NEURAL NETWORK BASED MACHINE CONDITION MONITORING SYSTEM

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAI'I IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

MAY 2004

By
Kenichi Kaneshige

Thesis Committee:

Vassilis Syrmos, Chairperson
James Yee
Tep Dobry

3857

Acknowledgements

I am deeply grateful to my advisor, Professor Vassilis L. Syrmos, for his enormous knowledge, patience, support, guidance and friendship throughout my M.S. program at University of Hawai'i. I would like to thank him for accepting me as his student and friend. I also want to thank Professor James Yee and Professor Tep Dobry for their so kindful help to my thesis work.

This work is dedicated to my parents for encouraging me always strive to be the best at everything I do.

I would like to thank my office mates, Xudong Wang, Mark Saewong, and Martian Binonwangan for their great encouragement and support.

ABSTRACT

In this paper, machine condition monitoring techniques based on multilayered feedfoward neural network (MLFFNN) where the weights in the network are updated based on node-decoupled extended Kalman filter (NDEKF) training method are proposed. Neural network based techniques have been widely recognized as powerful approaches for condition monitoring system, and the use of NDEKF has better performances in computational complexity and memory requirement among the Kalman filtering algorithm family. The condition monitoring system detects and identifies conditions of components through the neural network based system identification of components. Sensor signals in both time and frequency domains are analyzed to show the effectiveness of the condition monitoring scheme. The performances of diagnostic tools presented in this thesis are evaluated using the cabin temperature control system that is specifically for Boeing 767 as practical application example, and the results show the effectiveness of the developed techniques.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Most of the maintenance actions today are carried out by the run-to-failure technique. In the past the asset condition was monitored by the five senses of the operator. With the run-to-failure technique, one lets the component, sub-system, or system run until they break down or an obvious fault occurs. It is absolutely essential to increase productivity and reduce operational costs by adopting a system that has the capability of predicting accurate and reliable time-to-failure.

System diagnostics is defined as a technology intended for timely, non-destructive detection and identification of incipient failures of hardware caused by system abnormalities. The diagnosing technique is to process the sensory data and perform diagnostic reasoning by comparing the measurements against the machine fault signatures. Typically, diagnosis methodologies are subdivided into two classes; one is model-based signal processing technique, and the other is a pattern recognition method. In case where the process can be represented by a model, model-based diagnostic technique can be applied to account for variations in the measurements caused by changes in operating conditions. The technique then detects faults through a use of parametric identification techniques. When suitable process models are not available or when they are too complicated to be extracted

from data, nonparametric identification techniques are used as a diagnostic system. In either case, the difference between sensor measurements and expected values in normal and fault conditions are used to identify the current condition.

In this thesis, pattern recognition based fault diagnostic system based on a neural network is described. Neural network's learning feature is effective in capturing the time-varying and individual behavior of a complex physical system. Therfore, neural network can improve the fidelity of the machinery used.

Originally inspired by the structure of the human brain, an artificial neural network is a mathematical tool modeled after the networks of nerve cells in the brain. The primary advantage of neural networks over traditional modeling is their ability to learn patterns and extract features. This ability is used as a powerful tool in treating time-varying characteristics, and it offers significant potential in practical applications that include pattern mapping, pattern classification, and prediction.

Figure 1.1 describes the fault diagnosis system in a block diagram form. In parallel with the system that can consist of actuators, process dynamics, and sensors, neural networks with the characteristics of possible conditions are used for residual generation. Based of the result of residual generation, the system can be diagnosed through neural network based classification techniques.

The ability of neural network to approximate an unknown input-output mapping can be exploited through system identification technique as described by the Figure 1.2, where the output of neural network y_i is produced in response to an input vector x_i. The difference between the plant output d_i and the network output y_i provides the error signal

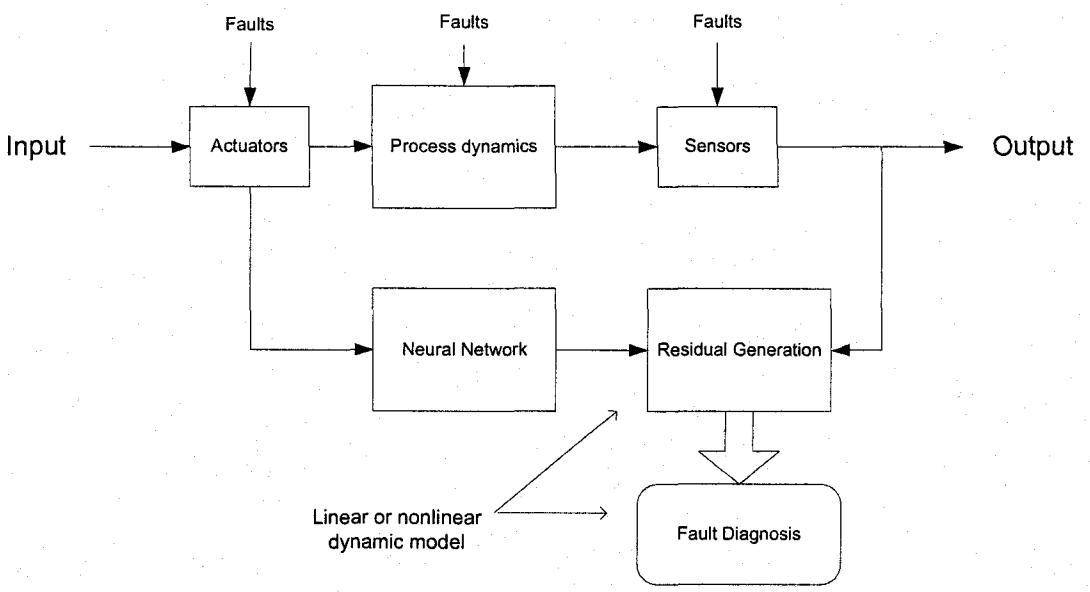


Figure 1.1: Fault diagnosis system in a block diagram form.

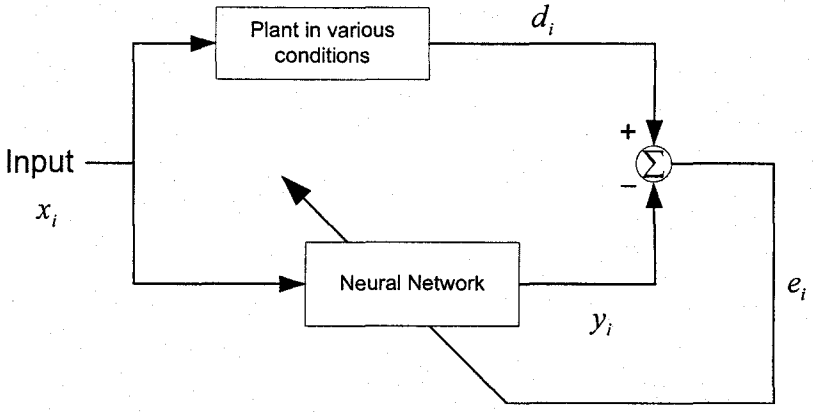


Figure 1.2: Block diagram of system identification.

vector e_i that is used to adjust the parameters of the network to minimize the difference of the plant and the network.

The Kalman filter is well known as a state estimation method for a linear system, while the Extended Kalman filter (EKF) is known as a state estimation method for a nonlinear system. The filter can also be served as a basis for a number of recent neural

network training algorithms. A multilayered feedfoward neural network (MLFFNN) is a nonlinear system having a layered structure, and its learning algorithm is regarded as a parameter estimation for such a nonlinear system. Since this EKF-based learning algorithm approximately gives the minimum variance estimate of the linkweights, the convergence performance is improved compared to the popular standard backpropagation (BP) algorithm.

Although the BP algorithm has shown successful results compared to the methods invented in the mid 1980s, many researchers have been looking for alternative methods that yield better performance, such as training speed, mapping accuracy, and generalization. The most promising training methods to satisfy these are the ones that update weights based upon the second-order derivative information, while the standard BP is based on the first derivative. Some popular second order methods are the Quasi-Newton, Levenburg-Marquardt, and Conjugate gradient techniques. However, these often converge to local minima because of the lack of a stochastic component in the weight update procedures. The EKF, on the other hand, is also a second order neural network basis, but its performance is more practical and effective compared to the methods mentioned earlier [1].

