2003

Blind fault detection using spectral signatures

Pallavi Chethan
Louisiana State University and Agricultural and Mechanical College

BLIND FAULT DETECTION USING SPECTRAL SIGNATURES

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

by
Pallavi. Virupaksha. Chethan
B. E, Siddaganga Institute Of Technology, Tumkur, 2000
August 2003

ACKNOWLEDGMENTS

I would like to first thank the almighty for his grace. My husband Mr Chethan Parameswariah has been a great support for me. His constant encouragement and love kept me going. Without him it would not have been possible for me to conquer this dream of mine.

I take this opportunity to express my deepest thanks and gratitude to Dr Jorge Aravena, my major professor, for his academic guidance and support during this program. He was always there to answer my never ending questions.

Last but not the least I would like to thank my family back at home, my constant source of energy. My mother Mrs Girija Virupaksh, my father Dr Virupaksh for always being there for me. My sister and her family Mrs Prathima, Mr Dayanidhi , my brother and his family Dr Shiva Prakash, Mrs Pavithra, my nephews Rohan and Pranav for their love and encouragement at all times. My in-laws for their encouragement. Finally my friends who are my family away from home.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This work studies a blind fault detection method, which only analyses a system's output signal for any change in the characteristics from pre-fault to post-fault to identify the occurrence of faults. In our case the fault considered to develop the procedure is change in time constant of an aircraft's aileron-actuator system and its simplified version - a position servo system. The method is studied as an alternative to conventional fault detection and identification methods.

The output signal is passed through a filter bank to enhance the effect of a fault. The Short time Fourier transform is performed on the enhanced pre-fault and post-fault signals components to obtain indicators. Fault detection is approached as a clustering problem determining distances to fault signatures. This work presents two techniques to create signatures from the indicators. In the first method, the mean of the indicators is the signature. Tests on a position servo system show that the method effectively classifies the indicators by more than 85 % and can be used for online classification. A second method uses Principal Component Analysis and defines vector sub-space signatures. It is observed that for the position servo system, the pre-fault indicators had 14 % of false alarms and post-fault indicators the missed the faults by 17%. This second method was also applied to one axis model of an F-14 aircraft's aileron-actuator system. The results obtained showed around 80 % of correctly identified pre-fault indicators and post-fault indicators.

The blind fault detection method studies has potential but needs to be understood further by applying it to more varied cases of faults and systems.

CHAPTER 1

INTRODUCTION

The world is full of processes, complex systems in general, which are part of our everyday life. The systems are likely to fail at some time. Timely detection and identification of the faults are important to prevent extensive damage to persons, the environment and the systems themselves. Fault Detection and Identification is a major area of research in the field of automatic control. This issue has been one of the concerns in aircraft safety systems, where the progressive faults of the engine parts not detected in early stages can lead to disaster. This research focuses on the methods of detection for such dormant faults in its initial stages. This research is a part of the project Aircraft Safety Control Upset Management, sponsored by NASA under the ESPCoR 2000 program whose goal is to improve aircraft safety [1].

One of the most widely used fault detection and identification methods is the "model based fault detection". In this method, a mathematical model is created knowing the input and output of the system, and this mathematical model is used to compare with nominal behavior or nominal model of the system and the residuals are formed. From these residuals, any change from the nominal behavior is detected and the fault is diagnosed. For the fault diagnosis several other methods have also been developed. For example, parameter estimation, state variable estimation [2]-[4], neural networks, fuzzy systems and

also neuro-fuzzy approaches [5]-[7] are being studied. The development of newer methods is still an ongoing process.

One of the interesting areas of research in the field of fault detection is the "Model-free fault detection system" [1], [8]. Unlike the model-based system, in this model-free method an accurate mathematical model is not necessary. When the system is complex, it is difficult to develop an accurate mathematical model that represents the true system. Approximations and assumptions are made in modeling that compromise the accuracy. Also in model-based fault detection, it is always essential to know the input to the system, which in certain cases might not be possible. For example, in an aircraft system, it is difficult to always have an access to the system's input, whereas output of the system is mostly available. In contrast to the model-based detection systems, model-free fault detection system does not need model of the system always (though some knowledge helps in the analysis of the behavior). A type of mode-free system is "Blind fault detection", which does not require the knowledge of the input to the system. Signal-processing techniques are applied on the output signals only and the system is blind with respect to its input signal. This can be more compared to the data-mining problem in geophysical studies, where the input that causes the disturbances in the earth is not easily known but we can analyze and predict the nature of the ground just by analyzing the output signal obtained from seismometers.

This research tests the concept that by using signal processing methods alone, it is possible to detect and identify faults. It is proposed to create signatures pertaining to different conditions such as normal and faulty condition. Creation of signatures is carried out offline with trained faults and outputs but with little or no prior knowledge of the

system. This knowledge of signatures is then applied for online fault detection where the output of the system is processed to create indicators and clustering techniques are used on the indicators to identify them with the signatures, and hence identifying the condition of the signal.

This research mainly focuses on a new method of creation of signatures based on the energy distribution changes in the signal between normal and faulty signals. Vector subspace and principle components analysis (PCA) techniques are then applied to separate the non-faulty and faulty signatures. The test systems used in the research are a F-14 jet fighter and a position servo system both simulated using Matlab's toolbox - Simulink. A simulated fault introduced to study the blind fault detection method is the change in time constant of the movement of the actuator which is responsible, for the movement of the aileron in the jet fighter and the position output of the position servo system.

CHAPTER 2

SIGNALS AND CREATION OF INDICATORS

The test systems used in the research and the signal processing methods applied on the output signals to create indicators are discussed in this chapter.

The two test systems considered for our research purposes are F-14 jet fighter's aileron-actuator control system and a position servo system. The reason for using the F-14's aileron-actuator system is simply because it is available in Matlab's Simulink toolbox as a standard demo and can be easily modified for use in the research. The F-14 Tomcat was designed as a carrier based aircraft for the US Navy [9]. With its variable geometry wings the speed range of the aircraft could be varied from slow for carrier landings up to supersonic speeds. The role of the F-14 Tomcat was brought to the fore during the Gulf crisis in the early 90's [10].

Figure 1 F-14 Tomcat aircraft

By computer simulation, we track the flight stick movement, which is a filtered random noise emulating evasive and tracking actions in a dogfight. The fault is introduced in the actuator which is responsible for moving the aileron (components of the wing) of the plane (Simulink diagram of this system in Appendix A). This research uses one example to create a systematic procedure that can be applied to detect any fault. That one fault considered here is the change in time constant in the actuator. By increasing the time constant, the actuator response time to the input movement increases. Input to the system is a random noise generated with seed value that helps to repeat the exact input for different experiments. Also duration of simulation, the time constant and the time at which the fault is introduced can be varied in the Matlab script that runs the simulation (Matlab script in Appendix C).

The second test system considered in the research is a position servo system. A motor with a position feedback is a position servo which allows the motor to be commanded to rotate by a specified angle. The position servo system resembles very much the F-14's aileron-actuator system responsible for the positioning of the aileron in the aircraft and is a less complex system for analysis purposes (Simulink diagram in Appendix A). This position servo system is used as a test case because it is easily implemented and a very common control system. Here again fault is the change in time constant of the position servo system's response to the input applied, which is similar to the fault introduced in the F-14 system. This type of fault is one of the common faults that could affect any control system component. The position servo system is simpler compared to the F-14's aileron-actuator system making the analysis easier for a particular fault. The transfer function of the position servo system is given below.

$$G(s) = \frac{1}{s.(s+1).(t_p.s+1)} \qquad -(1)$$

Where t_p is the time constant that undergoes a fault.

A Matlab script in Simulink for the position servo system simulates fault applied and the position servo system. The simulation time, fault occurring time and magnitude of the fault can be specified before simulation (Matlab script in Appendix C). The input to the system is random noise generated from a seed which is very convenient to repeat the same conditions for different experiments.

In the simulations, fault was introduced after a specified time by changing the value of a time variable in the scripts. Figure 2 shows the output signals of F-14 aircraft's aileron-actuator servo system.

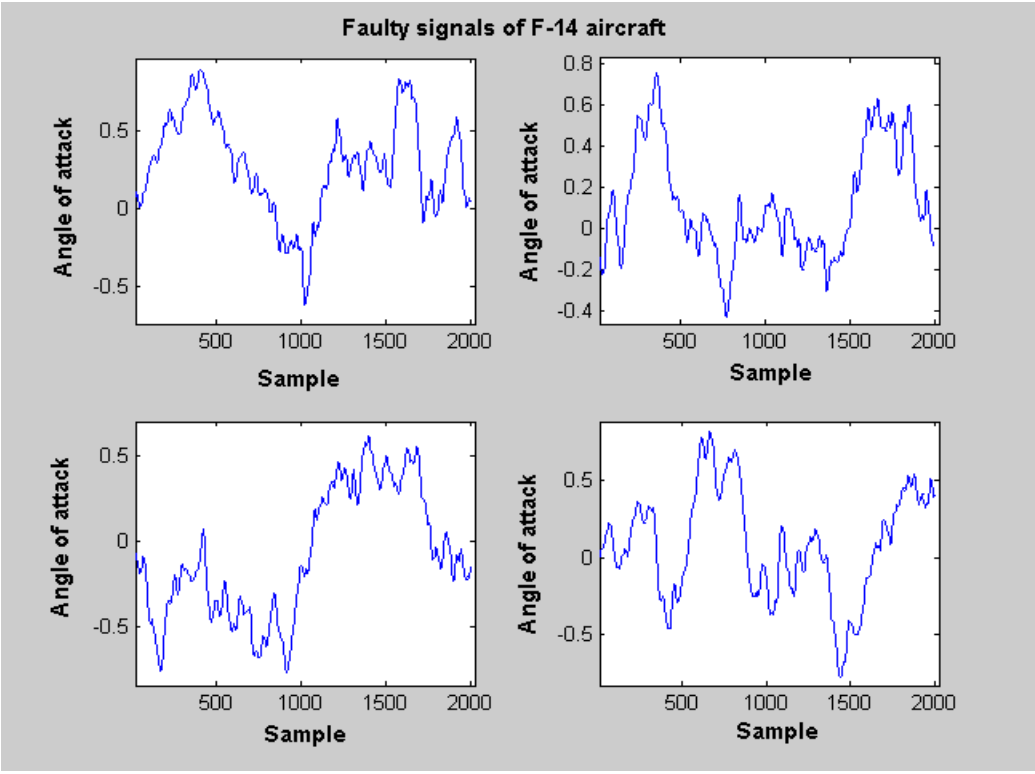


Figure 2 Simulated signals of F-14 used in the analysis. Fault introduced from 1000th sample

The effect of the fault is not apparent in the measured signal as it can be seen in figure 2. In fact, the varying output (which is normal) acts as "nuisance signal" for the purpose of fault detection and needs to be segregated from the fault effectively. This task of isolating the fault disturbance signal from the output is based on the premise that changes due to fault will be small details in the complete signal and the changes affect only narrow frequency bands of the signal [1]. Dividing the signal into narrow frequency bands and analyzing the signal in these narrow frequency bands, the small changes due to fault are enhanced. This kind of reduction by subtracting the frequency bands can be readily implemented by using existing perfect reconstruction filter bank techniques.

The basic perfect reconstruction filter is shown in the figure below.

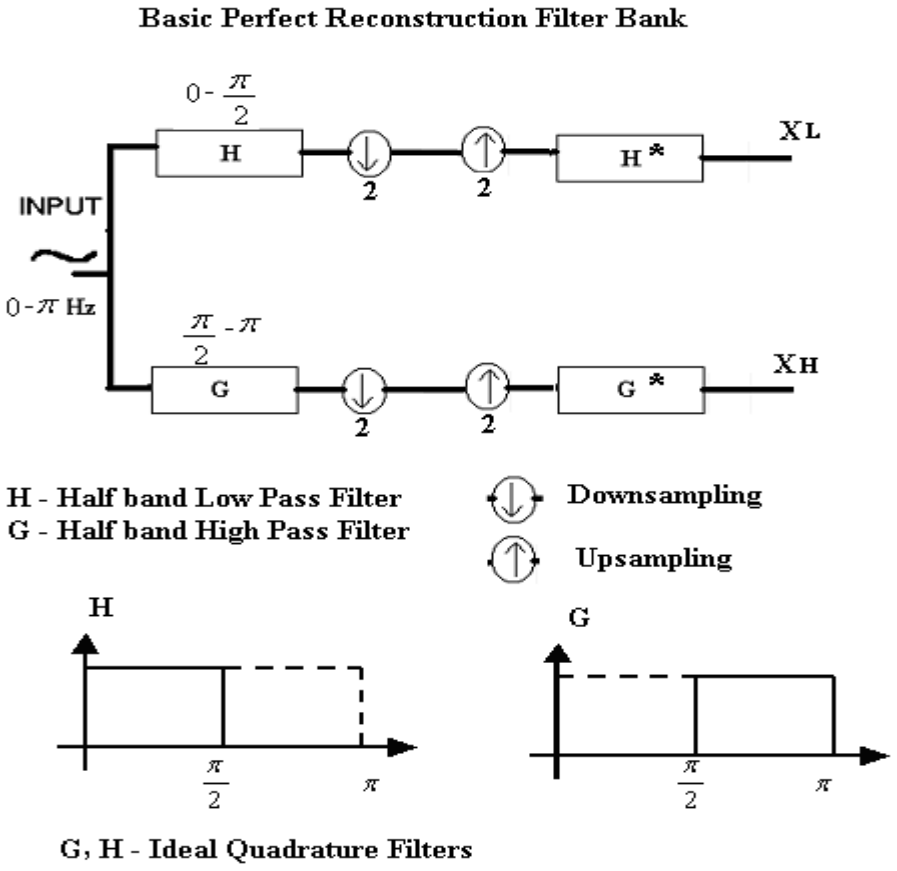


Figure 3 A basic perfect reconstruction filter bank structure showing one levels of decomposition

Figure 3 shows a basic structure of filter bank. Practically the downsampling and upsampling sets to zero every other value of the signal. Effectively it eliminates redundancy. *XL* and *XH* are orthogonal signals. The *H* and *G* are low pass and high pass quadrature filters respectively. A quadrature mirror (QMF) filter can be defined, independently of the dimensionality of the signal space, as a filter that is zero over one half of the Fourier space. The QMF high pass filter coefficients $g(k)$ are related to the low pass filter coefficients $h(k)$ by:

$$g(k) = (-1)^k h(N-k) \qquad \text{- (2)}$$

The frequency response of the H and G filters are shown in figure3. The filters *H** and *G** in the reconstruction path are the conjugates of *H* and *G* filters respectively.

Using the filter bank, it is possible to get narrow bands of frequency in which change will be enhanced. In this study, we found that the low resolution signal acts as noise or unwanted signal which has to be eliminated to get a detail signal that enhances the effect of the fault. This type of filter bank is implemented based on filter bank used for wavelet decomposition [1]. Figure 4 shows a filter bank with second level decomposition, where at each level the signal is passed through high pass and low pass filter to obtain orthogonal decomposition; i.e., the signal's frequency is divided into low frequency components and the high frequency components.

The output signals are all added except the first lowest band signal to obtain detail signal. For the analysis purpose, detail signals are mainly considered as they contain the changes due to fault. In the first level decomposition the detail signal 'XDET' (output of high pass filter 'G') consists of upper half frequencies of the input signal while the low frequency signal 'XL' (output of low pass filter 'H') consists of the lower half frequencies

of the input signal.Both the high pass and low pass filters outputs are continuously analyzed in the next consecutive levels providing more sub-bands of smaller frequencies ranges. The low frequency band with all the additional detail levels with higher frequencies can reconstruct the input signal at each level. At each consecutive level, little more of low-level frequency component is segregated as the detail. The idea is that the change due to fault will be apparent in some detail outputs when decomposed into sufficient levels.

