,D);                                     % initial velocity
for ni=1:N                                          % for each swarm
f(ni,1)=objfun(x(ni,:));                            % fitness or goal
end                                                 % end for each swarm
[fgbest,igbest]=min(f);                             % fittest minimum in all
gbest=x(igbest,:);                                  % Best among all gbest
pbest=x; fpbest=f;                                  % Best in each swarm
Out=[];                                             % Output as pbest gbest
Out=[Out;0 gbest fgbest mean(fpbest)];              % initial pbest and gbest
for it=1:itermax                                    % for each iteration
v(1:N,1:D)=w(it)*v(1:N,1:D)+...                     % update velocities
    c1*rand*(pbest(1:N,1:D)-x(1:N,1:D))...          % .
    +c2*rand*(repmat(gbest,N,1)-x(1:N,1:D));        % .
x(1:N,1:D)=x(1:N,1:D)+v(1:N,1:D);                   % update positions

for loop1=1:D %nvar
    for loop2=1:N %population
    if(x(loop2,loop1)<lb(loop1))
         x(loop2,loop1)=lb(loop1);
    end
        if(x(loop2,loop1)>ub(loop1))
         x(loop2,loop1)=ub(loop1);
        end

    end
```

APPENDIX C

CODING: FITNESS FUNCTION (MONITORING BASED)

```
function y=OPT(x)
global datafile;
x=floor(x);

% ambil sepuluh data
Ema=[datafile(x); datafile(x+1); datafile(x+2);datafile(x+3); datafile(x+4); datafile(x+
     datafile(x+20);datafile(x+21);datafile(x+22);datafile(x+23);datafile(x+24);datafile
     datafile(x+40);datafile(x+41);datafile(x+42);datafile(x+43);datafile(x+44);datafile
     datafile(x+60);datafile(x+61);datafile(x+62);datafile(x+63);datafile(x+64);datafile
     datafile(x+80);datafile(x+81);datafile(x+82);datafile(x+83);datafile(x+84);datafile
x=x+100;
Ema1=[datafile(x); datafile(x+1); datafile(x+2);datafile(x+3); datafile(x+4); datafile(x
     datafile(x+20);datafile(x+21);datafile(x+22);datafile(x+23);datafile(x+24);datafile
     datafile(x+40);datafile(x+41);datafile(x+42);datafile(x+43);datafile(x+44);datafile
     datafile(x+60);datafile(x+61);datafile(x+62);datafile(x+63);datafile(x+64);datafile
     datafile(x+80);datafile(x+81);datafile(x+82);datafile(x+83);datafile(x+84);datafile


EmaPeak=max(Ema);
EmaMin=min(Ema);
EmaBeza=EmaPeak-EmaMin;

EmaPeak1=max(Ema1);
EmaMin1=min(Ema1);
EmaBeza1=EmaPeak1-EmaMin1;
```

main.m PSO_fixed.m OPT.m

```
Y=100;
if (EmaBeza < (2*0.18))
        if(EmaBeza1 > (2*0.18))
            Y=EmaBeza1-EmaBeza;
            x=x-20;
            Emma=datafile(x+8)-datafile(x+2)/6;
            Emma1=datafile(x+18)-datafile(x+12)/6;
                        if (Emma1<Emma +0.000002) & (Emma1>Emma -0.000002) % tolerance
                            Y=100;
                        else
                            if (std(Ema1)>0.45&&std(Ema1)<0.7)

                             Y=100-Y;

                            else
                                Y=100;
                            end
                        end





    end
end
```

APPENDIX D

CODING: FIND STANDARD DEVIATION VALUE

```
main.m | PSO_fixed.m | OPT.m | std_graph.m

clc
clear all

% LOAD FLIGHT DATA TO SEARCH FOR TURNING
load vibrationfull
datafile=vibration1ch.ATT;
[m,n]=size(datafile);
u=rem(m,100);
x=0;
m=m-u;
for i=1:100:m
    x=[x; std(datafile(i:i+99))];
end
```