In this thesis, the Node Decoupled Extended Kalman Filter (NDEKF), which reduces the computational complexity and memory usage dramatically, is used as a diagnostic reasoner for nonlinear dynamical systems in both time domain and frequency domain. The methods used to analyze the sensor signals include probabilistic analysis, time domain analysis, and frequency domain analysis [15][16][17][18][19]. Although the frequency analysis approach is the most popular method among those due to the availability of the Fourier Transform technique that is suitable to extract vibrational features of components with periodical movements (e.g. bearing[15]), it would be better to use time domain anal-

ysis if transients occur at a certain phase of rotation (e.g. detecting imminent failure due to crack formation [16]).

The prognostic schemes that uses Kalman filtering based neural network have been introduced by Dash [7] and extended Kalman filtering by [8][9][10]. The node-decoupling scheme can be used in these prognostic techniques as well.

The remaining part of this thesis is organized as follows: A description of neural network with Kalman filtering that is used for fault diagnosis is presented in the chapter 2. Chapter 3 describes a standard cabin temperature control system used on current aircrafts. A set of simulations for introduced fault detection and identification schemes applied to the system model are presented in chapter 4 in order to validate the methods in industrial applications. Finally in chapter 5, the discussions and future work are presented.

Chapter 2

Neural Network with Kalman Filtering Algorithm for Fault Diagnosis

The fault detection and identification (FDI) system utilizes a bank of dynamic input-output observers, where a hybrid of neural network and Kalman filtering scheme can be utilized. In this chapter, the idea of neural network with its training algorithm based on Kalman filtering algorithm and the failure detection and identification scheme are introduced. The section of Kalman filtering includes the idea of extended Kalman filtering algorithm, which is applied for nonlinear models, and node-decoupled extended Kalman filtering algorithm that requires less computational loads than EKF does. Both time domain and frequency domain analyses used for an FDI scheme are then introduced with neural network based classifier.

2.1 Neural network

A parallel processing structure that has a large number of processors and many interconnections are utilized in neural networks. Although processors used in neural net-

works are much simpler than typical central processing units (CPUs), there are hundreds of thousands of processors linking their neighbors. The power of neural network lies in this interconnected network of the tremendous number of processors. An example of typical processing unit of an artificial neural network is depicted in Figure 2.1. The multiple inputs, each arriving from another unit, are connected to the processing unit. Each interconnection has an associated connection strength called weights, given as $w_1, w_2, ..., w_n$. The processing unit performs a weighted sum on the inputs and uses a nonlinear function f to compute its output. The computed outputs are then sent to the target cells along the output connections.

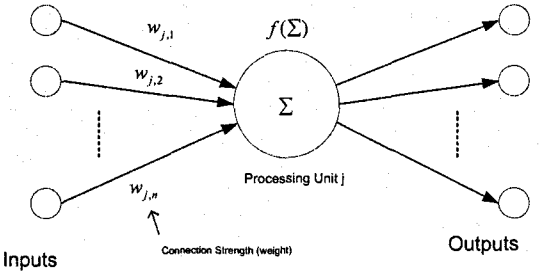


Figure 2.1: An example of typical processing unit of an artificial neural network.

In a layered neural network the neurons are organized in the form of layers. When there are one or more hidden layers, which are the layers between input layer and output layer, the neural network is called multilayered feedfoward neural network (MLFFNN). The purpose of hidden neurons is to intervene between the input and the network output in some useful manner. With the addition of one or more hidden layers, the network obtains an ability to extract higher-order statistics.

An architectural graph for layout of MLFFNN for the case of 100 input neurons, 20 hidden neurons in the first hidden layer, 10 hidden neurons in the second hidden layer, and 100 output neurons (abbreviated as 100-20-10-100 neural network) is illustrated in

Figure 2.2. The bias units provide a constant term in the weighted sum of the units in the next layer and improves the convergence properties of the network.

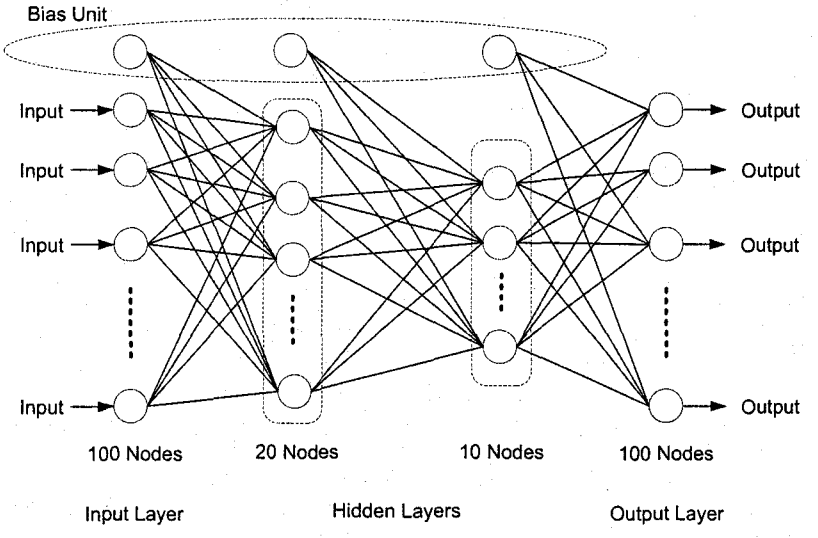


Figure 2.2: Diagram of 100-20-10-100 MLFFNN neural network.

The input signals to the neural network are supplied from the input layer and sent to the neurons (computational nodes) in the second layer (i.e., the first hidden layer). The output signals of the second layer are used as inputs to the third layer, and so on for the rest of the network. The set of output signals of the neurons in the output layer of the network constitutes the overall response of the network to the inputs supplied at the input layer.

2.2 Kalman Filtering Algorithm and Neural Networks

A Kalman filter is an optimal iterative (on-line) data processing algorithm that is used for stochastic estimation from noisy sensor measurements. It can also be looked as a Predictor-Corrector type estimator since a Kalman filter predicts the estimate at some time and corrects it using the prediction error. A Kalman filter is optimal in a sense that it incorporates all the available information (measurements) regardless of their precision

[3],[5]. The Kalman filtering is an effective tool, especially when the signal to noise ratios (SNR) are low, and improves the performance of the FDI system. In this section, 1. The extended Kalman filtering algorithm, which is used for the estimation of a state vector in a nonlinear model of a dynamical system, 2. The decoupled extended Kalman filtering algorithm family that provides an approximation to extended Kalman filter, and 3. The node-decoupled extended Kalman that provides the approximation based upon the nodes are introduced.

2.2.1 Extended Kalman Filtering and Neural Networks

Numerous methods have been proposed for estimation methods of general nonlinear systems, and the best known among those is the Extended Kalman filter (EKF) [6],[13]. It is also known as a parameter estimation method with augmentation of the state, and this can also be regarded as the learning algorithm of a multilayered feedfoward neural network with determination of the weights of the network iteratively.

The EKF is known not only as an optimal recursive data processing algorithm, but also known as the best second-order neural network training method that is practical and effective. During the training of a neural network with EKF algorithm, not only the weights, but also an error covariance matrix that encodes second-order information is maintained. This EKF procedure was developed in late 1980s to enable the application of feedfoward and recurrent neural network in various fields including signal processing, control, and pattern recognition, and is shown to be substantially more effective than standard BP [1]. The disadvantage of the EKF is a computational complexity due to second-order information that correlates every pair of network weights.