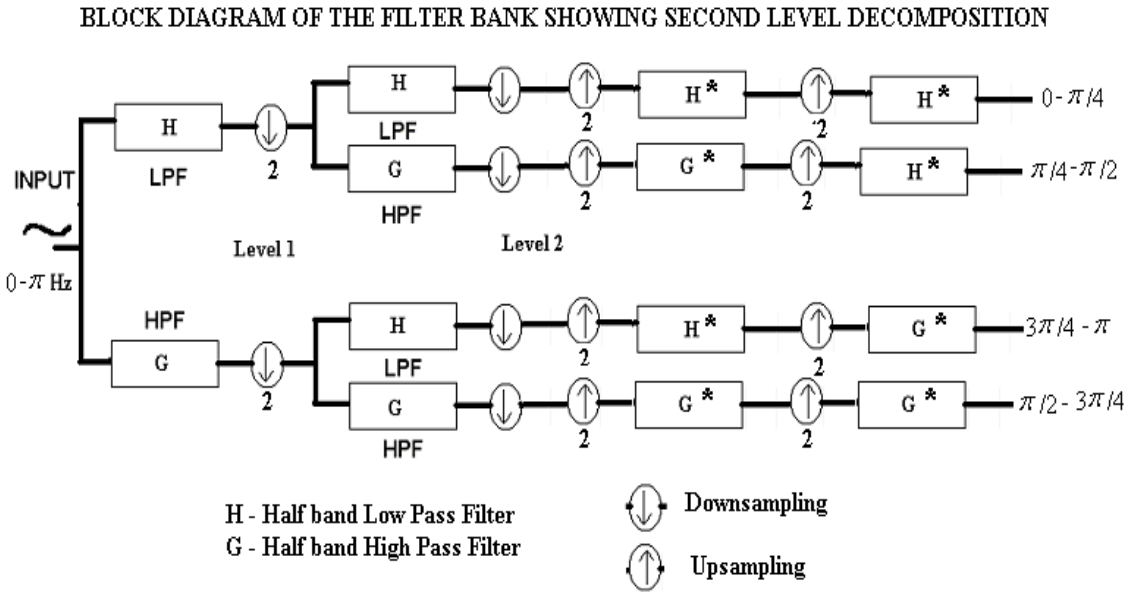


Figure 4 Block diagram of the type of filter bank used in the study showing a second level decomposition

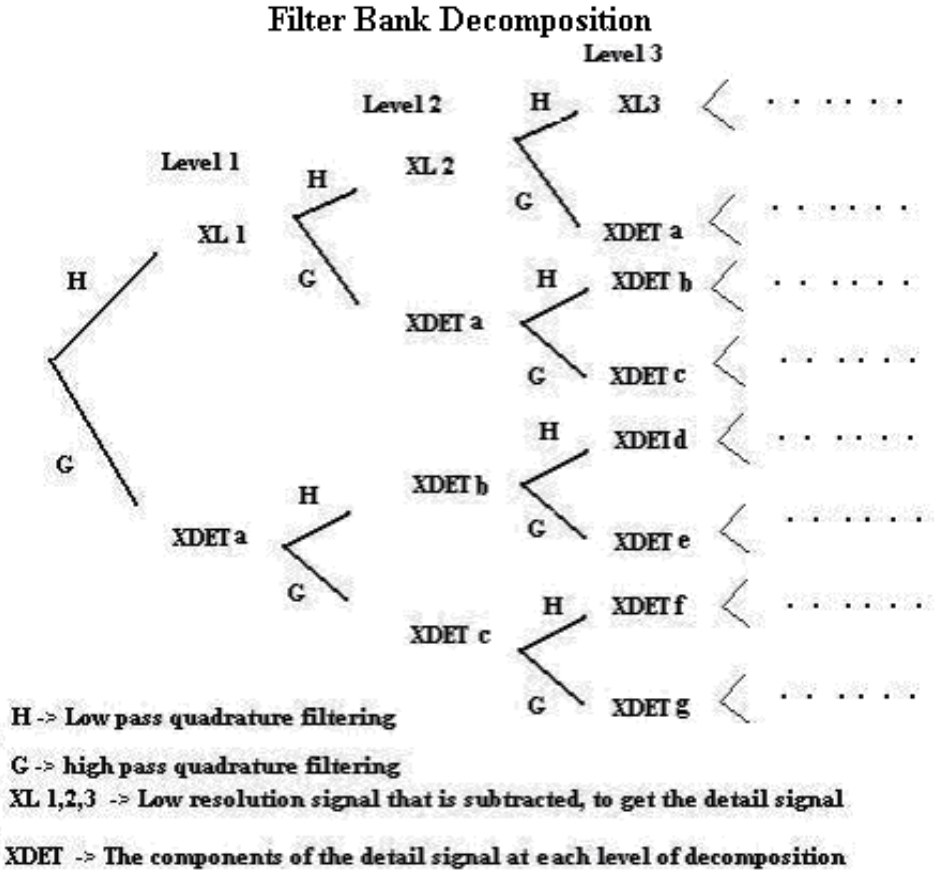


Figure 5 Filter bank decomposition.

Figure 5 shows the diagram for higher level decomposition. At the required level of decomposition, the signal is upsampled and reconstructed as shown in figure 4. For a level two decomposition, signal is passed though filters twice and upsampled and passed though the conjugate filters twice and the signals XDET a, XDET b, and XDET c are added to get the detail signal. Similarly for level three decomposition XDET a through XDET g are added to get a third level detail signal for this research. If the lower level detail signals do not clearly show the changes, we go through next higher levels of decomposition and hence more frequency bands are considered as part of the detail signal.

2.1 Selecting Decomposition Level

The selection of the right level of filter decomposition and its detail components which enhances the change due to fault is very important. Figure 6 shows detail signal considered in this research (which is the sum of XDET details at each level) for level 1 to level 4 wavelet decomposition for the simulated signal of F-14 from figure 2.

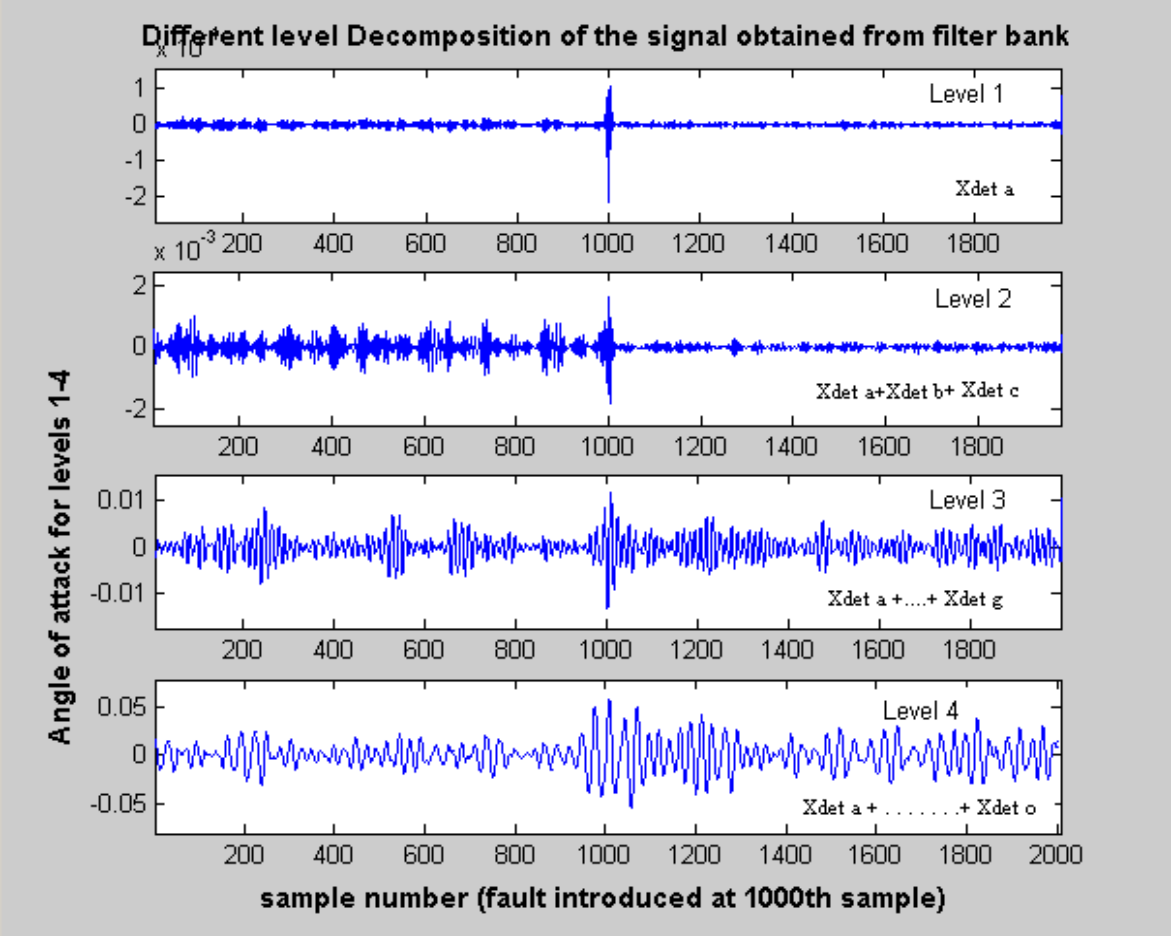


Figure 6 Plot of detail signal of the input signal for the F-14's servo system obtained by filter bank (Level's 1 - 4). Fault introduced from 1000th sample.

The change in the detail signal characteristics for a change from a non-fault condition to fault condition can be used to identify and characterize the condition of the signal. As seen from the above figure, in the second and third level details, a considerable difference in signal characteristics between fault and non-fault conditions can be observed. These two level detail signals might be a good candidate to be considered for further characterization. First level detail signal has a glitch at the change but there is no apparent difference between non-faulty and faulty part of the input signal. Besides, the magnitude of the signal is too small. Fourth level detail already has too much low frequency signal that masks the small change due to the fault in the post-fault part of the signal. So fourth level decomposition is a more than the required decomposition for this research for the F-14th signal.

In the second and third level details, the amplitude of the post-fault signal is much less compared to the pre-fault part. This could be due to the increase in time constant which is in turn responsible for reducing the magnitude of the details after the fault, the effect is discussed in detail for the position servo system in Appendix A under Magnitude response of Servo system. This also gives an important information that changes due to fault are not in the highest frequency range but somewhere in the intermediate frequency range.

Figure 7 below shows the filter bank detail signal obtained for first to fourth level decompositions for the position servo system. From this figure one can say third and fourth level details do not show much difference between pre-fault and post-fault detail signal. Level 1 shows difference between pre-fault and post fault signals but the amplitude is very low.

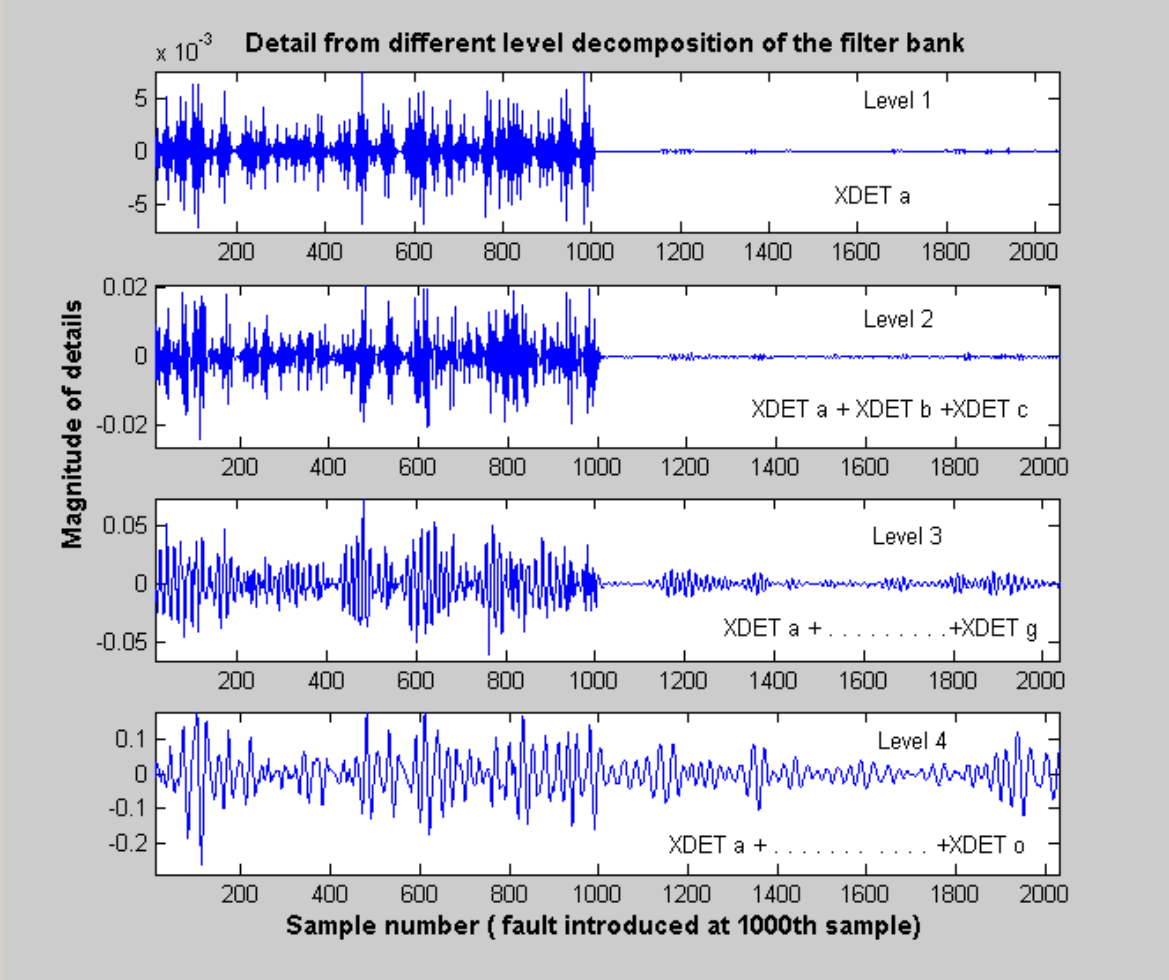


Figure 7 Plot of details of filter bank decomposition of the input signal for the position servo system obtained by filter bank (Level's 1-4). Fault introduced from 250th sample.

As a measure to find the decomposition level that best enhances the difference between no-fault (or pre-fault) and faulty (or post fault) conditions, difference in standard deviation of pre and post fault signals normalized with respect to the non-faulty signal standard deviation can be computed. The mathematical representation of the normalized difference is shown in equation (3).

$$\frac{(\text{Std Deviation})_{pre\text{-}fault} - (\text{Std Deviation})_{post\text{-}fault}}{(\text{Std Deviation})_{pre\text{-}fault}} \quad \text{- (3)}$$

The computation made for the filter bank decomposition levels - two, three and four for various time constants is tabulated in table 1. The faulty time constants ranged from .01 to 100. This gives information as to which detail level is consistent with its difference between pre and post fault signals. The positive values indicate that pre-fault signal has higher power than the post-fault signal and negative values indicate vice versa.

Table 1 The normalized difference in standard deviation of pre and post fault signals for different levels of wavelet filter bank decomposition of position servo system's detail signals for different faulty time constants

Time constant	Level 2	Level 3	Level 4
0.01	-0.1925	-0.1829	-0.1564
0.05	-0.0013	-0.3462	-0.3494
0.1	0.5177	-0.5051	-0.6338
0.5	0.9214	0.5295	-0.2222
1	0.9618	0.7993	0.4468
2	0.9811	0.9069	0.7969
5	0.9925	0.9639	0.9156
10	0.9962	0.9813	0.9383
20	0.9981	0.9889	0.9435
50	0.9992	0.9921	0.9442
75	0.9994	0.9925	0.9441
100	0.9995	0.9926	0.9440

From the above table three observations can be made. Firstly for time constant greater than one, all levels have pre fault signals at higher power level than their post-fault

signals (indicated by a positive number). The time constants at which the difference changes from a positive to negative value is different for all the levels – two, three and four. The level 2's normalized standard deviation difference changes from negative to positive at a time constant greater than 0.05 but for the level 3 it changes for time constants greater than 0.1. Similarly for level 4 the change is at 0.5. The difference stabilizes to a constant value at time constants greater than 1, 5 and 10 for levels 2, 3, and 4 respectively as shown in figure 8 below.

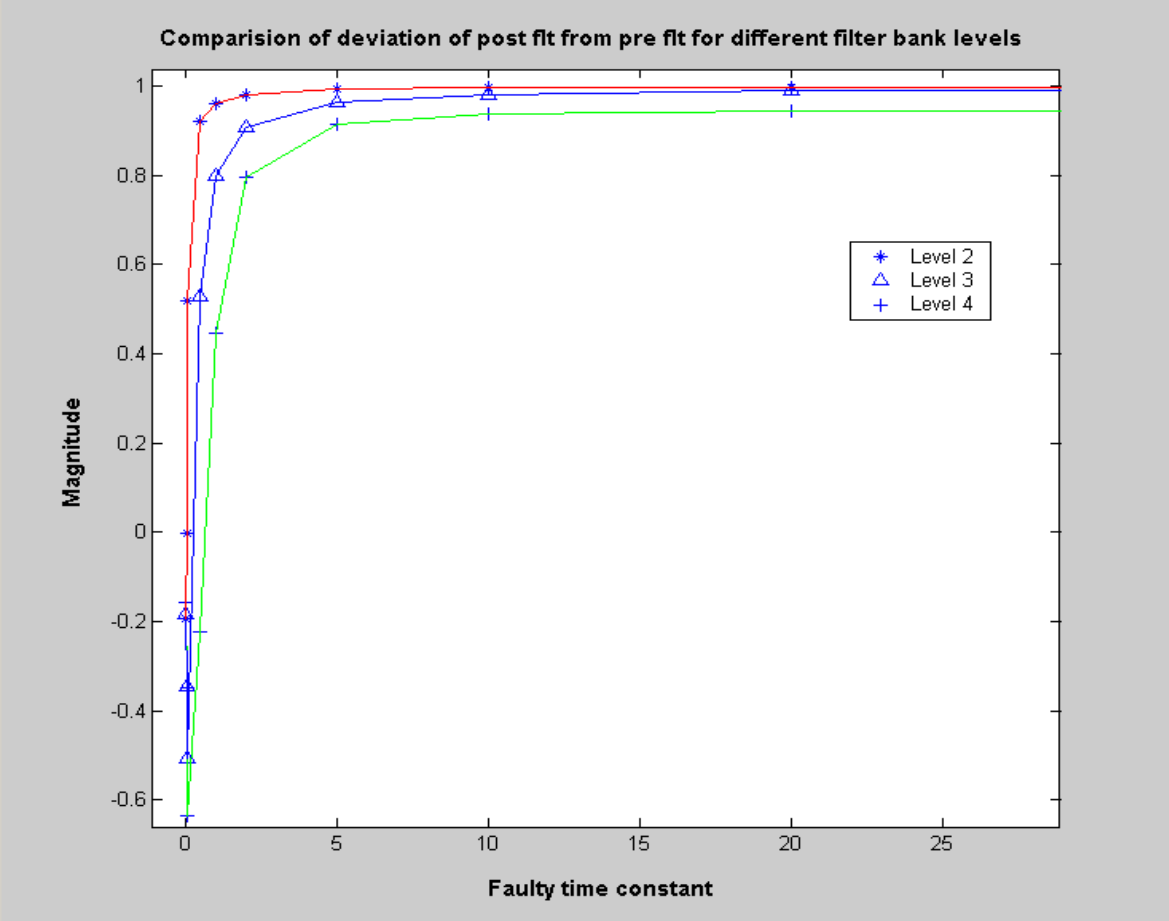


Figure 8 Plot of standard deviation difference (between pre and post fault signals) for levels 2,3, and 4 and increasing time constants, for the readings in table 1.