A behavior of neural network can be described by the following nonlinear dynamical system:

$$\begin{aligned} w_{k+1} &= f(k, w_k) + \omega_k \quad Process \quad equation \\ y_k &= h(k, w_k) + \nu_k \quad Measurement \quad equation \end{aligned}$$

Here, the $f(k, w_k)$ is a nonlinear equation of the dynamical system with ω_k and ν_k representing the process and measurement noise respectively. The nonlinear function $h(k, w_k)$ in the measurement equation relates the state w_k to the measurement y_k. The idea of extended Kalman filter is to extend the Kalman filtering through a linearization of the two equations above at each time instant with the most recent state estimate. The summary of extended Kalman filtering training algorithm is following:

EKF Algorithm

Process Equation: $w_{k+1} = w_k + \omega_k$

Measurement Equation: $y_k = h_k(w_k, u_k) + \nu_k$

Process Noise: $\omega_k \sim N[0, Q] \quad E[\omega_k, \omega_l^T] = \begin{cases} Q, & \text{for } k = l \\ 0, & \text{for } k \neq l \end{cases}$

Measurement Noise: $\nu_k \sim N[0, R] \quad E[\nu_k, \nu_l^T] = \begin{cases} R, & \text{for } k = l \\ 0, & \text{for } k \neq l \end{cases}$

where

w_k: Weight Parameter Vector

y_k: Desired Response Vector

h_k: Nonlinear Function

u_k: Input Vector

$$A_k = [R_k + H_k^T P_k H_k]^{-1} \quad (2.2.1)$$

$$K_k = P_k H_k A_k \quad (2.2.2)$$

$$\hat{w}_{k+1} = \hat{w}_k + K_k \varepsilon_k \quad (2.2.3)$$

$$P_{k+1} = P_k - K_k H_k^T P_k + Q_k \quad (2.2.4)$$

where

$\hat{w}$: Estimated value of weight w for group i

P: Approximate error covariance matrix of w for ith group's weights

K: Kalman filter gain matrix

ε: Error between desired and actual outputs

R: Measurement noise covariance matrix

Q: Process noise covariance matrix

H: Matrix of derivatives of the network's outputs with respect to all trainable weight parameters

The vector $\hat{w}_{k+1}$ is the estimates of the state (i.e. weight) of the system at time step k. The Kalman filter residual vector is defined as $\varepsilon_k = y_k - \hat{y}_k$, where y_k is the desired vector and $\hat{y}_k$ is the network's output vector for kth presentation of a training pattern. Figure 2.3 shows the signal flow diagram for extended Kalman filtering neural network training.

The EKF training procedure required for one step is divided into 5 steps as follows:

1. An input training pattern u_k is propagated through the network to produce an output vector $\hat{y}_k$. The Kalman filter residual vector ε is also generated in this step.

2. The matrix of derivatives of the network's outputs $H_k$are obtained by backpropagation.

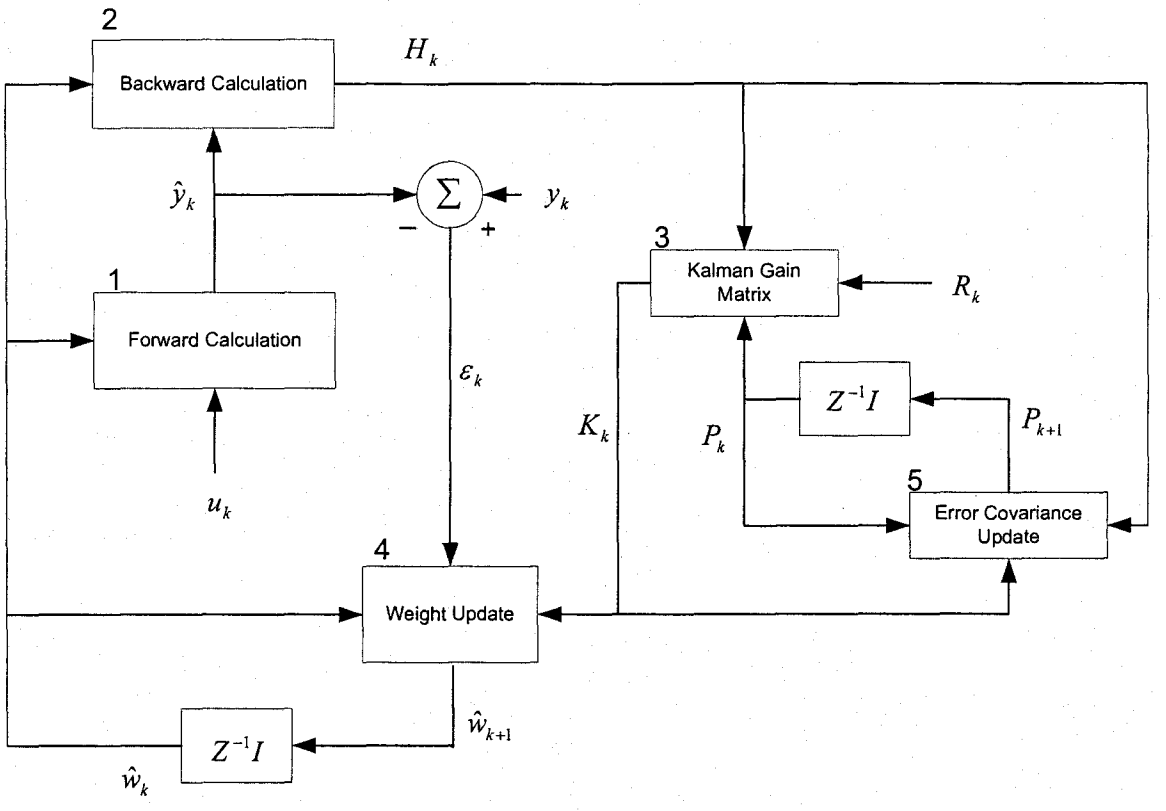


Figure 2.3: Neural network based on Extended Kalman Filter (EKF) as training algorithm.

3. The Kalman filter gain matrix is computed as a function of the derivative matrix H_k, the approximate error covariance matrix P_k, and the measurement covariance noise matrix R_k. The global scaling matrix A_k is computed in this step as well.

4. The weight vector of the network is updated using the computed Kalman filter gain matrix K_k, the Kalman filter residual ε, and the current values of the weight vector w_k.

5. The approximate covariance matrix is updated using the Kalman gain matrix K_k, the current values of the approximate error covariance matrix P_k, and the derivative matrix H_k.

2.3 Decoupled Extended Kalman Filter Family

While the EKF develops and maintains correlations between each pair of network weights, Decoupled Extended Kalman Filter (DEKF) family provides a natural simplification of the standard extended Kalman filtering algorithm by ignoring the interdependence of mutually exclusive groups of weights, thereby allowing reductions of computational complexity and resources. The family of DEKF includes Layer Decoupled EKF, Node Decoupled EKF, and Fully Decoupled EKF. Each one of them reduces the computational burden compared to EKF. The weight training algorithm using DEKF for MLFFNN is well described by [14], and for recurrent networks by [11],[12].

2.4 Node Decoupled Extended Kalman Filter

Among the family of DEKF, Node Decoupled Extended Kalman Filter (NDEKF) where the weights of the network is decoupled by node, exhibits a substantial reduction of the computational complexity and the memory requirement. Other types of DEKF such as the layered DEKF, in which the weights are grouped by layers, and the fully DEKF, where each of the weights are decoupled, also reduce those unwanted complexity and memory requirements. The reduction rate of the computational burden for FDEKF is the least among the DEKF family, but it yields more errors compare to the NDEKF due to the less available error covariance matrix. LDEKF, on the other hand, yields less error, but computational burden is still much higher than the one of NDEKF [1]. With these characteristics of the DEKF family, NDEKF is chosen as a neural network training algorithm that takes the advantage of both the reduction of the computational complexity and the resultant accuracy.

During the weight updates of the neural network, NDEKF recursively updates a smaller part (i.e. matrix that represents the weights grouped by nodes) of the error covariance matrix at a time and continues until all the nodes in the network are updated.

The NDEKF procedure is based on the following state variable representation [1],[2]:

NDEKF Algorithm

$$A_k = [R_k + \sum_{i=1}^{g} (H_k^i)^T P_k^i H_k^i]^{-1} \quad (2.4.1)$$

$$K_k^i = P_k^i H_k^i A_k \quad (2.4.2)$$

$$\hat{w}_{k+1}^i = \hat{w}_k^i + K_k^i \varepsilon_k \quad (2.4.3)$$

$$P_{k+1}^i = P_k^i - K_k^i (H_k^i)^T P_k^i + Q_k^i \quad (2.4.4)$$

where

g: The number of weight groups

$\hat{w}$: Estimated value of weight w for group i

P: Approximate error covariance matrix of w for ith group's weights

K: Kalman filter gain matrix

ε: Error between desired and actual outputs

R: Measurement noise covariance matrix

Q: Process noise covariance matrix

H: Matrix of derivatives of the network's outputs with respect to all trainable weight parameters

Assume the network architecture has N outputs, M weights, and the number of nodes in the network as G. Then, the computational complexity for the EKF procedure is $O(2M^2N)$ for (2.2.1), $O(N^2M + M^2N)$ for (2.2.2), $O(NM)$ for (2.2.3), and

$O(2M^2N)$ for (2.2.4), thereby the overall complexity is $O(NM^2)$. Similarly, the one for NDEKF procedure is $O(2\sum_{i=1}^{g} M_i^2 N)$ for (2.4.1), $O(N^2M + M^2N)$ for (2.4.2), $O(NM)$ for (2.4.3), and $O(2M^2N)$ for (2.4.4), thereby the overall computational complexity is $O(N^2G + \sum_{i=1}^{G} M_i^2)$.

Note the matrix inversion in (2.2.1) and (2.4.1) requires $O(N^3)$, operations, but this is not significant to overall computational complexity since $M \gg N$ [25].

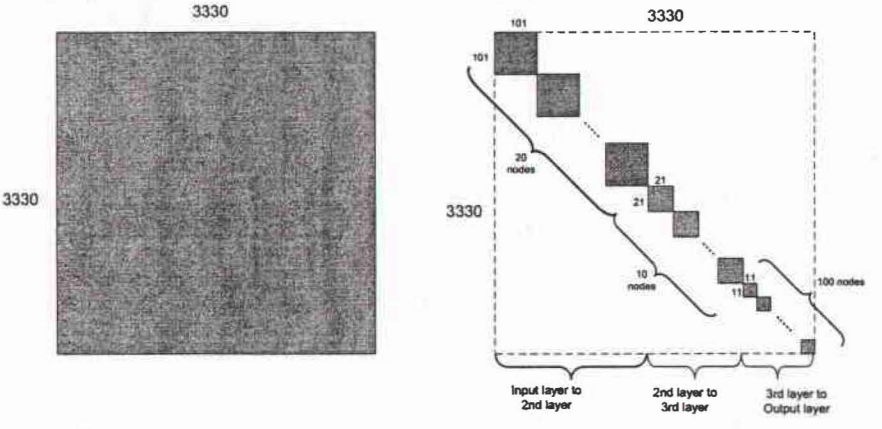


Figure 2.4: Block diagonal representation of the approximate error covariance matrix P for 100-20-10-100 neural network. Note the figure is not scaled.

While EKF generates an error covariance matrix P where each pair of wights in the network are stored, NDEKF performs the decoupling of weights by node. Then the only parts we need to consider are the interdependent weights that are fed into the same node, ignoring the mutually exclusive weights. For the neural network whose architecture is 100-20-10-100 as an example, the memory required is 11088900 for EKF while 220530 for NDEKF, which is only 1.989 %. With this decoupling technique, the error covariance matrix can be reduced as shown in the Figure 2.4.

2.5 Failure Detection and Identification Scheme

A plant may be running in a normal or fault condition. Based on a bank of both normal and fault conditions as a priori knowledge, the neural network with the introduced NDEKF as a training method stores those conditions in the weights of the network. In this section, the neural network based classifications in both time and frequency domain FDI are introduced.

2.5.1 Neural network based classifier

Once a normal condition and failed conditions for a system are prepared, training inputs are fed into the system to obtain a set of inputs and outputs for each condition, which are used to train the neural network based on NDEKF algorithm. The trained neural network remembers the characteristics of the condition by storing them in the weights of the network.

To detect and identify the system condition, the Mean Square Errors (MSE) between the output of the actual system (vibration signal) and the output of the neural network for all the conditions are utilized to classify the current conditions.

Let x be an observation vector when the task is to assign x to one of K classes. A decision rule in terms of discriminant functions is written as follows:

$$\text{Decide } x \in \psi_k \text{ if } f^{(k)}(x) = \min f^{(j)}(x)$$

where $f^{(k)}$ is the discriminant function for class ψ_k. Therefore, a classifier is composed of K discriminant functions. In the case of a neural network classifier, the kth output unit corresponds to the discriminant function for ψ_k.

2.5.2 Time domain analysis

When an input signal is fed into the plant, the same input is fed into neural network. The plant output is then compared with neural network output. Running a plant with a bank of neural network as shown in Figure 2.5 yields the residuals $e_{t,i} = y - \hat{y}_i$, where y is the signals from sensors on the plant and $\hat{y}_i$ is the output from the neural network. The residuals are then used to identify the current condition of the plant.

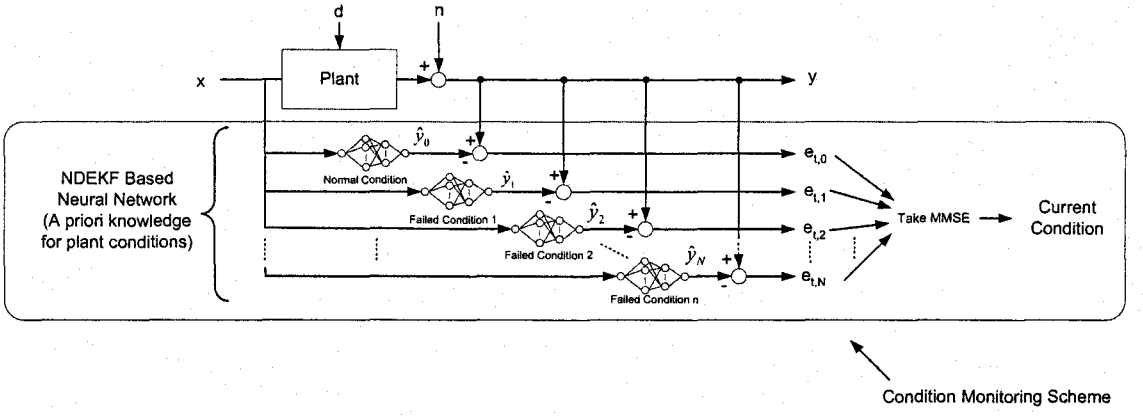


Figure 2.5: Diagram of FDI scheme with time domain analysis

A condition of the plant using time domain analysis(CT) from time 0 to T(sec) with N number of available conditions can be identified by

$$CT_{min} = min(C_{t,1}, C_{t,2}, \cdots, C_{t,N}) \quad where \quad C_{t,n} = \frac{1}{T}\sum_{i=0}^{T} e_{t,n}^2(i)$$

The time T should reasonably be large since we want to know the *trends* of the plant and want to avoid the generation of false alarm (intermittent misclassifications) due to possible dramatic changes in the plant condition.

2.5.3 Frequency domain analysis

Another FDI scheme is the frequency domain analysis approach. This approach is more popular than time domain analysis, especially when the feature of the vibration signals are more important factor. Frequency domain analysis is also popular since most plants as well as the signals from sensors are tend to be in periodical form, which is easy to extract the vibration characteristics from the frequency domain with Fourier transform techniques [15].

When an input signal is fed into the plant and the neural network, the corresponding output from the plant and the neural network are processed through the powerful and popular Fast Fourier Transform (FFT) technique to extract the frequency characteristics of the vibration signals. Then the residuals between the magnitude of the FFT for the plant and the neural networks are obtained and used in a similar manner as in the time domain analysis to identify the current condition of the plant. The diagram of this procedure is represented in Figure 2.6.

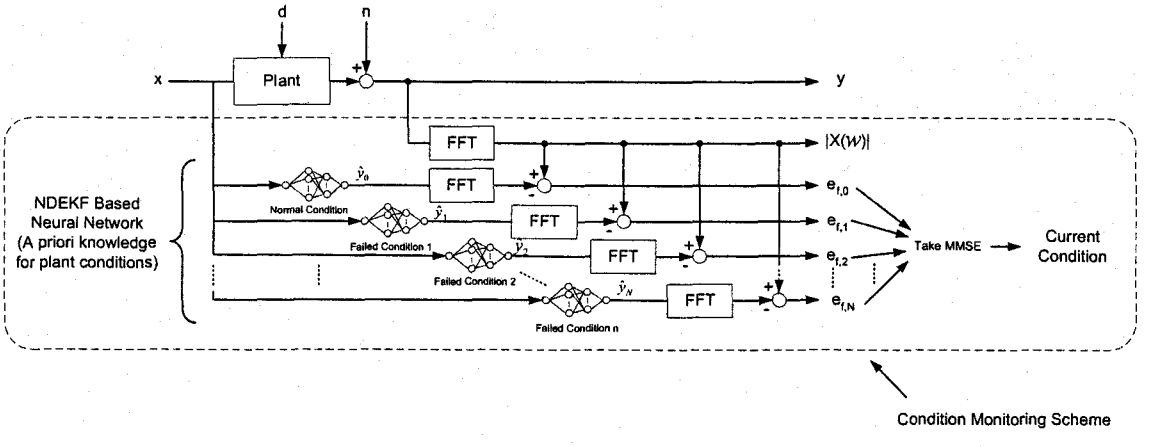


Figure 2.6: Diagram of FDI scheme with frequency domain analysis

A condition of the plant using frequency domain analysis(CF) from time 0 to T(sec) with N number of available conditions can be identified by

$$CF_{min} = min(C_{f,1}, C_{f,2}, \cdots, C_{f,N}) \quad where \quad C_{f,n} = \frac{1}{T}\sum_{i=0}^{T} e_{f,n}^2(i)$$

Chapter 3

Fault Diagnosis to Cabin Temperature Control System

An Environmental Control Systems (ECS) [23] for an aircraft cabin use many types of controls such as temperature, flow, and pressure. In this section, fault diagnosis schemes introduced earlier are applied to the realistic temperature control system models as a practical application example. A standard cabin thermal model is not an active controller like a temperature controller or a flow regulator, but it is required to give a valid representation of the thermodynamical analysis of a cabin.