The above table and plot give important information about choosing the best the filter bank level that provides enhanced information about the change due to fault. Only level two is consistent in the power level of pre and post-fault signal difference for time constants as low as 1 (indicated by positive values for all time constants greater than one). For levels three and four, the difference in standard deviation is not flat till the time constant reaches a higher value. This shows the level two output of the wavelet decomposition could give a better enhancement of change in the signal for no-fault and faulty signal analysis for lower time constants.

The above described section gives the procedure for selecting the right decomposition level in any fault detection case. In particular for the considered example fault, level two is chosen as the best decomposition level for further analysis.

2.2 Creating Indicators

For further analyses two parts of the signals are obtained, faulty and non-faulty after processing from the filter bank. Indicators are created pertaining to pre-fault and post fault condition as discussed in this sections. Signatures are then created for faulty and non-faulty indicators as described in the next chapter. This research focuses on analyzing the signal for detection and identification of normal condition and one possible fault condition (which is change in the time constant of the stick movement).

Indicators are created by applying Short time Fourier transform (STFT). STFT is defined as shown in equation (4).

$$X(\omega,\tau) = \int_{-\infty}^{\infty} x(t).w(t-\tau).e^{-j\omega t}\,dt \qquad \text{- (4)}$$

where τ is the time shift of the window

w is the window function

STFT creates a time-frequency energy distribution. It is a windowed Fourier transform whose short time window helps to identify the time location of the fault. Indicators are the magnitude of STFT given time and frequency energy distribution and are expected to change due to a fault.

To get the discrete samples, Discrete Fourier Transform (DFT) is applied to the window data. DFT is defined as:

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k).e^{-jk2\pi n/N} \qquad -(5)$$

where N is the number of data points

The Discrete Fourier Transform (DFT) operates on finite length (length N) sequences [11]. In this research, the Discrete Fourier transform (DFT) is applied on the level 2 detail signal by choosing fixed number of samples (N) in powers of 2. The process is continued on the next consecutive window of N points and the DFT is applied again. There are two issues however in applying STFT. Firstly, the shape of the window which if rectangle creates abrupt discontinuities at the ends and secondly, the implicit assumption for DFT that the signal is periodic and hence an error due to periodicity. Various window shapes have been proposed to overcome these issues [12]. For this research, no window function was used(implying a rectangular window function), as the incoming data itself is constantly changing. To reduce the waiting time to apply the window, overlapping windows can be considered as shown in figure 9.

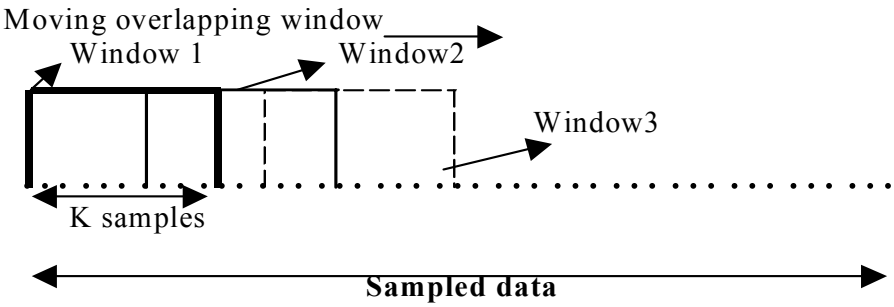
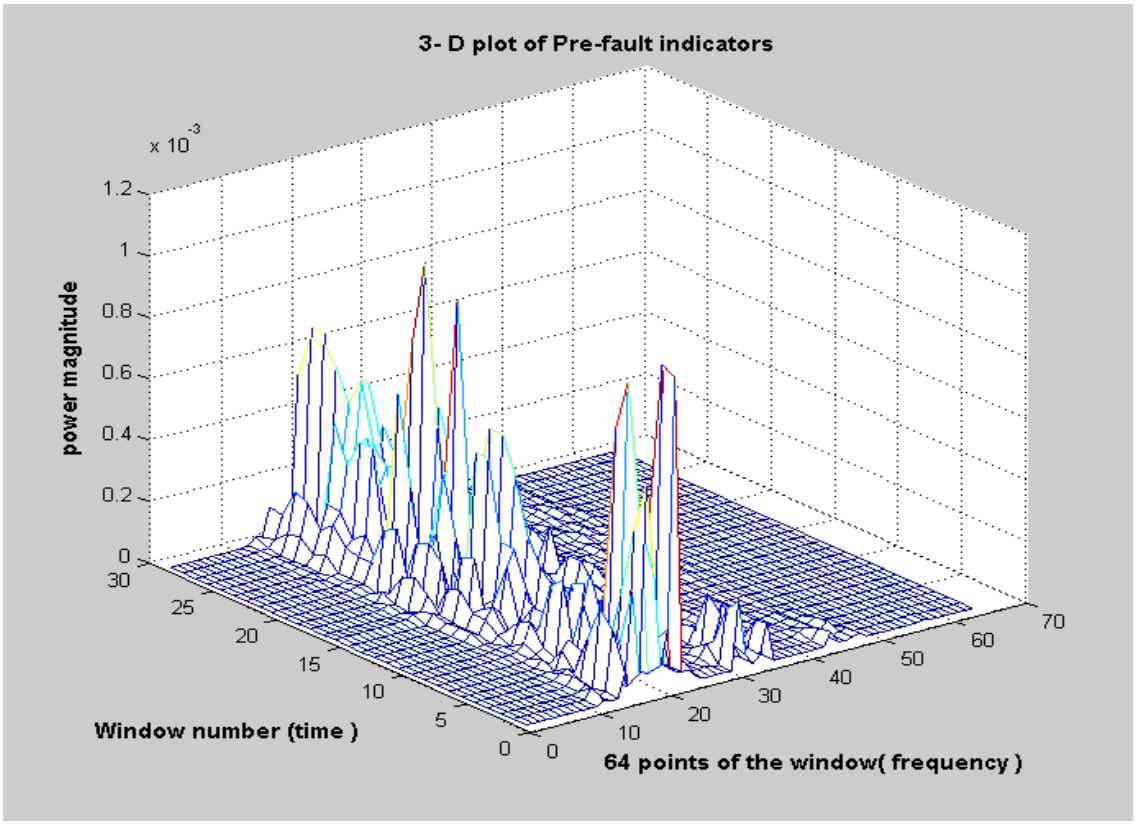


Figure 9 Overlapping windows to compute Short time Fourier transform

In this research a 128-point window with a 75% overlap is considered and the detail signals were analyzed by applying DFT which gives us a 64-point Fourier spectrum. The 64-point power content vector of the detail signal in each window is computed by squaring the magnitude of the frequency spectrum for each frequency point. This power content vector of each window forms an indicator for that window. Repeating the above for the next window obtained by sliding the window by 32 points, the indicators for each window are created. The indicators so obtained are stored as columns of a matrix. Two such matrices are formed, one containing indicators obtained from the processing of pre-fault signal and the other obtained from the post-fault signal. Figure 10 shows the 3-D plot of the pre-fault and post-fault indicator matrix, which localizes the time frequency information of the signal. Next chapter discusses the clustering algorithm and the formation of signatures from these indicators. This study intends to quantify the quality of indicators in each of the following three categories: missed faults, false indications of faults and correct fault detection.



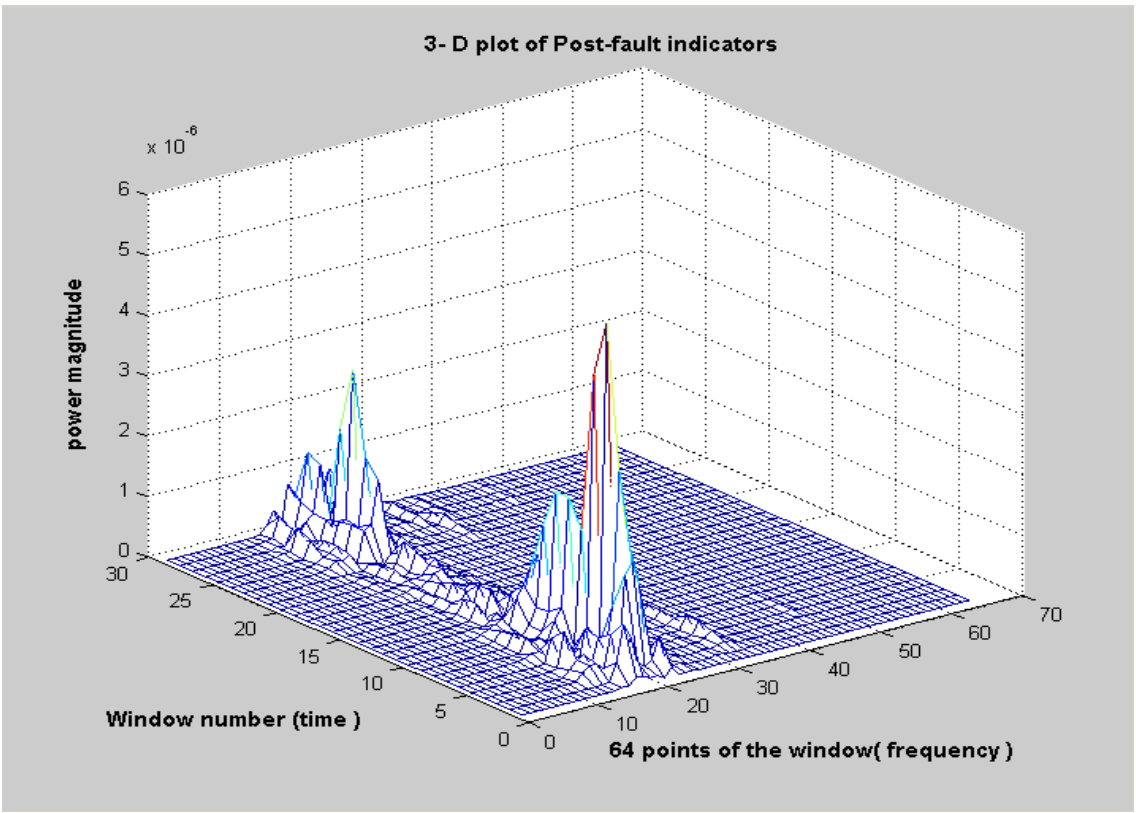


Figure 10 3-D plot of the Pre-fault and post-fault indicator matrices

CHAPTER 3

SIGNATURES AND CLUSTERING TECHNIQUES

This chapter discusses about the clustering techniques to create signatures from the indicators. Clustering involves grouping data points together according to some measure of similarity [13]. The goal is to develop a compact representation of a data set by creating a set of models that represent it. This research uses clustering technique to group the data into faulty and non-faulty cases. In general, there are two general types of clustering: supervised and unsupervised clustering. Supervised clustering uses the pre-defined classes generated from a set of example data to classify other data. This can be called as classification and here the task is to learn to assign instances to pre-defined classes [14]. Unsupervised clustering, on the other hand, tries to discover the natural groupings inside a data set without any input from a trainer. A typical unsupervised clustering algorithm needs the number of classes it should find as input (see Appendix B for a simple example).

In this research, supervised clustering based on spectral energy density is used since we have prior knowledge that the incoming data should be clustered in either faulty or non-faulty classes. This is based on the premise that when the signal changes due to occurrence of fault, the energy distribution of the signal is altered. These changes in the signal are enhanced using the filter bank and then applying STFT to create indicators as described in the previous chapter. The indicators obtained are then segregated using clustering technique described below.

Clustering is carried out in two phases, training phase and classification phase. In the training phase for the supervised clustering case, the representatives of the pre-defined clusters are formed from the example data – indicator matrices created after signal processing the simulated signal. The representatives which represent the clusters are called "signatures". In the classification phase, the distance of the indicators to the signatures created is calculated and these indicators are grouped to the cluster represented by the closest signature. In this research, the same indicators used to create the signature are also used in the classification phase to check and validate the clustering effectiveness. This research examined two techniques for signature creation.

In the first technique, for the training phase, the signatures for faulty and non-faulty cases are the mean vectors of the indicator vectors obtained for each window. The mean of pre-fault indicators is the signature for non-faulty class and the mean of post-fault indicators is the signature for faulty class. In the classification phase, to analyze the performance of this clustering technique, the same pre-fault and post fault indicators were classified into either faulty or non-faulty class depending on their vector distance to the above pre-determined signatures. The indicator is classified to the group to which the indicator's distance to the group's signature is minimum. Figure 1 shows the plot of pre-fault and post-fault indicators that were created for a position servo system with faulty time constant of one. STFT was applied to this 2020-sampled 2^{nd} level detail signal (first 1000 samples are pre-fault samples and the last 1000 samples are faulty samples with the middle 20 samples not considered as it is the transition of the signal from pre-fault to post-fault condition).

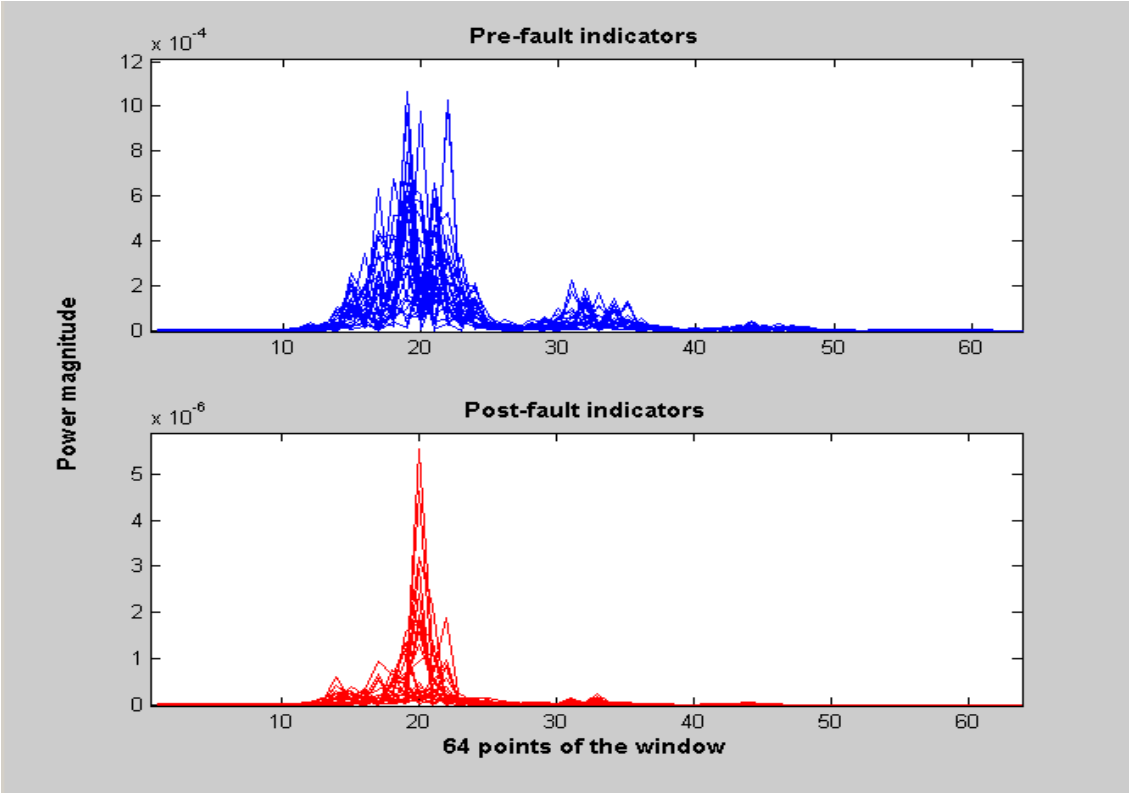


Figure 1 Plot of pre-fault and post-fault indicators created for the position servo system's level 2 detail signal with post-fault time constant 1.0

It is clear from the figure that the spectral energy of the pre-fault indicators is higher that the post-fault indicators. The above clustering technique was applied to pre and post-fault indicators shown in figure 1 separately. Results showed post-fault indicators were nearer to post-fault signature (mean of indicator vectors for this clustering technique) than the pre-fault signature. However around 86 percent of pre-fault indicators were nearer to pre-fault signature while the rest were nearer to post-fault signature indicating a 14 percent of false alarms. The same algorithm applied to the data from F-14 aircraft's aileron-actuator also showed similar results.

The same clustering technique was also applied to another type of indicators, a variation where indicators were created choosing only the most sensitive frequency bands

(details with considerable magnitude of the filter bank output). Looking at the plot of the indicators in figure 10 and 11, it can be observed that the spectrum is prominent for only a small set of frequencies (from points 10 – 30) and is almost small elsewhere. The filter bank detail outputs that have considerable magnitude values only were considered to form detail signal to be processed by the STFT method to form the indicators. The clustering technique is applied to find the pre-fault and post-fault indicator's distance to the signatures and the indicators are clustered. This method did not show any considerable improvements over the previous method.