The system model introduced in this thesis uses U.S. common engineering units, and Table 3.1 lists the major quantities with metric units for reference.

The unit Rankine is a common U.S. common engineering unit for temperature scale that sets zero at absolute zero and uses Fahrenheit degree. The conversion of Rankine can be done through the following equation, and the comparison to other common temperature units are listed in the Table 3.2.

Table 3.1: Units of measure for the introduced model.

Quantity	Metric Unit	U.S.-common engineering unit
Temperature	K	$^{\circ}R = Rankine$
Pressure	bar	psi(pound square inch)
Volume	m^3	ft^3
Volumetric Flow	$\frac{m^3}{min}$	$cfm = \frac{ft^3}{min}$
Heat Transfer Rate	$\frac{W}{K}$	$\frac{Btu*}{hr^{\circ}R}$
Power	W	$\frac{Btu}{hr}$

*Btu (British thermal unit) = 1055.05585J

Table 3.2: Comparisons of temperature units

d	Abs. Temp.	Ice to Water	Water to Vapor
F	-459.67	32	212
C	-273.15	0	100
K	0	273.15	373.15
R	0	491.67	671.67

3.1 Environmental Control System (ECS)

Nearly all commercial aircraft manufactured today use Environmental Control System (ECS) based on bleed air that is the high-pressure hot air ($200kPa$, $392^{\circ}F$) extracted from the engine core through bleed air openings in the side of the engine as described in the Figure 3.1 [24]. The bleed air is then bypassed to air-conditioning 'packs,' where it is cooled and expanded in a rotating air-cycle machine to produce low-temperature air ($80kPa$, $41^{\circ}F$). In the mixing manifold, the air from air-conditioning pack is mixed with extracted cabin air that passes through recirculation fans and filters, and then is distributed to zones in the cabin. Trim air is hot bleed air ($80kPa$, $200^{\circ}F$) that is depressurized by throttling valve.

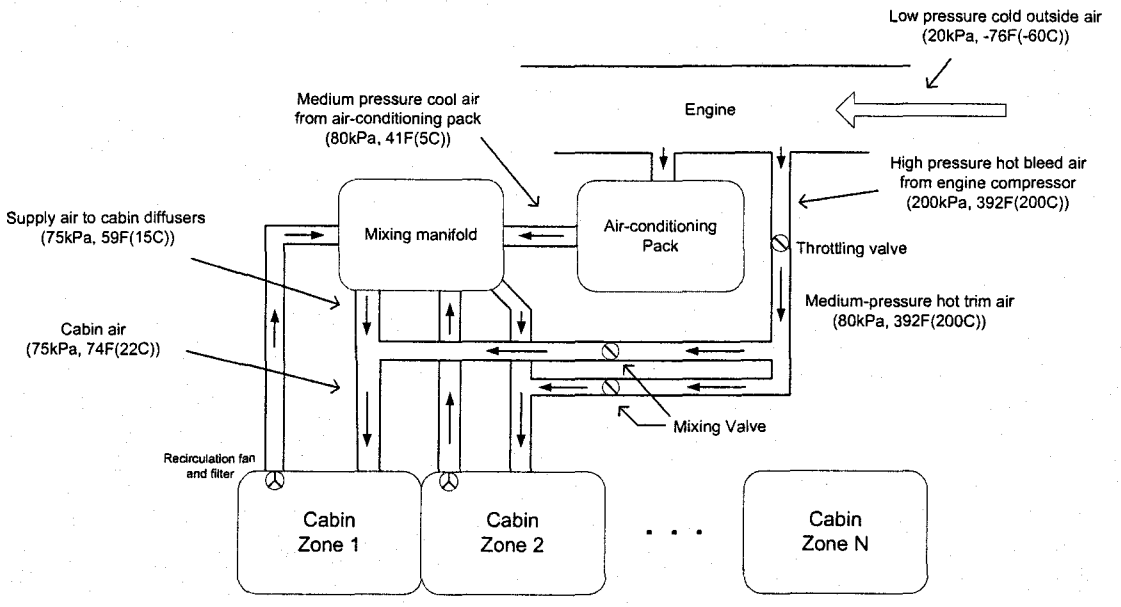


Figure 3.1: The standard cabin air distribution system.

3.2 Cabin Temperature Control System

Cabin temperature control system on current aircraft is normally composed of electronically powered actuators and valves. A cabin air temperature is normally adjusted by varying the temperature of the amount of trim air mixed with supply air to cabin diffusers $(75kPa, 59°F)$.

Some of the factors that would be necessary in terms of the design and complexity of the temperature control system are a. Aircraft size and mission, b. Type of power available, and c. System load requirements. The least sophisticated type of temperature control system acts as a feedback controller where a cabin air temperature sensor initiates the repositioning of valve angle to vary the amount of hot air that goes into cabin. However, this simple control system tends to cause an excessive cabin temperature cycling called 'hunting' due to the relatively large thermal lag of the cabin temperature.

More sophisticated control system can be consists of a supply air temperature sensor, a cabin air temperature sensor, and an anticipator to provide some anticipation and reduce the magnitude of the cabin temperature excursions.

For large cargo aircraft it is necessary to have multizone temperature control as there is the possibility of widely varying loads at different sections.

3.2.1 Standard Cabin Temperature Control System

The analysis of the standard cabin temperature control system model requires the cabin air temperature (T_c), cabin air temperature selected by a pilot (T_{sel}), and the temperature of the duct air that enters the cabin (T_d). The output of the model is the angle of the mixing valve that varies the duct air T_d [23].

The overall standard cabin control system in block diagram form is illustrated in Figure 3.2. The control features of the standard model include an integral controller and a temperature anticipator.

An integral controller consists of an integrator with lead-lag compensation. An integrator is used to achieve a zero steady-state error characteristic for the cabin temperature error signal ϵ_t, while the lead-lag compensation is used to improve the dynamic characteristic of the model. The output of the controller is the a reference mixing valve angle $\theta_{ref}(deg)$ that represents a degree of angle to move from current valve position.

A temperature anticipator is to prevent or reduce overshooting the desired temperature settings. Just before the desired temperature is reached, the valve is adjusted to have a larger angle or a smaller angle. However, the heat remaining in the system brings the resultant temperature off from the desired temperature. A heat anticipator increases or decreases the heat before the temperature overshoots, and this is particularly effective for systems that involve a large thermal lags or varying thermal loads like a cabin.

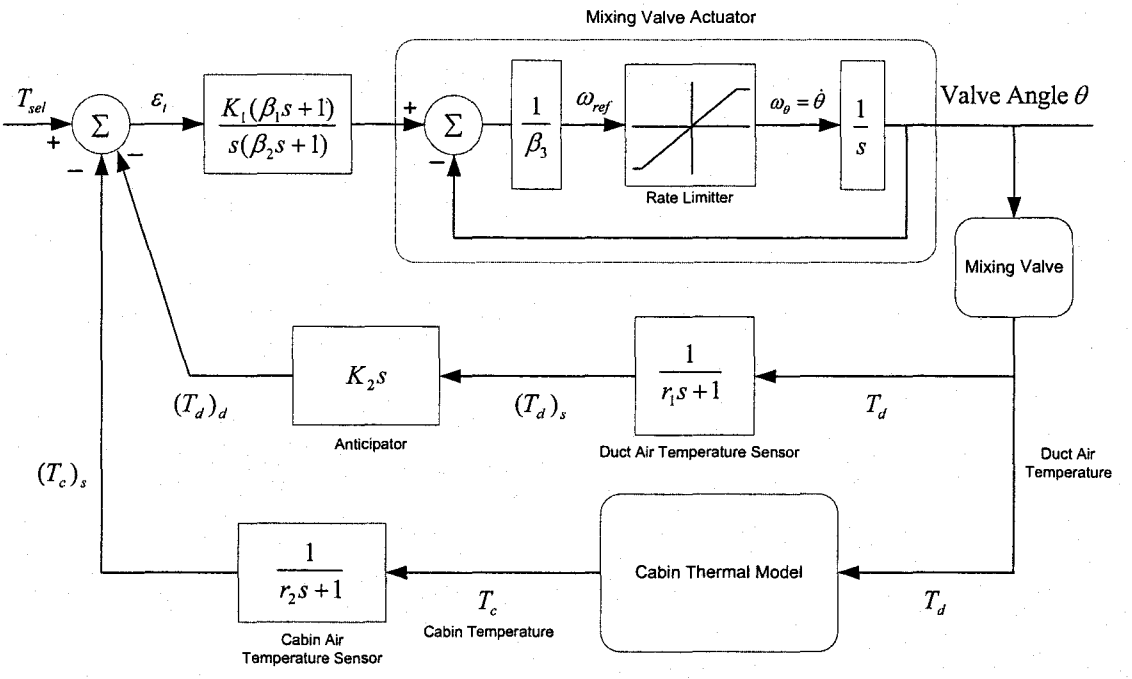


Figure 3.2: The overall standard cabin temperature control system in block diagram form.