In the second clustering technique, clustering is based on Principal Component Analysis (PCA) and vector subspace signature concept. Here the signatures are vector subspaces instead of the mean in the previous case. The singular value decomposition is the factorization of a given (*m*,*n*) matrix A, as:

$$A = U.S.V^{*} \qquad -(6)$$

where U is an (m,m) orthogonal matrix, V is an (n,n) orthogonal matrix and S is an (m,n) diagonal matrix with real, non-negative elements $\sigma_i[i = 1,\ldots\ldots\ldots,\min(m,n)]$ in descending order:

$$\sigma_1 \geq \sigma_2 \geq \ldots\ldots\ldots\ldots \geq \sigma_{\min(m,n)} \geq 0 \qquad -(7)$$

The σ_i are the *singular values* of A and the first $\min(m,n)$ columns are the left and right *singular vectors* of A. S has the form:

$$\begin{bmatrix} \sum \\ 0 \end{bmatrix} \text{ if } m \geq n \text{ and } \begin{bmatrix} \sum & 0 \end{bmatrix} \text{ if } m < n \qquad -(8)$$

where $\sum$ is a diagonal matrix with the diagonal elements $(\sigma_1, \sigma_2, \cdots\cdots, \sigma_{\min(m.n)})$.

We assume now $m \geq n$. If $r = rank(A) \leq n$, then

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots\cdots \geq \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_n = 0 \qquad \text{- (9)}$$

If $\sigma_r \neq 0$ and $\sigma_{r+1} = 0$, then *r* is the rank of *A*. In this case, *S* becomes an (*r*,*r*) matrix, and *U* and *V* shrink accordingly. SVD can thus be used for rank determination.

$$A = \sum_{i=1}^{r} \boldsymbol{u_i} \ \sigma_i \ \boldsymbol{v_i}^* \qquad \text{- (10)}$$

$\boldsymbol{u_i}$ and $\boldsymbol{v_i}$ are columns af matrix **U** and **V** respectively.

The indicator matrix can be approximated by the summation of products shown in the equation (10). If there is one significant singular value, the matrix can be approximated by a single product, where *i = 1* and the product is a 'rank one' matrix.

In our case, A is the indicator matrix, where each row is energy distribution for a given window and is used as the distribution at the time corresponding to the center of the window. Hence the A matrix row number can be compared to time.

If we can write

$$A \cong \sum_{i=1}^{r} \boldsymbol{u_i} \ \sigma_i \ \boldsymbol{v_i}^* \qquad \text{- (11)}$$

Then every row of A is of the form

$$\boldsymbol{row_k} = \sum_{i=1}^{r} u_{ik} \ \sigma_i \ \boldsymbol{v_i}^* \qquad \text{- (12)}$$

which is equivalent to equation (13)

$$\begin{bmatrix} \\ \\ \\ \mathbf{r_k} \\ \\ \\ \\ \end{bmatrix} = \sigma_1 \begin{bmatrix} \mathbf{u_1} \\ \\ \\ u_{1k} \\ \\ \\ \end{bmatrix} \begin{bmatrix} \\ \mathbf{v_1^*} \end{bmatrix} + \sigma_2 \begin{bmatrix} \mathbf{u_2} \\ \\ \\ u_{2k} \\ \\ \\ \end{bmatrix} \begin{bmatrix} \\ \mathbf{v_2^*} \end{bmatrix} + \ldots\ldots\ldots \text{ r terms}$$

i.e. $\boldsymbol{row_k} \in \text{span } \{ \boldsymbol{v_1}^*, \boldsymbol{v_2}^* \ldots . \boldsymbol{v_r}^* \}$ - (13)

The r orthogonal vectors of the matrix V form the orthonormal basis for an ‘r’ dimensional space [16]. Hence the subspace is defined as the signature subspaces for all the rows. For our case the signature subspace for the energy distribution.

This clustering technique is applied in this research where two signature subspaces, non-faulty subspace and faulty subspace are formed from the pre-fault indicator matrix and post-fault indicator matrix respectively. Figure below shows the singular values obtained from the indicator matrices by applying the singular value decomposition (SVD) function in Matlab for the position servo system.

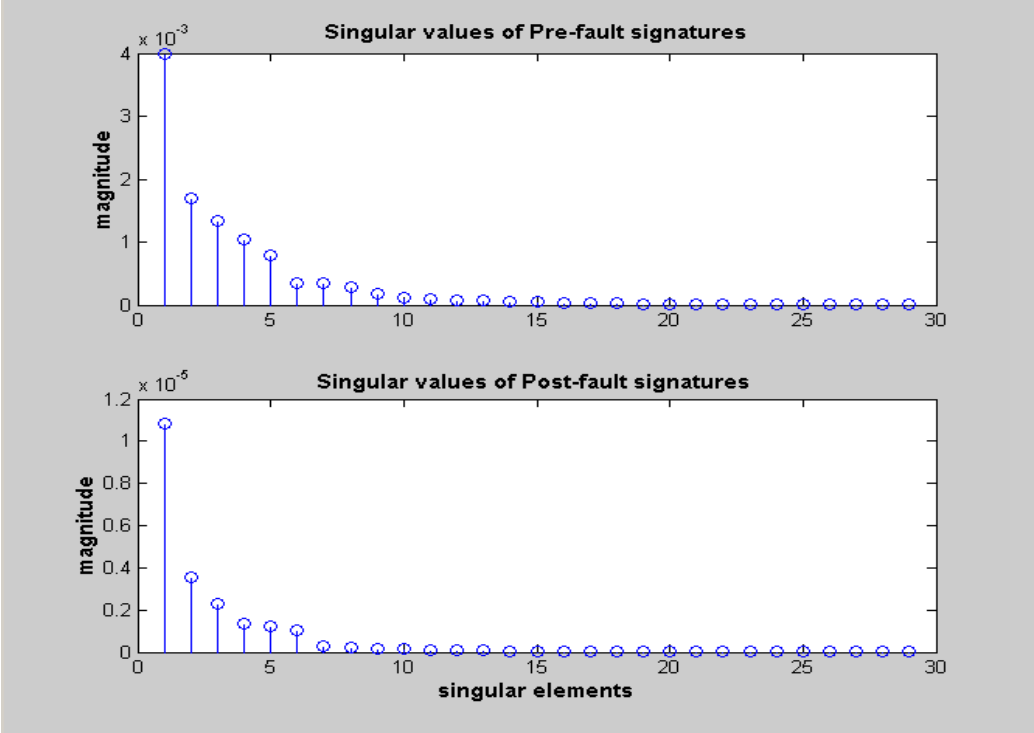


Figure 2 Singular values of pre-fault and post-fault indicator matrix created for the position servo system’s level 2 detail signal with post faulty time constant 1.0

Figure 12 shows that there is more than one significant singular value for both the pre-fault and post-fault indicator matrices. The first value is high but the following values are not negligible. To decide on the number of singular values which best approximates the indicator matrix, the singular values that make up 95 percent of the total

magnitude of all the singular values is chosen. In this research the upper limit on the number of singular values is chosen as four. The chosen V vectors for both the pre-fault and the post-fault cases form the orthonormal basis for the two subspaces, non-faulty and faulty. In the classification phase, to analyze the performance of this clustering technique, the same 29 pre-fault and 29 post-fault indicators of the position servo system are classified. The pre-fault indicators closer to the faulty sub-space, indicates false alarms and the number of post-fault indicators closer to non-faulty subspace indicates the percentage of missed faults. More false alarms means the system is more expensive and more percentage of missed faults means more undetected faults which could be fatal to the system. Figure 3 shows the concept of subspaces (a two dimensional plane is considered for illustration).

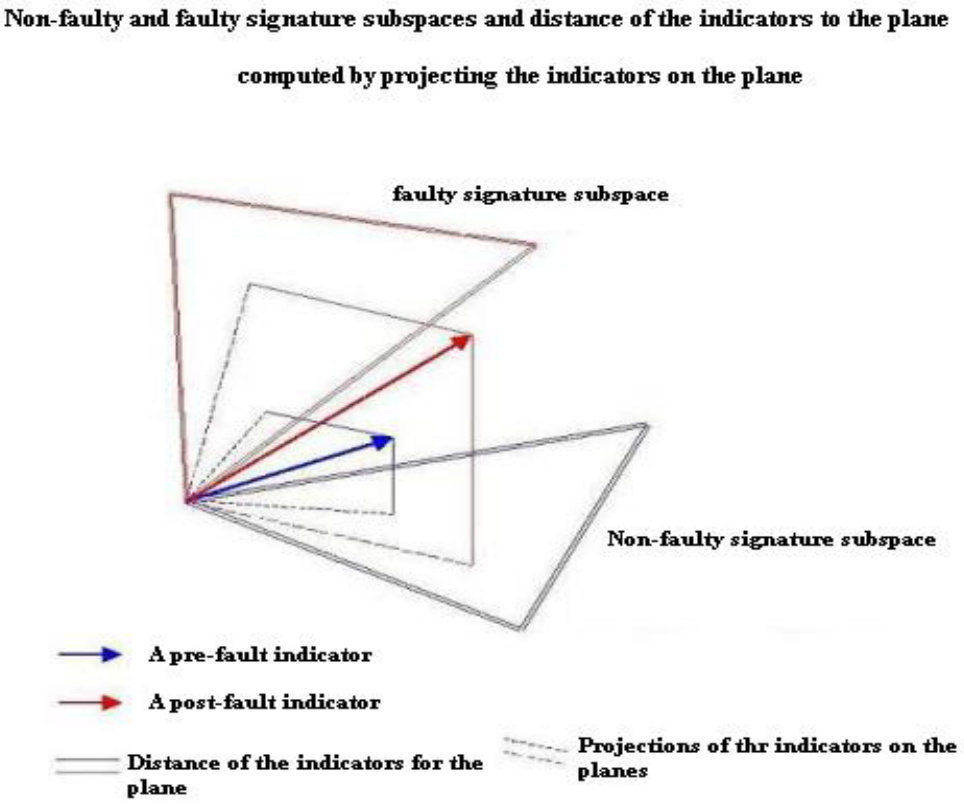


Figure 3 Concept of pre-fault and post-fault subspaces showing the distance of indicator vectors to the subspaces (for simplicity, a 2-D subspace is shown here).

Figure 13 shows two 2-Dimensional sub-spaces, both faulty and non-faulty. To classify the incoming indicator as belonging to non-faulty or faulty class, the distance of the indicator from both the sub-spaces is compared and the closest vector subspace signature identifies the condition of the indicator.

Figure 14 below shows the distance computed for pre-fault indicators and post fault indicators to both the pre-fault and post fault vector sub-space.

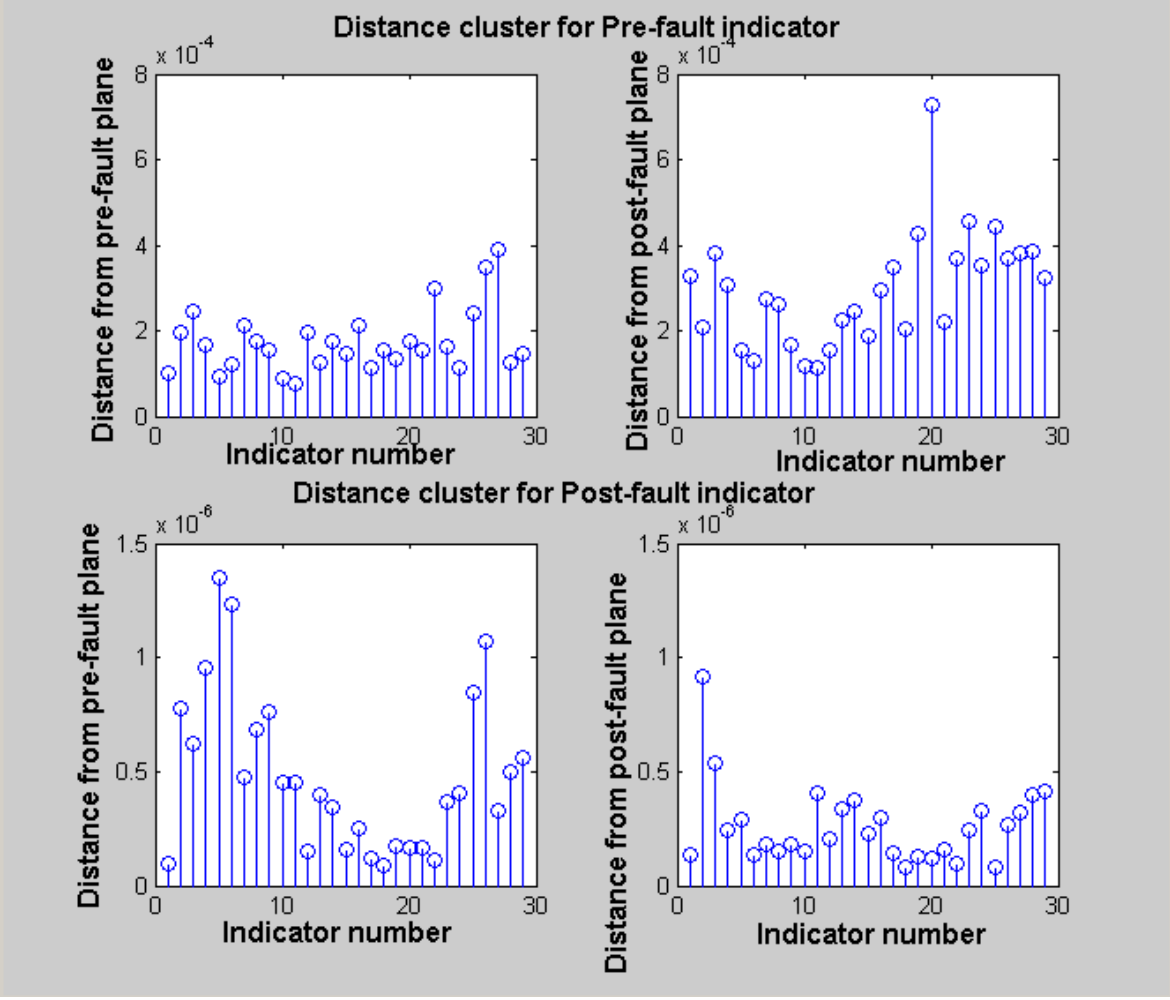


Figure 4 Distances of the indicators (pre-fault and post-fault) from the planes (pre-fault and post-fault) for position servo system.

It can be seen from figure 14, for the position servo system, the pre-fault indicators are closer to the pre-fault sub-space as shown by the smaller distance while their distance

is large for the post-fault sub-space. Similar reasoning with opposite distance holds good for post-fault indicators.

The distance computation is as follows: given the orthonormal basis { $\boldsymbol{v_1}$,$\boldsymbol{v_2}$ .....}, the projection ($\hat{\boldsymbol{p}}$) of the indicator ($\boldsymbol{p}$) onto the subspace is computed as :

$$\hat{\boldsymbol{p}} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \ldots\ldots\ldots\ldots \quad \text{- (14)}$$

since $\boldsymbol{p} - \hat{\boldsymbol{p}} \perp \boldsymbol{v_i}$, $< \boldsymbol{p}_1, \boldsymbol{v}_1 > = < \hat{\boldsymbol{p}}_1, \boldsymbol{v}_1 >$ - (15)

hence $< \boldsymbol{p}_1, \boldsymbol{v}_1 > = \alpha_1$, $< \boldsymbol{p}_2, \boldsymbol{v}_2 > = \alpha_2$ .....

Once the projection is computed, the difference between the indicator vector and its projection on the subspace gives the distance of the indicator to the subspace.

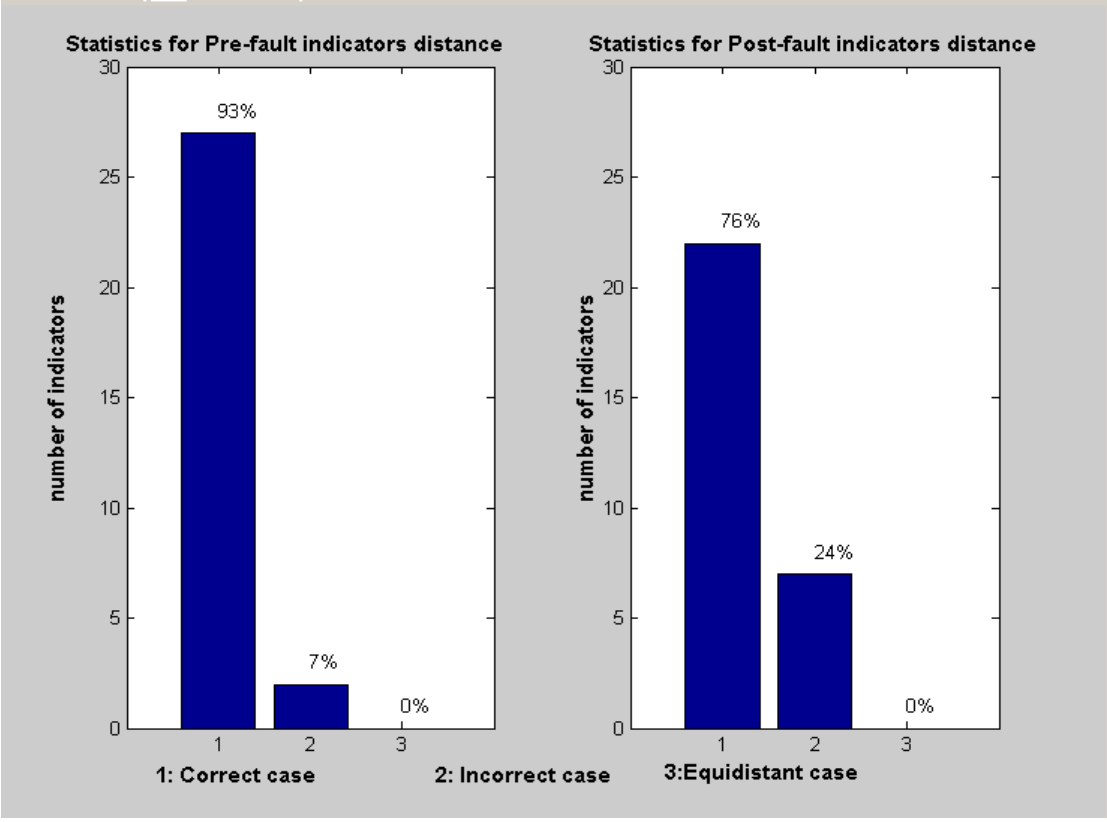


Figure 5 Statistics for the vector subspace classification for pre-fault and post-fault indicators created for position servo system's level 2 detail signal with post faulty time constant 1.0

From the distance plot in figure 14, a histogram for pre-fault and post-fault indicators is computed showing the percentage of indicators correctly classified, percentage of false alarm for pre-fault indicators and the percentage of correctly classified and percentage of missed faults for post-fault indicators. Figure 15 gives the histogram plot of the position servo system signal with faulty time constant 1.0

This method of clustering was applied for signals by varying time constant faults from 1 to 100 for the position servo system and the results were promising. The results are tabulated in table 2.

Table 1 Results of clustering technique tabulated, showing the percentage of indicators that were correctly classified (for pre and post fault indicators) for various faulty time constants of the servo system.

Faulty time constant	Percentage of pre-fault indicators nearer to pre-fault plane	Percentage of post-fault indicators nearer to post-fault plane
1	93	76
2	90	76
5	90	76
10	90	76
20	90	72
50	93	76
75	97	69
100	97	69

Form the tabulated results, it can be observed that the pre-fault indicators more than 90% are classified correctly and for post-fault indicators the classification is greater than

69% for all time constants. This percentage figure was fairly constant for other simulated conditions (random noise simulated using different seeds).

In this second variation of indicator creation, only the sensitive frequency components of the detail signal were considered to form signatures. In each level (L) of decomposition of the filter bank, the detail signal is constructed with only details with significant magnitude. The indicators formed by performing STFT on this detail signal. The difference of the pre and post-fault signal for each of the components showed that certain detail components (components 1 and 3 in our research example) always had the highest positive difference. So these detail components 1 and 3 were added together and considered as input signals to create indicators applying STFT method. These indicators were then tested using the clustering techniques as described in this chapter. The statistics of the indicators showed no improvement over the previously discussed example where all the detail components of the input signal were considered, so this method of creating indicators using selected detail components was not pursued further.

The above clustering techniques showed that for the simulated position servo system signal, they are able to detect and identify the faults successfully. The same technique is applied to the F-14's aileron-actuator example and studied in the next chapter.

CHAPTER 4

AN APPLICATION OF BLIND FAULT DETECTION

This chapter explains a step-by-step procedure of the detection and identification analysis for the signal obtained for a simulated F-14 aircraft's aileron-actuator system using the SVD and vector sub-space method. This signal is simulated by the Matlab script (Appendix C - program listing 1) with a time constant 0.05 for the normal system and in this example, a faulty time constant value of 1.0 is chosen. The noise seed used is 31412. Figure 16 shows the simulated signal with the fault applied at the 1000th data point.

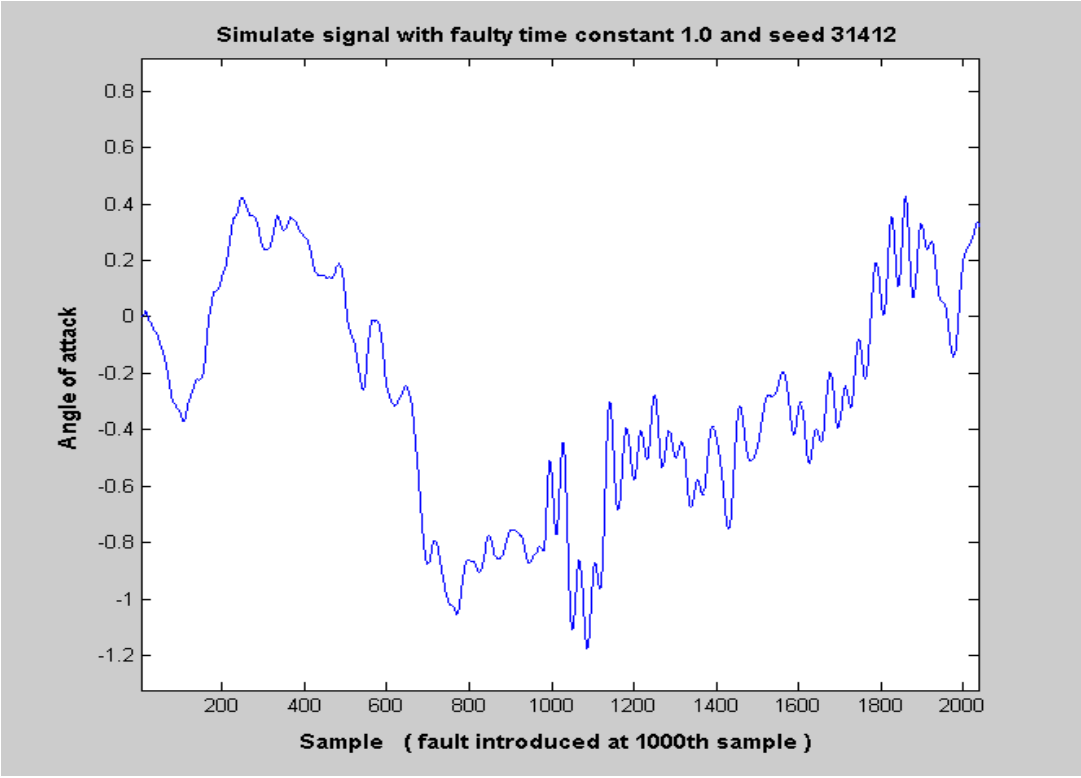


Figure 1 F-14 simulated signal. Fault introduced at 1000th sample, faulty time constant being 1.0

4.1 Filter Bank Decomposition To Get Detail Signal

The first step is to obtain the detail signal. The signal in figure 16 is processed by a filter bank (script in Appendix C – program listing 3) to obtain second level decomposition detail signal of the filter bank. Figure 17 below shows the detail signal

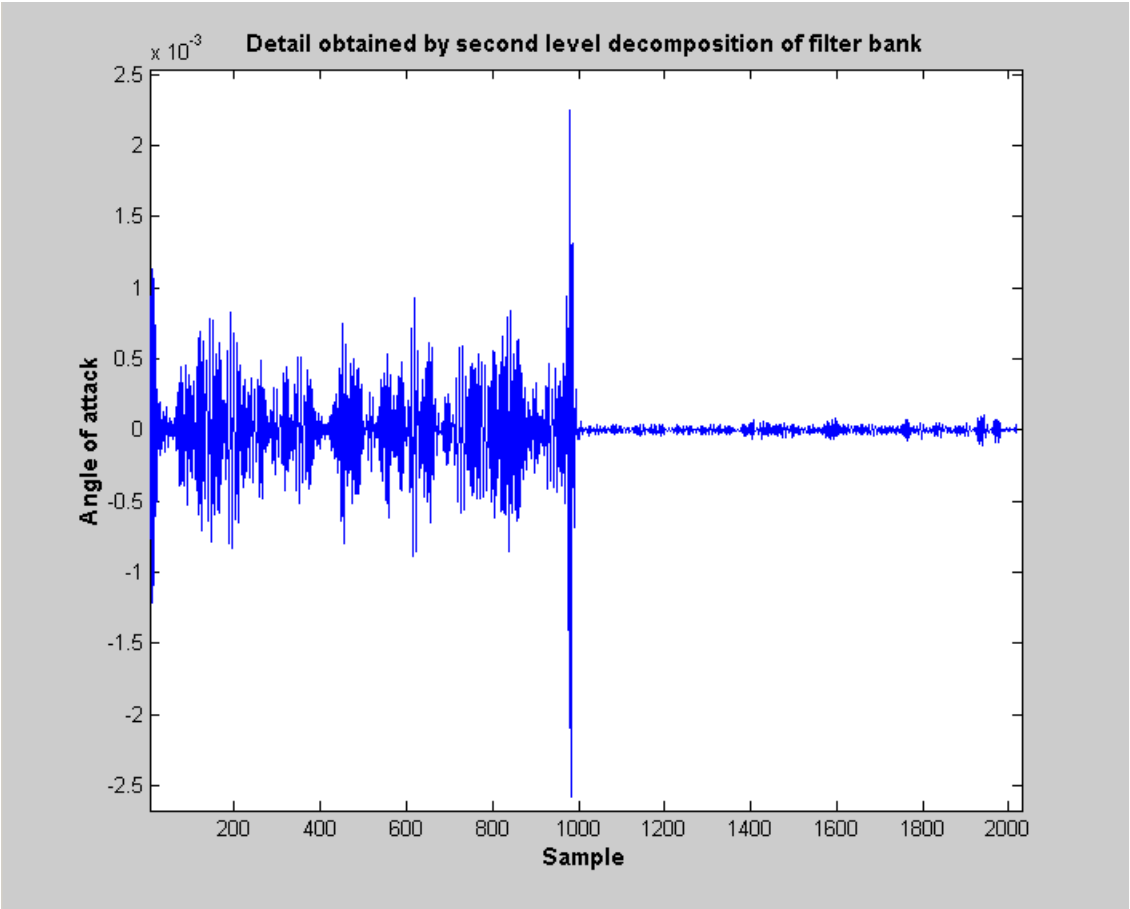


Figure 2 Detail signal obtained from second level decomposition of filter bank

4.2 Creation Of Indicators

First 1000 samples of the detail signal of figure 17 forms the pre-fault signal and the samples from 1020 to end are taken as post-fault signal. Short time Fourier transform is applied to the pre-fault and post-fault signals separately to obtain two matrices - pre-fault indicator matrix and post-fault indicator matrix (Matlab script in Appendix C – program

listing 4). Window size of 128 was chosen in this example and with an overlap of 96 points. As the signal itself is very random it was decided not to use any window shape. A 128 point FFT was applied to each window and only 64 points of the FFT are taken (as FFT gives a reflection so the second half is just the mirror image of first half) and the magnitude is squared to get the power content. Figure 18 shows 29 pre-fault indicators and 29 post-fault indicators obtained after performing a STFT on the input signal.

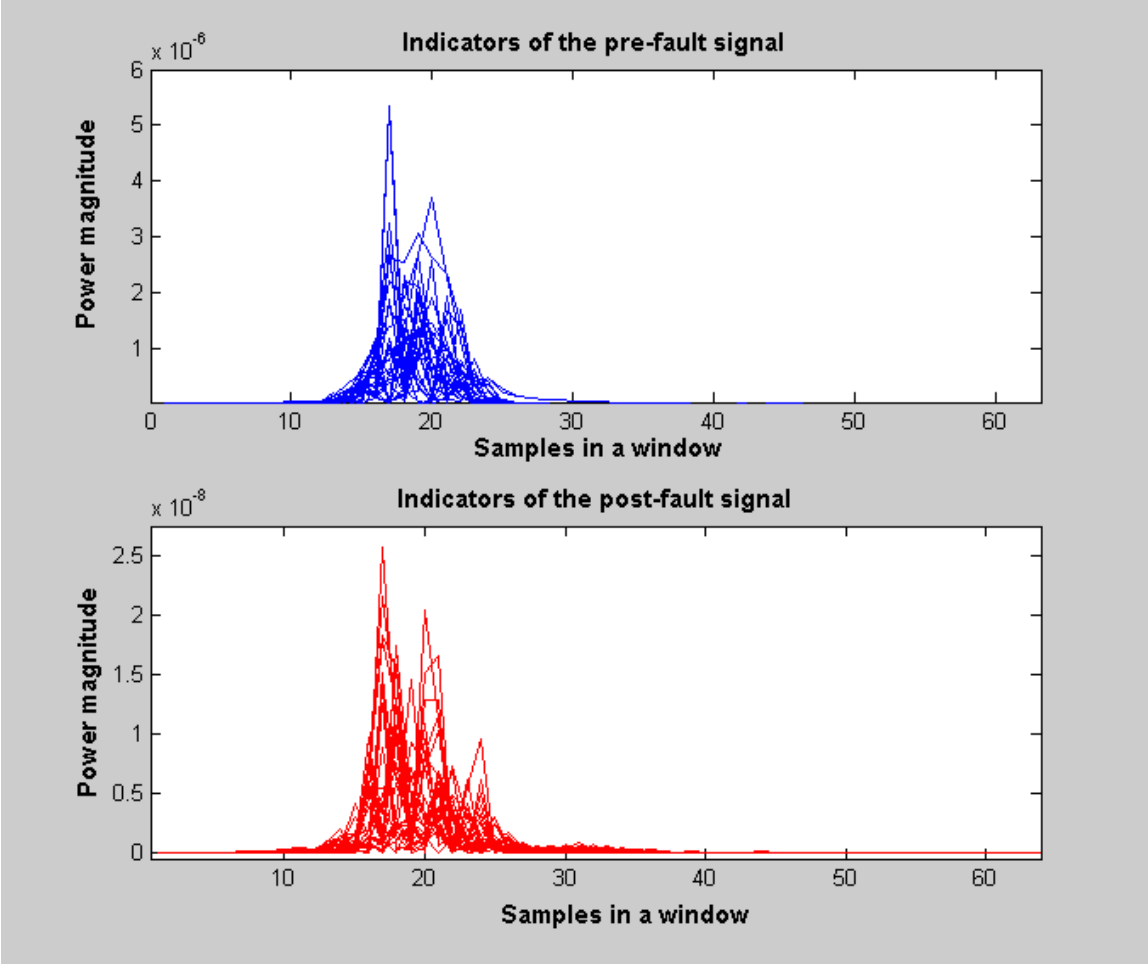


Figure 3 Indicators created for the pre-fault signal and post-fault signal

4.3 Singular Value Decomposition Of The Indicator Matrices

Singular value decomposition of pre-fault indicators matrix and post fault indicators matrix gives the singular values for the respective matrices (script in Appendix C – program listing 5). Figure 19 shows the singular values obtained for both the pre-fault and post-fault matrices.

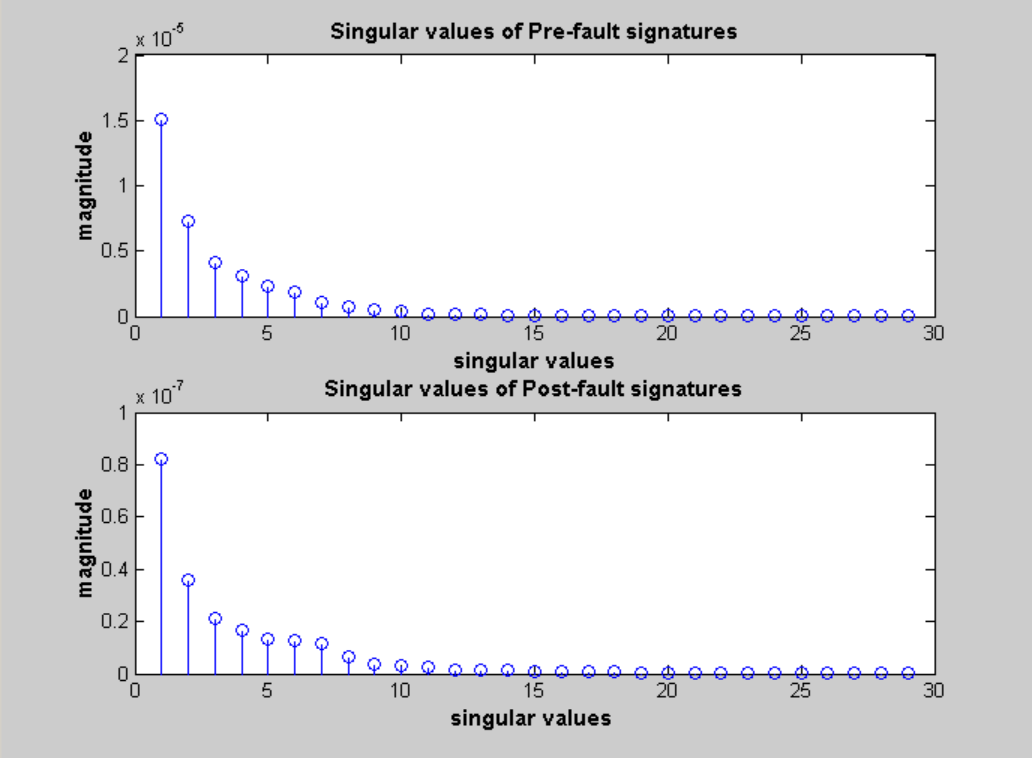


Figure 4 Singular values of the pre-fault and post-fault indicator matrices

There is more than one significant singular value as seen in the figure. Number of singular values that can best approximate the system is chosen by computing the percentage of singular values which constitute more than 95 percent of the total magnitude of the singular values. The upper limit on the number of singular values is chosen as four in our research. In this example, the number of significant singular values was computed to be four. The four column vectors of matrix V for both the pre-fault and post-fault cases form

the two sub-spaces non-faulty and faulty. Angle between the pre-fault and post-fault subspaces computed gave a separation of 8.9 degrees (script included in program listing 5 – Appendix C).