3.2.2 Controller Analysis

The relationship of inputs and outputs for the controller represented in Laplace notation is

$$\theta_{ref} = \frac{K_1(\beta_1 s+1)}{s\beta_2 s+1)}\epsilon_t$$

where

ϵ_t : The cabin temperature error signal

θ_{ref} : Reference mixing valve angle

K_1 $(\frac{Btu}{hr°R})$: Controller constant

β_1 $(\frac{Btu}{hr°R})$: Lead compensation time constant

β_2 $(\frac{Btu}{hr°R})$: Lag compensation time constant

The time domain representation of the controller is represented by

$$\frac{d\theta_{ref}}{dt} = (\frac{K_1}{\beta_2})\beta_1\epsilon_t + \frac{K_1}{\beta_2}\int \epsilon_t dt - \frac{\theta_{ref}}{\beta_2}$$

Actuator Analysis

The mixing valve actuator is modeled as depicted in Figure 4.2, where

ω_{max} $(\frac{Btu}{hr})$: Maximum mixing valve angular velocity

ω $(\frac{Btu}{hr})$: Actual mixing valve angular velocity

β_3 $(\frac{Btu}{hr})$: Mixing valve time constant

The rate limiter represents the limitation of the mixing valve angular velocity and can be represented as

$$\omega_\theta = \begin{cases} \omega_{max} + \varepsilon(\omega_{ref} - \omega_{max}) & \text{if} \quad \omega_{ref} > \omega_{max} \\ \omega_{ref} & \text{if} \quad -\omega_{max} \leq \omega_{ref} \leq \omega_{max} \\ -\omega_{max} + \varepsilon(\omega_{ref} - \omega_{max}) & \text{if} \quad \omega_{ref} < -\omega_{max} \end{cases}$$

where $\omega_{ref} = \frac{\theta_{ref} - \theta}{\beta_3}$ and ε is an arbitrarily positive small number.

The valve angle θ is represented in Laplace domain by

$$\theta = \frac{\omega_\theta}{s}$$

and in time domain by

$$\frac{d\theta}{dt} = \omega_\theta.$$

3.2.3 Temperature Sensor

A general expression that describes a temperature sensor is

$$T_{sensed} = \frac{T_{actual}}{\tau s + 1}$$

where τ is sensor time constant. Accordingly, the sensed cabin temperatures T_cs is represented by

$$(T_c)_s = \frac{T_c}{\tau_1 s+1}$$

$$(T_d)_s = \frac{T_d}{\tau_2 s+1}$$

An anticipator is used for the duct air temperature, and the output $(T_d)_d$ is proportional to the time derivative of the sensed duct air temperature. The anticipator, thus, is represented by

$$(T_d)_d = K_2 s(T_d)_s$$

Manipulations of the equations given earlier yields

$$(T_d)_d = \frac{K_2(T_d-(T_d)_s)}{\tau_2}$$

With the variables $(T_c)_s$ and $(T_d)_d$ compared to the cabin air temperature selected by a pilot $(T_{sel}(^\circ F))$, the temperature error signal ϵ_t is given by

$$\epsilon_t = T_{sel} - (T_c)_s - (T_d)_d$$

When the system is at steady state, the error is zero.

3.3 Cabin Thermal Model

A cabin thermal model is a component that describes the thermodynamic representation of a typical aircraft cabin and is a necessary link between the two models, the model of the airflow into a cabin and the model of the cabin temperature and pressure control systems which require the cabin air temperature as an input value.

Although the cabin thermal model derived in this section is appropriate for relatively small cabins such as in battle plane and bombardment aircraft, it would be more appropriate if the model is used as a multizone analysis for large cabins such as in commercial airplanes and cargo aircraft.

For the thermodynamical analysis of the cabin, the heat input/output and heat storage rate that describes how fast a heat can be absorbed by an air are considered. In an airplane the air conditioning system supplies the air that could be a warm or cold air to the cabin as a heat source. We also have heat sources from cabin equipments of the cabin, such as the meal, electronics, lumps, and so on. We can't forget that the passengers and crews also generates heat. On the other hand, the heat is absorbed by the wall that is cooled down by the ambient air whose temperature is $-65°F$ at typical cruise altitude ($36000ft$ ($11000m$)), and also the heat is going back to the air conditioning system. Figure 3.3 illustrates the cabin thermal model in block diagram form.

The cabin thermal model is the model that describes the typical aircraft cabin thermodynamically and is based on an energy equation where heat flow inputs and outputs are equated as following:

$$\text{Energy Equation: } H_1 + Q_m + Q_c = Q_{air} + H_2 + Q_{wi}$$

The definition of the components above is following:

Heat Flow Inputs:

1. H_1 $(\frac{Btu}{hr})$: The enthalpy of the air entering the cabin
2. Q_m $(\frac{Btu}{hr})$: The heat flow from the equipment mass
3. Q_c $(\frac{Btu}{hr})$: The heat generated from passengers

Heat Flow Outputs:

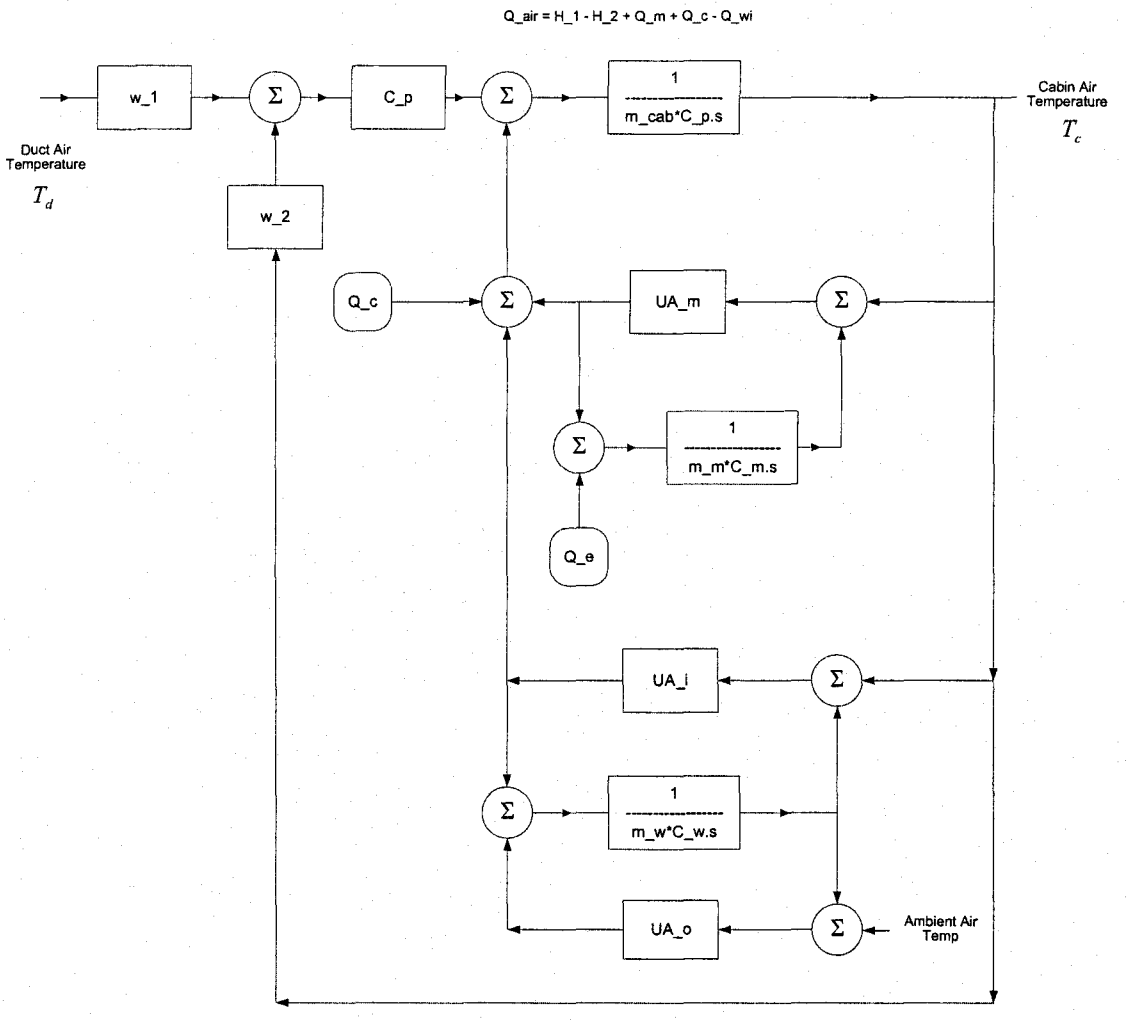


Figure 3.3: Cabin Thermal Model

4. Q_{air} $(\frac{Btu}{hr})$: The heat storage rate of the cabin air
5. H_2 $(\frac{Btu}{hr})$: The enthalpy of the air leaving the cabin
6. Q_{wi} $(\frac{Btu}{hr})$: The heat flow into the cabin walls

The thermodynamic representation of each of the terms above are defined as following:

1. The enthalpy of the air entering the cabin:

$$H_1 = w_1 C_p T_1$$

H_1 ($\frac{Btu}{hr}$) : The enthalpy of the air entering the cabin

w_1: The flow rate of the inlet air

C_p: The mean specific heat of humid air calculated

T_1: The temperature of the inlet air

The specific heat means the amount of heat required to change its temperature by one Kelvin. The specific heat of humid air is calculated by first calculating the specific heat of dry air. The specific heat of dry air is calculated by the following polynomial equation:

$$(C_p)_{air} = 0.24876 - 0.4204563\text{x}10^{-4}T + 0.5767857\text{x}10^{-7}T^2 - 0.1493056\text{x}10^{-10}T^3$$

where T is the temperature of the air

The specific heat of humid air is then calculated by the following equation:

$$C_p = \frac{(C_p)_{air} + 0.46SH}{1+SH}$$

where SH is the specific humidity of the air

2. The heat flow from the equipment mass here includes the heat flow generated from the cabin equipment mass such as electronics, oven, and lights. This heat flow can be approximated by the following equations where the equality on the left describes the static equation, and the on the right describes the dynamic equation. The dynamic equation propagates the time (note that the equipment temperature is a function of time), and the static equation solves the value at that time. UA_m is the overall conductance that works as a measure of how fast the heat of the equipment mass is absorbed by the cabin air.

$$Q_m = UA_m(T_m - T_{cab}) = Q_e - m_m C_m \frac{dT_m}{dt}$$

The above equations can equivalently be represented in state-space form as:

$$\dot{T}_m = -\frac{UA_m}{m_m C_m} T_m + \begin{bmatrix} \frac{1}{m_m C_m} & \frac{UA_m}{m_m C_m} \end{bmatrix} \begin{bmatrix} Q_e \\ T_{cab} \end{bmatrix}$$

$$Q_m = UA_m T_m + \begin{bmatrix} 0 & -UA_m \end{bmatrix} \begin{bmatrix} Q_e \\ T_{cab} \end{bmatrix}$$

where

m_m $(\frac{Btu}{hr})$: Cabin equipment mass

C_m $(\frac{Btu}{hr})$: Specific heat of mass

T_m $(\frac{Btu}{hr})$: Equipment temperature

UA_m $(\frac{Btu}{hr^\circ R})$: Overall heat transfer coefficient between cabin air and internal equipment mass

Q_e $(\frac{Btu}{hr})$: The heat generated by lumped equipment

3. The heat generated in the cabin other than equipment (i.e., from passengers and crews):

Q_c $(\frac{Btu}{hr})$: User input

A person just sitting produces $400(\frac{Btu}{hr})$ and physically active people produces $1450(\frac{Btu}{hr})$. Therefore, Q_c can be approximated as

$Q_c(\frac{Btu}{hr})$ =(num. of passengers)x$(400(\frac{Btu}{hr}))$+ (num. of crews)x$(1450(\frac{Btu}{hr}))$

4. The heat storage rate of the cabin air is the rate that describes how fast heat is stored into the air and is based the ideal gas law that takes the cabin air pressure, air temperature, and the cabin volume into account. The heat storage rate is defined as:

$$Q_{air} = m_{cab} C_p \frac{dT_{cab}}{dt}$$

$$\text{where } m_{cab} = \frac{P_{cab} V_{cab}}{R T_{cab}}$$

P_{cab} $(\frac{Btu}{hr})$: Cabin air pressure

V_{cab} $(\frac{Btu}{hr})$: Cabin volume

T_{cab} $(\frac{Btu}{hr})$: Cabin air temperature

R $(\frac{Btu}{hr})$: Gas constant

5. The enthalpy of the air leaving the cabin is calculated using a similar equation as for the enthalpy of the air entering the cabin.

$$H_2 = w_2 C_p T_{cab}$$

w_2: The flow rate of the exit air

C_p: The mean specific heat of humid air calculated.

T_2: Cabin air temperature

6. The heat flow stored in the cabin walls due to the ambient air can be calculated as following equation:

$$Q_{wi} = U A_i (T_{cab} - T_w)$$

with an energy balance equation for the walls where the change of the uniform wall temperature T_w with its mass and the specific heat is equated with the enthalpy difference between the cabin air and the ambient air as following:

$$m_w C_w \frac{dt_w}{d_t} = U A_i (T_{cab} - T_w + U A_o (T_{rec} - T_w))$$

On the right side of the equation, the difference of the temperature between cabin air and walls are taken with its conductance into account, and the one between the wall and the ambient air is computed similarly. The sum of these differences is then used to find the overall heat loss Q_{wi}

The equations can equivalently be represented in state-space form as:

$$\dot{T_w} = -\frac{(UA_i+UA_o)}{m_w C_w} T_w + \begin{bmatrix} UA_i & UA_o \end{bmatrix} \begin{bmatrix} T_{cab} \\ T_{rec} \end{bmatrix}$$

$$Q_{wi} = -UA_i T_w + \begin{bmatrix} UA_i & 0 \end{bmatrix} \begin{bmatrix} T_{cab} \\ T_{rec} \end{bmatrix}$$

where

m_w $(\frac{Btu}{hr})$: Cabin wall mass

C_w $(\frac{Btu}{hr})$: Specific heat of walls

T_w $(\frac{Btu}{hr})$: Uniform wall temperature

T_{rec} $(\frac{Btu}{hr})$: Recovery air temperature

UA_i $(\frac{Btu}{hr°R})$: Overall heat transfer coefficient (conductance) between cabin air and cabin wall mass

UA_o $(\frac{Btu}{hr°R})$: Overall heat transfer coefficient (conductance) between cabin wall mass and ambient air

Additionally, the following inputs to this cabin thermal model are required:

V_{cab} (ft^3) : Cabin volume

Q_e ($\frac{Btu}{hr}$) : Heat generated by equipment within cabin

Q_c ($\frac{Btu}{hr}$) : Heat generated in the cabin other than Q_e

UA_m ($\frac{Btu}{hr^\circ R}$) : Overall heat transfer coefficient between cabin air and internal equipment mass

UA_i ($\frac{Btu}{hr^\circ R}$) : Overall heat transfer coefficient between cabin air and cabin wall mass

UA_o ($\frac{Btu}{hr^\circ R}$) : Overall heat transfer coefficient between cabin wall mass and ambient air

3.3.1 Mixing Valve Analysis

A control used to change the amount of hot air in the cabin temperature control system is a butterfly valve as described in Figure 3.4. Note that the temperature of the hot inlet air is 392 degree, thus the amount of the air passing the valve is only a small amount.