4.4 Classification Of The Indicators

In the next step, i.e., the classification phase, the pre-fault and post-fault indicators are classified using the vector sub-space clustering technique. Then the distance of pre-fault and post-fault indicators is computed separately. Distance from both the non-faulty subspace and faulty subspace is computed (script in Appendix C- program listing 5) by projecting the indicators on the plane. Figure 20 below shows these four distances (distance of pre-fault and post-fault indicators from both non-faulty and faulty subspace).

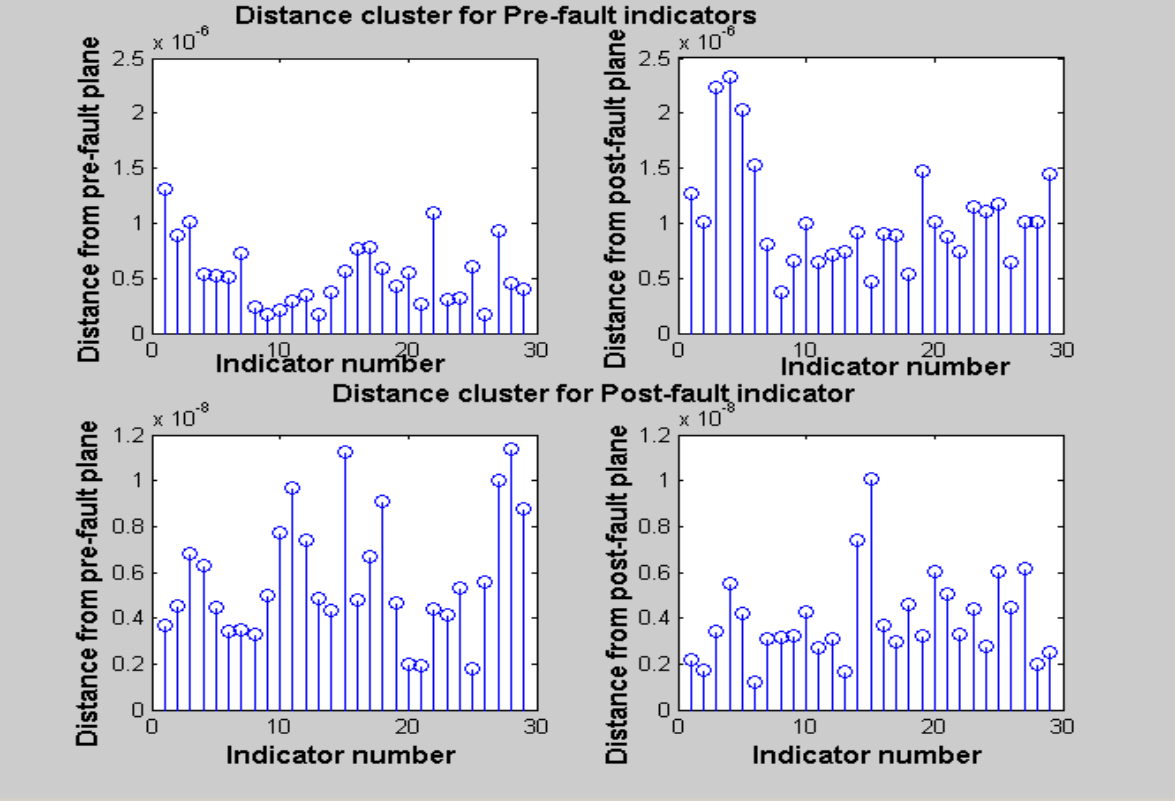


Figure 5 Distance of the indicators (pre-fault and post-fault) from the two subspaces, non-faulty and faulty

The purpose of classifying these indicators is to quantify the performance of this technique. Once the distances are computed, the condition of the indicator is identified with the condition of the closest sub-space signature (script in Appendix C – program listing 6). The percentage of pre-fault indicators closer to faulty subspace plane gives the percentage of false alarms and the percentage of post-fault indicators nearer to non-faulty sub-space plane gives the percentage of missed faults. A histogram indicating the percentage of indicators correctly classified and incorrectly classified for both pre-fault and post-fault indicators computed separately is shown in the figure 21.

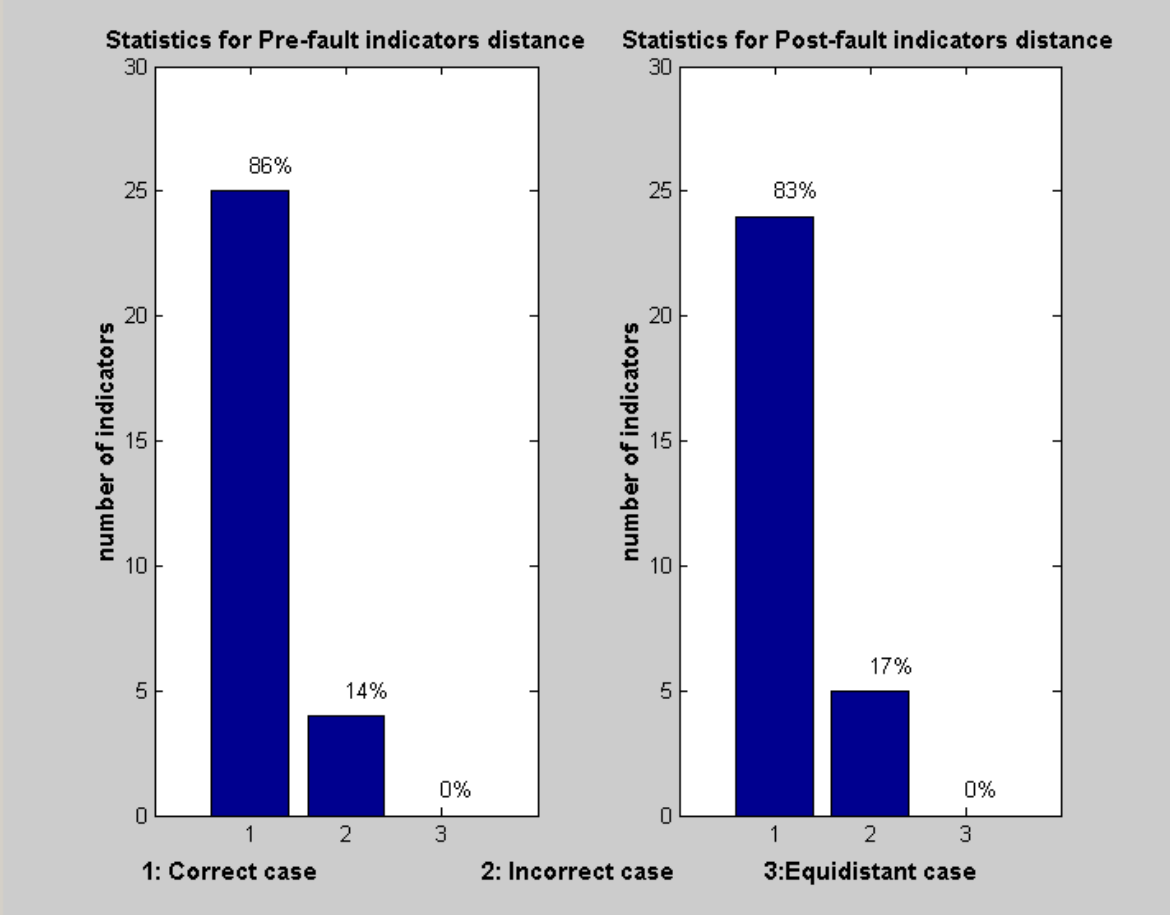


Figure 6 Histogram showing the percentage of indicators (pre-fault and post-fault) correctly identified and incorrectly identified. Equidistant case is where the indicators are exactly in the middle of the subspaces, equidistant from both subspaces

The histogram shows that 86 percent of pre-fault indicators are classified correctly and 14 percent triggered false alarms. In case of post-fault indicators, 17 percent is missed the fault and 83 percent are correctly classified.

Table 1 Results of clustering technique tabulated, showing the percentage of indicators that were correctly classified (for pre and post fault indicators) for various faulty time constants for the F-14.

Faulty time constant	Percentage of pre-fault indicators nearer to pre-fault plane	Percentage of post-fault indicators nearer to post-fault plane
0.1	76	79
1	86	83
2	79	86
5	79	86
10	76	86
20	76	72
50	76	90
75	76	90
100	76	90

This example of applying the blind fault detection and identification to the F-14's aileron-actuator servo system clearly shows that this method has a potential of becoming one of the powerful tools in signal processing.

CONCLUSION

This research studies blind fault detection using spectral analysis. This research developed signatures based on spectral energy density for two conditions, normal and faulty condition where the fault was change in time constant of the actuator. The signals were simulated using two test systems : F-14 jet fighter's aileron-actuator system and a position servo system. The fault – change in time constant was introduced to the simulated signals at a known time.

The simulated signal was processed by an orthogonal filter bank to enhance the effect of change due to fault. Level 2 decomposition's high frequency (normalized frequency range $[\frac{\pi}{4}, \pi]$) which were found to have better enhancement of these changes was used to form the detail signal. A 128 point Short time Fourier transform was applied to create a 64 point indicators which are energy (squared magnitude) - frequency distributions of the signal.

Two types of signatures were created and a clustering technique based on distance from the signatures were used to classify and identify these indicators as non-faulty and faulty conditions. In the first technique, the mean of the indicators was calculated for both pre-fault and post-fault cases, which formed the signatures and the shortest distance of the indicators to the means classified them accordingly. The results showed more than eighty percent of the indicators were classified correctly. The second technique is based on Singular value decomposition (SVD) and vector subspace signature method , where SVD is performed on the two pre-fault and post-fault indicator matrices and the principal

singular values are calculated. The corresponding columns vectors of matrix V form the two subspaces pertaining to normal condition and faulty conditions.

The indicator is then classified as a faulty or a non-faulty one depending on their distance to vector sub-spaces. The sub-space to closest to the indicator identifies the condition of the indicator. This was tested for the same faulty and non-faulty indicators to check the efficiency of the technique. The statistics showed that more than 86% of non-faulty and faulty indicators were classified correctly for the position servo system. This test was carried on for various magnitudes of the fault by changing the time constant and the results were found to be consistent.

The technique using SVD and vector sub-space signature was applied to the F-14's aileron-actuator system and the results were found to be good.

These results prove that the blind fault detection method is a promising tool for fault detection. Further research on this method can be done by applying this method to find signatures for other types of faults and also this method can be tested for online application.

REFERENCES

[1] J. L. Aravena, and F. Chowdhury, "Fault Detection of Flight Critical Systems," 20th Digital Avionic Systems, October 2001, Daytona Beach.

[2] Won-Yong Lee, Cheol Park, and George E. Kelly, "Fault Detection in Air-Handling

[3] Unit Using Residual and Recursive Parameter Identification Methods, ASHRAE Transactions V.102, Pt. 1,1996.

[4] Chen Xiangrong, Chen Zhaolin and Zhu Jiandong, "Design of Robust Detection and Isolation observers for Singular Systems," *Proceedings of the 15th IFAC World Congress on Automatic Control*, , Barcelona,july 2002.

[5] B.K. Walker, "Fault Tolerant Control System Reliability and Performance Prediction Using Semi-Markov Models," in Patton and Chen (eds.), Preprints of SAFEPROCESS'97 (Hull, UK), IFAC, pp. 1056-1067, August 1997.

[6] Hwang, B., Saif, M., and Jamshidi, M., A Neural Network Based Fault Detection and Identification (FDI) for a Pressurized Water Reactor, *Proceedings of the 12th IFAC World Congress on Automatic Control*, Sydney, Australia, 1993.

[7] Jia-Zhou He, Zhi-Hua Zhou, Xu-Ri Yin, and Shi-Fu Chen, "Using Neural Networks for Fault Diagnosis, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, 2000, vol.5, pp.217-220.

[8] Mariusz Pawlak, Jan M. Koscielny, and Michaeł Z. Bartys, " Application of Fuzzy Neural Networks for Instrument Fault diagnosis of Condensation turbine control," *Proceedings of the 15th IFAC World Congress on Automatic Control*, , Barcelona,july 2002.

[9] J. L. Aravena, "Detecting change using Pseudo Power Signatures," *Proceedings of the 15th IFAC World Congress on Automatic Control*, , Barcelona,july 2002.

[10] http://www.airbornemagazine.com.au/artF14.htm

[11] http://www.jetplanes.co.uk/f14.html

[12] http://cnmat.cnmat.berkeley.edu/~alan/MS-html/MSv2.html#RTFToC1

[13] ftp://ftp.ece.lsu.edu/pub/aravena/ee7000FDI/SpectralAnalysis/Windows4FFT.pdf

[14] http://www.palantir.swarthmore.edu/loicz/help/clustering.htm

[15] ‘ Clustering – Connections and statistical language processing’ , Frank Keller, University of Saarlandes

[16] http://ikpe1101.ikp.kfa-juelich.de/briefbook_data_analysis/node265.html

[17] http://ccrma-www.stanford.edu/~jos/mdft/Gram_Schmidt_Orthogonalization.html

APPENDIX A : SIMULINK DIAGRAMS

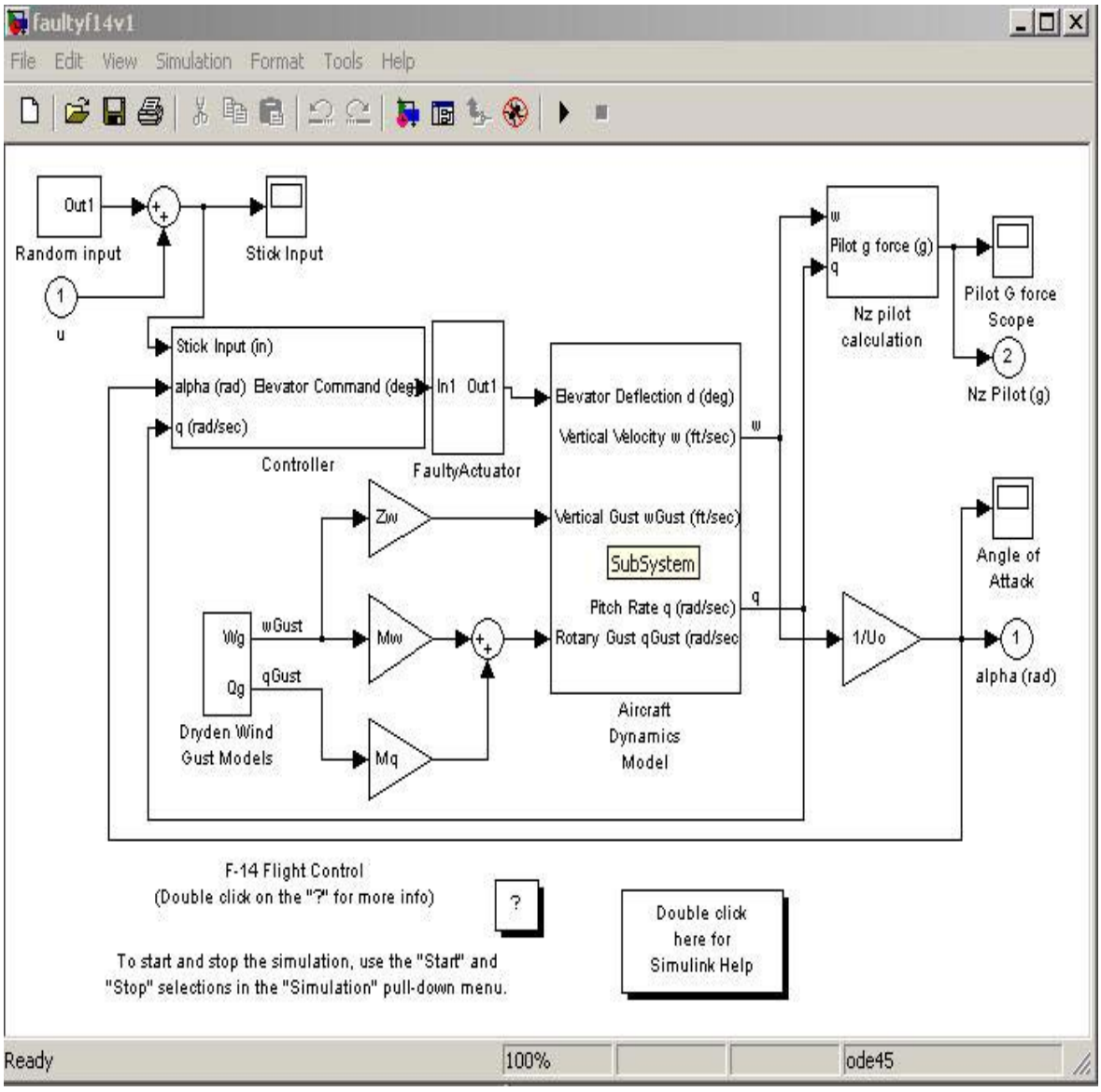


Figure 1: Simulink diagram of F-14 jet fighter showing the actuator, where fault is introduced.

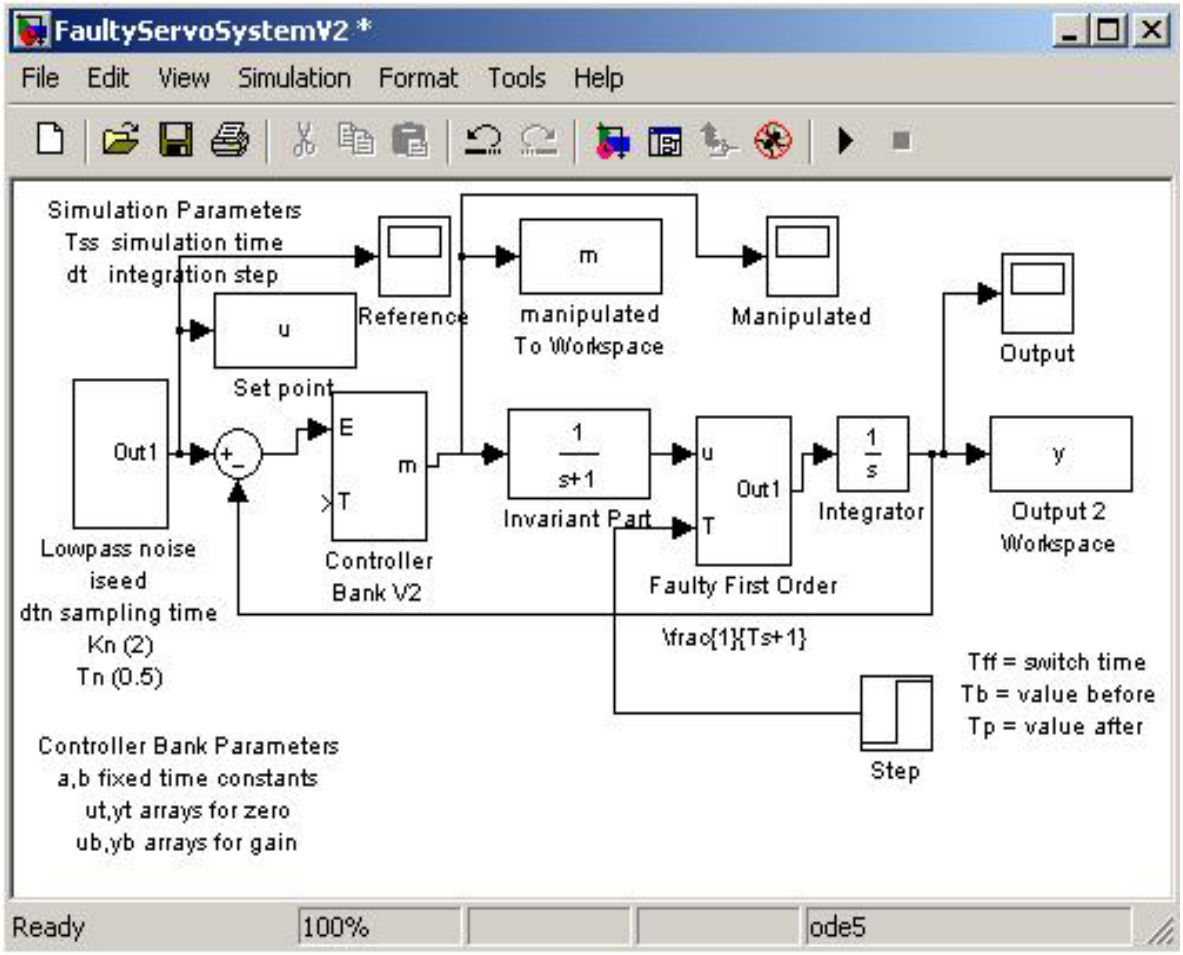


Figure 2 Servo system simulink diagram showing the faulty first order where the fault is introduced

Actuators of F-14 and servo system where fault is introduced

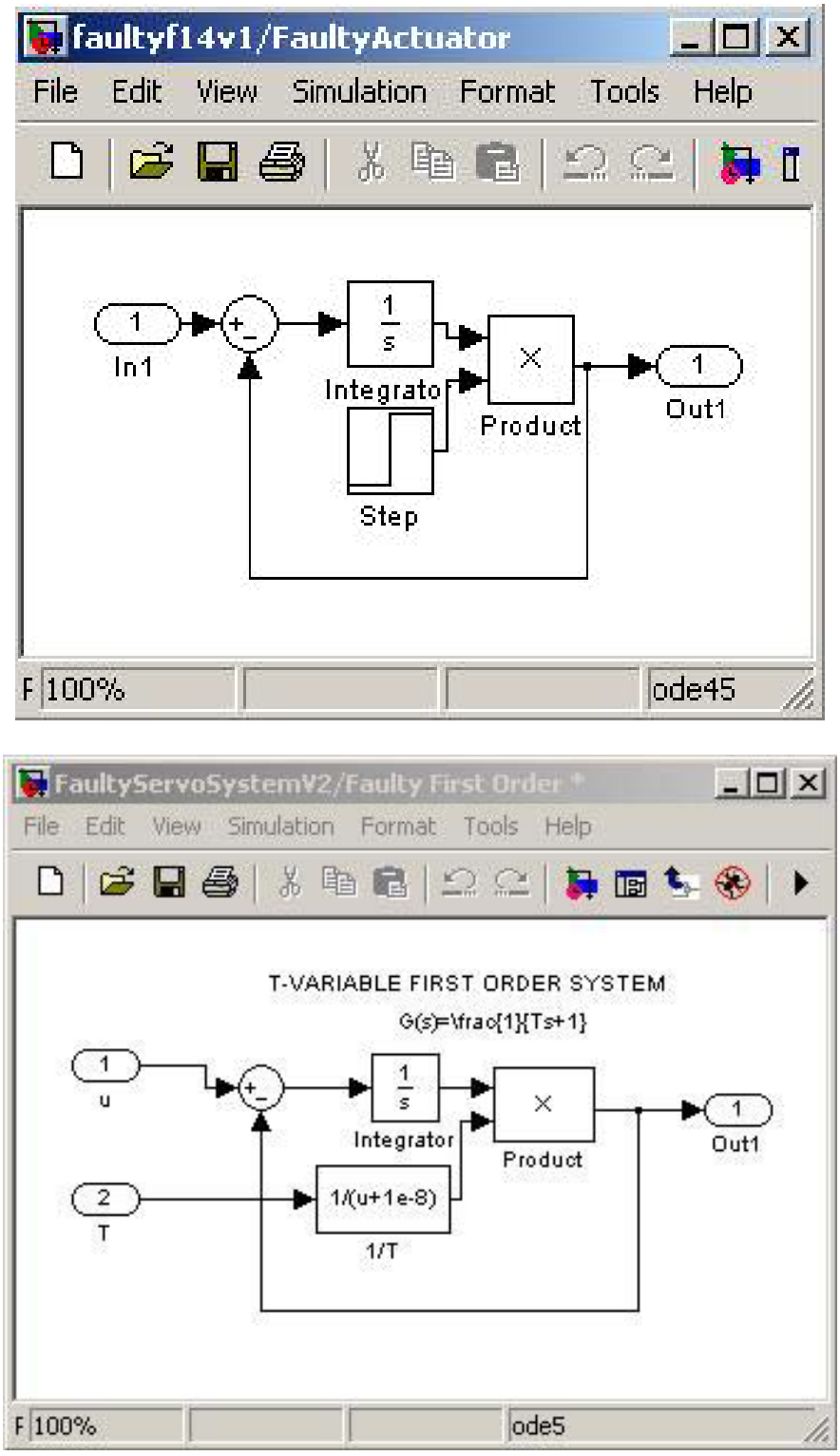


Figure 3: Faulty actuator whose time constant is controlled to introduce faults. On top is F-14 jet fighter actuator on bottom is that of position servo system

Magnitude Response for Servo system for various time constants of the system

One of the test systems used for the study was position servo system. This is a simple system and the model of the system is known. During the study, it was found that post fault signal has lesser power magnitude as compared to the pre-fault signal. In this section magnitude response of the servo system is studied for increasing time constants (which is the fault in this case). This study helps us to understand the observation of the research and illustrates the advantages of having a model for the system.

The transfer function of the servo system is

$$Sys = \frac{1}{s.(s+1).(tp.s+1)}$$

Where tp, is the time constant of the system.

Figure 25 shows the magnitude response of the servo system (one of the test systems used in our research) for various time constants of the system.

The magnitude response in figure 25 explains the decrease in post-fault power level with increase in the faulty time constant (which is tp in the above equation). In our case, the sampling frequency is 50Hz. So the filter bank frequency division for signal components in different levels would be $\frac{50}{2^l}$ where l, is the level of decomposition. The frequency range of the low level signal in each level is in the range (0 - $\frac{50}{2^l}$)Hz and therefore the frequency range of the detail signal considered will be in the frequency range ($\frac{50}{2^l}$ - 50)Hz.

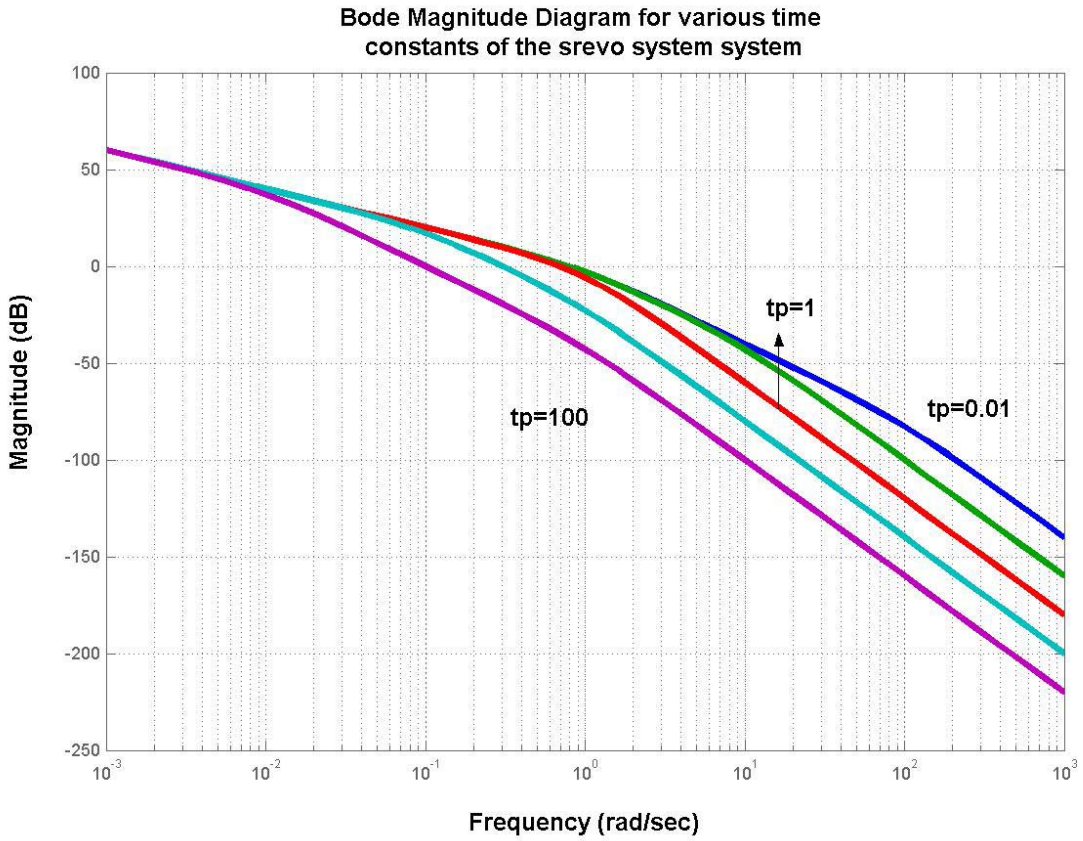


Figure 4: Magnitude response of the servo system for various time constants (0.01, 0.1, 1.0, 10, 100)

For example, for 4th level decomposition, the low level signal component is the frequency range 0 – (50/16) ~ 3 =(2*pi*f) 18.8 rad/sec. The detail signal is the signal in the frequency range (18.8 – 314) rad/sec (50Hz = 314 ra/sec). Similarly for level 2 decomposition the frequency range of the detail signal is (78.85 – 314) rad/sec.

Looking at the magnitude plot, it is apparent that the magnitude for time constant 100 is much lower than magnitude at time constant 1.0 and less for frequencies greater than 18 rad/sec. The conclusion is , as the time constant (i.e. fault) increases, the magnitude of the faulty detail signal reduces and so the post-fault signal has lesser and lesser power magnitude.

APPENDIX B : EXAMPLE FOR SUPERVISED AND UNSUPERVISED CLUSTERING

consider a set of colored balls (all colors) that you want to classify into three groups: red, green, and blue. A logical way to do this is to pick out one example of each class--a red ball, a green ball, and a blue ball--and set them each next to a bucket. Then go through the remaining balls, compare each ball to the three examples and put each ball in the bucket whose example it matches the best.

This example of supervised clustering is illustrative because there are two potential problems. First, the result you get is going to be dependent upon the balls you select as examples. If you were to select a red, an orange, and a blue ball, then it might be difficult to classify a green ball. Second, unless you are careful about selecting examples, you may select examples that don't represent the distribution of data. For example, you might select red, green, and blue balls, only to discover that most of the colored balls were cyan, purple, and magenta (which are in between the other 3 primary colors). This shows the importance of selecting representative samples when you execute supervised clustering.

In the unsupervised case, considering the same colored balls case, this would be like dumping them into an automatic sorting machine and telling it to create three piles. The goal of unsupervised clustering is to create three piles where the balls within each pile are very similar, but the piles are different from one another. Here no pre-defined classification is required. The task is to learn a classification from the data.

APPENDIX C : MATLAB PROGRAMS

Program Listing 1

To run the F-14 simulation, specifies simulation time, fault time, time constant

```
%chkflight.m
% runs a simulation and asks for seed
% It does the psd and SVD analysys to the details obtained
%by running simulation
% will have to run faultyf14v1 first and close it and then
%run this program
%% seed  defines random signal applied to stick
%       deafults to a random selection
% details  array with details to be analyzed
% sfft     function that computes stft with rectangular
%window
% svd      computes svd for signatures(row wise) and does
%svd analysis

Tsim=120; %simulation time
Tf=49; % fault time
Ta=0.05; %prefault value
Ta2=input('enter value for Ta2 :'); %post fault value
noiseS=input('enter seed [return for random choice] ');
if isempty(noiseS)
    s=clock;
    seed0=100*s(4)*s(5);
    disp(['using seed = ' num2str(seed0)])
    noiseS=seed0;
end
sim('faultyf14v1');
aa=awf2(:,2);
L=input('enter the detail level: ');
details=get_details(aa,L);
details=details(1:2000);
```

Program Listing 2

For simulation of servo system

```
%Define parameters for faults system simulation
%%%Simulation Parameters
Tss=30; %simulation time
dt=1e-5; %fixed step
%%%Random Reference
iseed=input('enter seed : ');
dtn=0.05; %noise update sampling period
Kn=2; %gain for noise generator
Tn=0.5; %noise low pass filter time constant
%%%Controller bank parameters
% defines inputs and outputs for all quantizers required to
%implement
% a bank of controllers of the form
% K(s)=c\frac{(s+1)(s+1/T)}{(as+1)(bs+1)}
% the values of T,c are given in tables
% a,b are constants
a=1.087e-3;
b=1.087e-4;
i=1:10;
T1=[0.1*i-0.09];
i=[11:20];
T2=10*(i-10);
T=[0.01;T1(:);T2(:)]; %array with 21 values
c1=[6;64;123;182;240;299;358;416;475;533];
c2=[5861;11723;17584;23445;29307;35168;41029;46891;52752;586
13];
c=[5.85;c1;c2];
[ub,yb]=pcode(T,c); % gain table
[ut,yt]=pcode(T,1./T); %table for zero location
%%% FDI delay
Tfdi = 100*dt; %FDI delay
ibuffz=1024; %initial buffer size for delay simulation
%%% Fault Definition
Tff=10; %fault time
Tb = 0.01; %pre fault value
Tp = 1; %post fault value
%%%%%%%%%%
```

Program Listing 3

Filter bank to get the detail signal

```
%get_details.m
%created Sep 7, 2001
%this routine uses a filter bank approach to create a low
resolution level
%and a details component for a signal x
%use [xdet,xlow,xcomp]=do_details(x,L)
% x  vector signal
% L  level of the low resolution
% xdet    details
% xlow    low resolution view
% xcomp    other orthogonal components of level L
function [xdet,xlow,xcomp,tagss]=do_details(x,L)
filters=[
   2.6670058e-02   1.3264203e-05
   1.8817680e-01   9.3588670e-05
   5.2720119e-01   1.1646686e-04
   6.8845904e-01  -6.8585669e-04
   2.8117234e-01  -1.9924053e-03
  -2.4984642e-01   1.3953517e-03
  -1.9594627e-01   1.0733175e-02
   1.2736934e-01   3.6065536e-03
   9.3057365e-02  -3.3212674e-02
  -7.1394147e-02  -2.9457537e-02
  -2.9457537e-02   7.1394147e-02
   3.3212674e-02   9.3057365e-02
   3.6065536e-03  -1.2736934e-01
  -1.0733175e-02  -1.9594627e-01
   1.3953517e-03   2.4984642e-01
   1.9924053e-03   2.8117234e-01
  -6.8585669e-04  -6.8845904e-01
  -1.1646686e-04   5.2720119e-01
   9.3588670e-05  -1.8817680e-01
  -1.3264203e-05   2.6670058e-02
];
h=filters(:,1);
g=filters(:,2);
if nargin==1 |isempty(L)
   L=4; %default setting
end
%xww=w_orthoN([x(:);flipud(x(:))],L,h,g);     %compute     the
components
```

```
[xww,xcc,lxcc,tagss]=w_orthoN([x(:);flipud(x(:))],L,h,g);

xlow=xww(1:length(x),end+1-2^L); % low resolution of level L
xdet=x-xlow;
xcomp=xww(1:length(x),end+2-2^L:end);
return

%    w_orthoN.m
% computes all the orthogonal components of an array X.
% if X has more than one column it will compute orthogonal %components

% for each column.
% keeps all components packed in arrays, xcc,xww, in the
%workspace
%USE [xww,xcc,lxcc,tagss]=w_ortho(x,L,h,g)
%    x    input data array
%    L    maximum decomposition level
%    h    basic wavelet filter to be used
%    g    high pass component of filter
%         name_k01,..., name_knn   are components of level k
%    xcc  array with compressed components
%    lxcc array of L elements with lenght of compressed
%components
%    xww  array with orthogonal components. Each the same
%length as x
%         2-D array. One component per column
%    tagss     array of strings with processing
%done;i.e.,'hghhx   '
%         entry 2^k+i is level k position i+1 (0<i<2^k)
%         this array also indicates the order in which the
%components
%         are packed in arrays xcc and xww
% routines called: hgx, hsgs
%USE  [xww,xcc,lxcc,tagss]=w_ortho_ws(x,L,h,g)

function [xww,xcc,lxcc,tagss]=w_orthoN(x,L,h,g)
[np nx]=size(x);
name1='ccc';
if nx>np
  disp(['you are computing decomposition of ' num2str(nx) '
curves ']),
  disp(['each with ' num2str(np) ' points '])
  disp('Going ahead with process... Press crtl-C to cancel
')
else
end
eval([name1 '001=x;']);  %start with level zero
```

```
tagss=['x       ']; %identify processing done
xcc=[]; %store here;
xww=[];

for k=1:L
% take previous level and arrays and apply h and g filter to
%each one
    na=2^(k-1);     %number of arrays in previous level.
%This is also pointer to first tag of previous level
    for ka=1:na
     id=num2str(ka);
     if ka<10
         id=['0' id];
     else
     end
     idkm1=[num2str(k-1) id];
     name=[name1 idkm1]; %previous level name
     tagkm1=tagss(na+ka-1,1:7);    %previous tag minus one '
     xw=eval(name); %put previous array in xw
     [xwh,nzh]=hgx(h,xw,1);   %apply h filter to it
     [xwg,nzg]=hgx(g,xw,1);   %apply g filter now
% generate names for storage
     idh=id;   %h is taken as binary zero as new MSB
     idg=num2str(2^(k-1)+ka); %g is a binary one as MSB
     if 2^(k-1)+ka<10
         idg=['0' idg];
     else
     end
     idkh=[num2str(k) idh];
     idkg=[num2str(k) idg];   %id for data created now
     nsth=[name1 idkh];  %form name strings
     nstg=[name1 idkg];
     tagkh=['h' tagkm1];
     tagkg=['g' tagkm1]; % k level tags
     eval([nsth '=xwh;']); %store in variable with that name
     eval([nstg '=xwg;']);
     xcc=[xcc;xwh;xwg]; %save the compressed components
     if ka==1
          [lxcc(k), dnx]=size(xwh); %length of this level
     end
     tagss=[tagss;tagkh;tagkg];    %save tags under their
%names
    end        %done with this level
    disp(['compressed components of level ' num2str(k) '
computed']),
end       %done with all levels
```

```
disp(['...proceeding to generate orthogonal
components...']),
%  now  must  compute  the  adjoint  operations  and  form  the
%orthogonal components
namesu=[];     %start array with components
for k=1:L
    for ka=1:2^k
     id=num2str(ka);
     if ka<10
          id=['0' id];
     else
     end
     x_proc=[name1 num2str(k) id];
     xw=eval(x_proc);    %put it in xw
     t_proc=[];
     idec=ka-1;
     for kdc=1:k
          d_p=rem(idec,2);
          idec=fix(idec/2);
          if d_p==1
               t_proc=['g' t_proc];
          else
               t_proc=['h' t_proc];
          end
     end
     for kp=1:k
         hg=t_proc(kp);
         if hg=='h'
              [yw,nz2,lyw]=hsgs(h,xw,1,np);  %do h_*
         else
              [yw,nz2,lyw]=hsgs(g,xw,1,np);  %do g_*
         end
         xw=yw;     %prepare for next processing
     end  %all string processed. Done with this signal
     xww=[xww yw]; %save it in output array
    end   %done with this level
disp(['components of level ' num2str(k) ' computed']),
end  %done with all levels
[ncomp,eight]=size(tagss);
disp(['...all ' , num2str(ncomp-1), ' components
computed...']),
tagss=tagss(2:ncomp,:);
%keyboard
%    hgx.m
% this function computes the operation Hx
% by filtering with a causal FIR and then down sampling.
```

```
% returns the position of the time origin in the output
%array
% if x is a matrix it applies the operation to each column.
% use   [y,nz2]=hgx(h,x,nz1)
%    h    filter used
%    x    signal to be filtered
%    nz1  position of time origin in input
% routines called: none
function [y,nz2]=hgx(h,x,nz1)
l=length(h)-1  ;
[lx,nxs]=size(x);
if lx<l
     x=[x;zeros(l-lx,nxs)]; %append zeros if too short
     lx=l;
end
nz2=nz1;
x=[flipud(x(1:l,:));x];
yy=zeros(lx+l,nxs);
for kx=1:nxs
     hx=conv(h,x(:,kx)); %using conv. No need to pad
     yy(:,kx)=hx(l+1:lx+2*l);
end
d=rem(nz2,2);
% d should be either zero or 1
if d>0.25
    y=yy(1:2:lx+l,:);% if nz2 is odd do downsampling from
%first element
else
    y=yy(2:2:lx+l,:);% if nz2 is even subsample from second
%element
end
% now determine the location of origin in down sampled
signal
nz2=fix((nz2+1.01)/2);


%    hsgs.m
% this function performs upsampling and filtering
% with an ant causal FIR filter. It also determines location
%of the origin
%%%%%
% modified 08/18/95
%    eliminates use of xup, the upsampling routine
%%%%%
% use   [y,nz2,ly]=hsgs(h,x,nz1,lX)
%    lX   length of original signal or maximum acceptable
%length.
```

```
%    h    filter to be used
%    x    signal to be filtered
%    nz1  position of time origin in input signal
%    nz2  position of time origin in output signal
%    ly   length of output signal
% if x is a matrix the operation is applied to each column
% routines called: none
function [y,nz2,ly]=hsgs(h,x,nz1,lX)
[tf,nxs]=size(x);
yy=zeros(2*tf,nxs);
yy(1:2:2*tf,:)=x(1:tf,:); %upsample
%yy=xup(x);    %first upsample the input
lh=length(h);
%yy=[yy;zeros(lh-1,nxs)];     %pad   with   zeros   to   avoid
%aliasing
[tf nxs]=size(yy);
% take the filter and make it ant causal
h_=flipud(h);
y=[];
for k=1:nxs
    yyt= conv(h_,yy(:,k));    %apply filter to each column
    y=[y yyt]; %save it in array y
end
% cut the initial segment
     y=y(lh:tf+lh-1,:);
% locate position of the time origin
 nz2=2*nz1-1;
[tf txs]=size(y);
if tf> lX
     y=y(1:lX,:);
end
[ly txs]=size(y);
```

Program Listing 4

To compute the short time Fourier series to create indicators

```
%sfft.m
%computes ShortTimeFourierTransform for columns in an array
% the user can specify the window. Defaults to Gaussian with
%256 points
% modified 06/23/97 to allow the user to specify number of
%FFT points
%    SFTF[x](w,tau)=Fourier[x*g(t-tau)]
% use [pyy]=w_stft(x,dtau,lw0)
%    x    (np,nx)   signal array
%    dtau integer increment for the window shift (tau)
%    lw0      desired length of FFT (>=length(g))
%    pyy      magnitude spectrum of fft rows=half the length
%of fft points
%           each column is a signature
% may 2002
function [pyy]=sfft(x,dtau,lw0)
lw=lw0;
lwf=lw0;

dtau=fix(dtau);
j=sqrt(-1);

% check if the data size is a multiple of dtau, if not pad
%zeros
rm=rem(size(x,2),dtau);

if(rm~=0)
    xx=[x; zeros((dtau-rm),1)];
end
[np,nx]=size(xx);

% xx=[zeros(lw,nx);x;zeros(lw,nx)];
gw=ones(1,nx); %one window for every signal
fx=[];tt=[];pyy=[];py=[];
tshift=-lw/2; %initial center for window
nff=0;

for k=1:dtau:np-lw+1
```

```
%    tt=[tt;  -lw/2+nff*dtau];  %current  center  location  for
%window
     nff=nff+1;
     xg=xx(k:k+lw-1,:).*gw; %apply the window
     fxg=fft(xg,lwf);
     %correct the phase
     phc=exp(-j*(2*pi*(k-1)/lwf)*tshift*[0:lwf-1]');
     fxg=fxg.*(phc*ones(1,nx));
    py=fxg.*conj(fxg)/lwf;
    %plot(py(1:lwf/2),'b');
    pyy=[pyy py(1:lwf/2)];
     fx=[fx fxg];
end
```

Program Listing 5

To compute the singular value decomposition , creates vector planes and compute distance of indicators from the plane

```
%s_v_d.m
%this program finds the singular value decomposition of the
%signature matrix and also inputs the number of eigen
%vectors to be considered
%a plane formed by these vectors is considered and the
%distance from these planes for pre and post fault
%signatures
%is the basis for clustering the signatures
%maj1       the diagonal elements of pre-fault S matrix
%maj2       diagonal elements of post fault S matrix
%d11        distance of pre sign from pre -fault plane
%d12        distance of pre from post plane
%d21        distance of post from pre
%d22        distance of post from post plane
%spre       pre-fault indicator matrix (each row is an
%indicator)
%spost      post-fault indicator matrix

function [maj1,maj2,d11,d12,d21,d22]=s_v_d(spre,spost)

spr=[];spo=[];maj1=[];maj2=[];
upr=[];upo=[];vpo=[];vpr=[];
%find the svd and choose the elements to be considered
[upr,spr,vpr]=svd(spre);
[upo,spo,vpo]=svd(spost);

maj1=diag(spr);
maj2=diag(spo);
figure
subplot(2,1,1)
stem(maj1)
title('Singular elements of Pre-fault signatures')
xlabel('singular elements')
ylabel('magnitude')
subplot(2,1,2)
stem(maj2)
title('Singular elements of Post-fault signatures')
```

```
xlabel('singular elements')
ylabel('magnitude')
spree=spre';
spoo=spost';

%  finding  the  number  of  principal  componenets  to  be
%considered
%c=input('enter the number of eigenvalues to be considered :
%')
c=num_PrComp(maj1)

%compute the angle between the planes
v1=vpr(:,1:c);
v2=vpo(:,1:c);
s=svd(v2'*v1*v1'*v2);
ang=(180/pi)*acos(sqrt(s(1)))


% this part tries to find the distance of the signatures(pre
%and post) from two planes,
% pre fault plane and post fault plane. The pre fault plane
%formed by (c) eigen vectors which have
% significant eigenvalues based on the svd of the signatures
%to  find  the  co-ordinates  of  the  projection  of  each
%signature on the plane formed by c vectors of V matrix
%alpha1 and alpha2
alpha11=[];alpha21=[];
alpha12=[];alpha22=[];

for x=1:c
    alpha11(x,:)=dot(spree,(vpr(:,x)*ones(1,size(spre,1))));
    alpha12(x,:)=dot(spree,(vpo(:,x)*ones(1,size(spre,1))));
    alpha22(x,:)=dot(spoo,(vpo(:,x)*ones(1,size(spost,1))));
    alpha21(x,:)=dot(spoo,(vpr(:,x)*ones(1,size(spost,1))));
end

% alpha11   %alphas for distnace of pre sig from pre-plane
% alpha12   %alphas for distance of pre sig from post-plane
% alpha21   %alphas for distance of post sig from pre-plane
% alpha22   %alphas for distance of post sig from post-plane
```

%to calculate the distance of the signatures from pre-fault plane and post fault plane

```
%distance= sign-alpha*v for all alpha and c number v's
%where a11=alpha*v for pre from pre
%       a12=pre from post
%       a21=post from pre
```

```
%       a22=post from post

av11=zeros(size(spree));
av21=zeros(size(spoo));
av12=zeros(size(spree));
av22=zeros(size(spoo));

for y=1:c
    av11=av11+(vpr(:,y)*alpha11(y,:));
    av12=av12+(vpo(:,y)*alpha12(y,:));
    av21=av21+(vpr(:,y)*alpha21(y,:));
    av22=av22+(vpo(:,y)*alpha22(y,:));
end

dif1=zeros(size(spree));
dif2=zeros(size(spoo));
dif3=zeros(size(spree));
dif4=zeros(size(spoo));

dif1=spree-av11;
dif2=spree-av12;
dif3=spoo-av21;
dif4=spoo-av22;

d11=zeros(size(spree));
d21=zeros(size(spoo));
d12=zeros(size(spree));
d22=zeros(size(spoo));

d11=sqrt(sum(dif1.*dif1));
d12=sqrt(sum(dif2.*dif2));
d21=sqrt(sum(dif3.*dif3));
d22=sqrt(sum(dif4.*dif4));

figure
subplot(2,2,1),stem(d11)
xlabel('Indicator number');ylabel('Distance from pre-fault
plane')
title('Distance cluster for Pre-fault indicator')
subplot(2,2,2),stem(d12)
xlabel('Indicator number');ylabel('Distance from post-fault
plane')
title('Distance cluster for Pre-fault indicator')
subplot(2,2,3),stem(d21)
xlabel('Indicator number');ylabel('Distance from pre-fault
plane')
title('Distance cluster for Post-fault indicator')
```

```
subplot(2,2,4),stem(d22)
xlabel('Indicator number');ylabel('Distance from post-fault
plane')
title('Distance cluster for Post-fault indicator')
```

Program Listing 6

To implement the clustering technique to classify the indicators

```
%bestcase.m
%this function clusters the pre and post fault signals
%divides the signal based on the distance from average
signatures of pre
%and post fault signals and divide them into correct
%decision, wrong decision and undecidable
%
%[rpre,rpost]=cluster(sigPr,sigPo)
%sigPr      matrix of pre-fault signatures each row is a
signature
%sigPo      matrix of post-fault signatures each row is a
signature
%rpre       best distance from pre- fault avg signature
%rpost      best distance from post-fault avg signature

function [m1,m2,n11,n12,n21,n22]=bestcas(sigPr,sigPo)

%find the average signature
avgPr=[];
avgPo=[];
avgPr=(mean(sigPr))'; % its a column vector
avgPo=(mean(sigPo))'; % its a column vector



%find the percentage of pre and post signatures lying around
%the pre and post avg within distance of r
%m1, m2 the principle diagonal elements for pre and post
%n11, n12, n21, n22 array of distance of each signature from
%pre/post avg
[m1,m2,n11,n12,n21,n22]=s_v_d(sigPr,sigPo);

[m11,n1]=size(n11);
[m22,n2]=size(n21);
p1=zeros(1,3);p2=zeros(1,3);
%arrays storing the number of case in each row col 1:correct
%cases
% col2: uncorrect col3: undecidable  col4:unaccountable
```

```
x=1;
%depending on the value of n's it is classified as correct,
%un correct and undecidable
%which doesn't belong to the above 3 cases is %unaccountable
%For pre-fault signatures
%        nearer to pre-fault avg        correct(1)
%        nearer to post-fault avg       uncorrect(2)
%        equidistant to both avg's      undecided(3)

for k=1:n1
    if(n11(1,k) < n12(1,k))
          p1(x,1)=p1(x,1)+1;
    elseif(n11(1,k) > n12(1,k))
          p1(x,2)=p1(x,2)+1;
    else
          p1(x,3)=p1(x,3)+1;

    end
end

%For post-fault signatures
%        nearer to post-fault avg        correct(1)
%        nearer to pre-fault avg       uncorrect(2)
%        equidistant to both avg's      undecided(3)

for k=1:n2
    if(n22(1,k) < n21(1,k))
          p2(x,1)=p2(x,1)+1;
    elseif(n22(1,k) > n21(1,k))
          p2(x,2)=p2(x,2)+1;
    else
          p2(x,3)=p2(x,3)+1;

    end
end

%calculate percentage correct,uncorrect and undecidable

p11=(p1(1,1)/n1)*100;
p12=(p1(1,2)/n1)*100;
p13=(p1(1,3)/n1)*100;

p21=((p2(1,1)/n2)*100);
p22=((p2(1,2)/n2)*100);
p23=((p2(1,3)/n2)*100);
figure
```

```
hold on

subplot(1,2,1),bar(p1(1,:));
text(1,p1(1,1)+1,[int2str(p11),'%'])
text(2,p1(1,2)+1,[int2str(p12),'%'])
text(3,p1(1,3)+1,[int2str(p13),'%'])

ylim([0 n2+1])
xlabel('1: Correct case          2: Incorrect case')
ylabel('number of indicators')
title('Statistics for Pre-fault indicators distance')
subplot(1,2,2),bar(p2(1,:));
text(1,p2(1,1)+1,[int2str(p21),'%'])
text(2,p2(1,2)+1,[int2str(p22),'%'])
text(3,p2(1,3)+1,[int2str(p23),'%'])

ylim([0 n2+1])
xlabel('3:Equidistant case')
ylabel('number of indicators')
title('Statistics for Pre-fault indicators distance')
hold off
```

VITA

Pallavi Virupaksha Chethan was born on June 27, 1978, in Tumkur, India. After graduation from " Sarvodaya" Pre University College in 1994, she studied at "Siddaganga Institute of Technology" of Tumkur, India, from which she graduated with a Bachelor of Engineering degree in 2000. From January 2002, Pallavi Virupaksha Chethan has been a graduate student and teaching/research assistant with the Department of Electrical and Computer Engineering at Louisiana State University. She is presently a candidate for the degree of Master of Science in Electrical Engineering at Louisiana State University, which will be conferred in August, 2003.