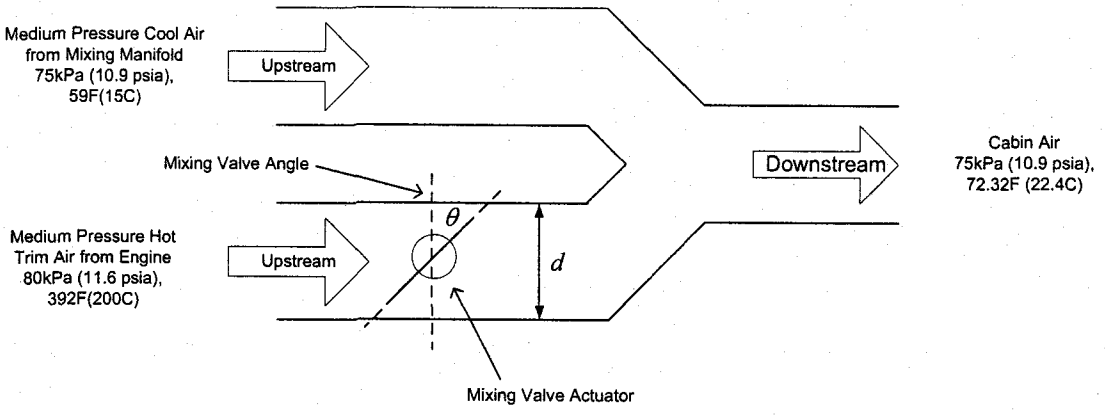


Figure 3.4: Mixing Valve where hot air is varied by butterfly valve.

The flow rate of the air that bypasses a butterfly valve is given by the equation

$$w = \frac{0.532(C_d)_v A_v (N_{cs})_v P_1}{\sqrt{T}}$$

where

$w(\frac{lb}{sec})$: The flow rate

$(C_d)_v$: The discharge coefficient

A_v : The geometric area of the valve

$P_1\ (psia)$: Upstream pressure

$P_2\ (psia)$: Downstream pressure

$T\ (^\circ R)$: Air temperature

$(N_{cs})_v\ (\frac{Btu}{hr})$: Chester Smith compressible flow factor

Assume the valve cross section is circular, then the geometric area of the cross section is given by

$$A_v = \frac{\pi d^2}{4}(1 - cos\theta)$$

where

$d(in^2)$: Valve and duct diameter

$\theta(deg)$: Valve angle

Chester Smith Function

The Chester Smith Function represent N_{cs} that is the ratio of $\frac{ActualFlow}{maximumflow}$ and is represented by

$$\omega_\theta = \begin{cases} (N_{cs})_v = \left[\frac{(\frac{P_2}{P_1})^{\frac{2}{\gamma}} - (\frac{P_2}{P_1})^{\frac{\gamma+1}{\gamma}}}{(\frac{\gamma-1}{2})(\frac{2}{\gamma+1})^{\frac{\gamma+1}{\gamma-1}}} \right]^{\frac{1}{2}} & \text{if } \frac{P_1}{P_2} < P_{crit} \\ (N_{cs})_v = 1 & \text{if } \frac{P_1}{P_2} > P_{crit} \end{cases}$$

where

$P_1\ (\frac{Btu}{hr})$: Cabin air temperature sensor time constant

P_2 $(\frac{Btu}{hr})$: Duct air temperature sensor time constant

γ : Ratio of specific heat $\frac{C_p}{C_v}$

$P_{crit} = (\frac{P_1}{P_2})_{crit} = (\frac{2}{\gamma+1})^{\frac{\gamma}{1-\gamma}}$: Critical pressure ratio where the flow is 'choked' or sonic flow for given values

The value of Chester Smith Function N_{cs} for various downstream pressure P_2 is shown in the Figure 3.5. When the downstream pressure P_2 decreases, an increase of flow cause the value of N_{cs} to be 1 that corresponds to sonic flow. On the other hand, if the downstream pressure P_2 increases, the value of N_{cs} is zero and no flow is passing valve from upstream side to downstream side. In fact, the flow is reversed.

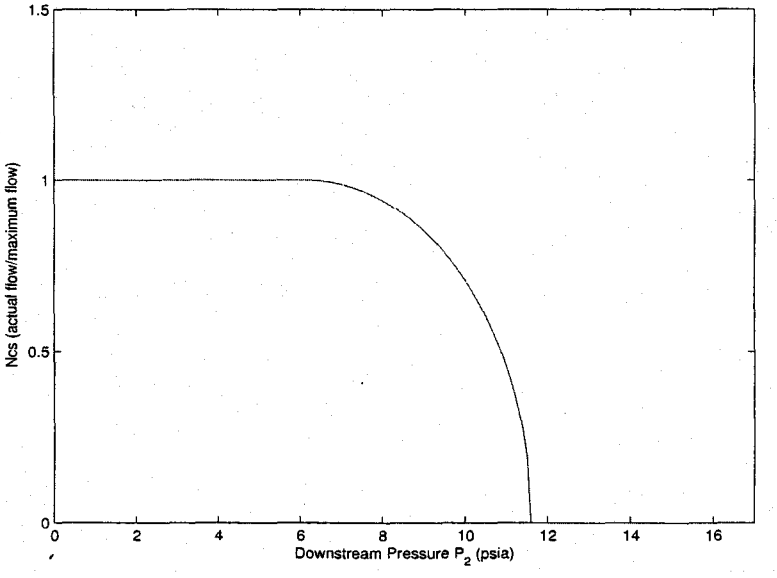


Figure 3.5: Ncs VS Downstream Pressure

Figure 3.6 shows the flow rate of the inlet air w_1 for various downstream pressure P_2 and mixing valve angle θ. While the ratio of upstream and downstream $\frac{P_1}{P_2}$ is between 1 and 1.89, the flow is in normal condition, and this is where typical cabin pressure 10.875 *psia* is in.

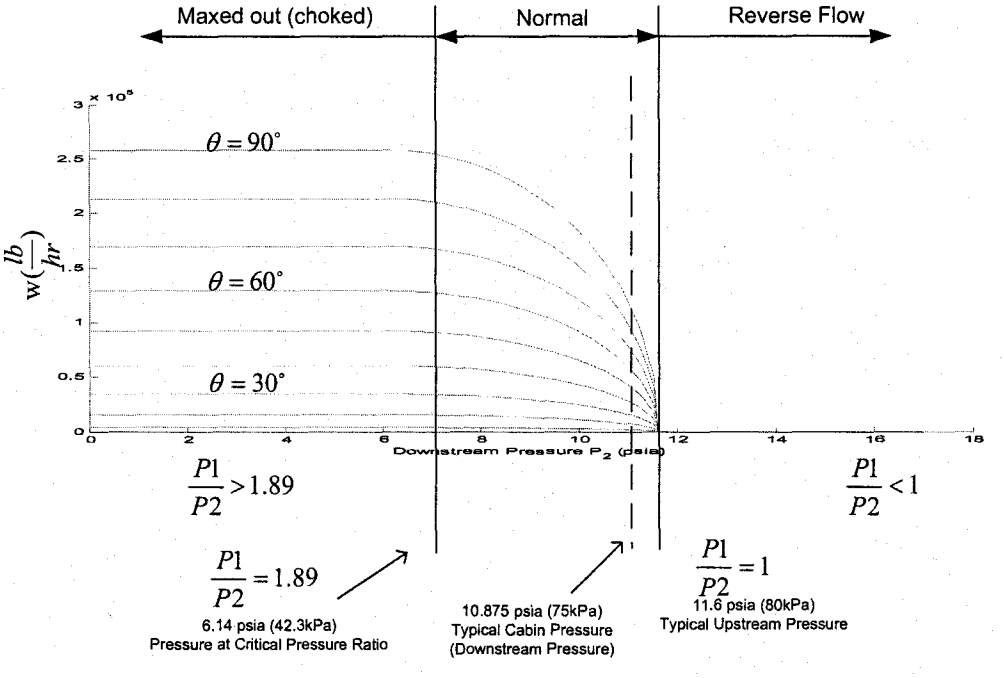


Figure 3.6: Downstream Pressure VS Flow Rate of the Inlet air w1.

Chapter 4

Simulation

In this chapter, cabin temperature control system is used to verify the feasibility of applying neural network with NDEKF algorithm for fault diagnosis. The training of the neural network and the diagnosis of the cabin temperature control system is performed by MATLAB, while the sensor data for the model are obtained through the implementation by Simulink. The fault diagnosis is performed by using banks of neural networks.

4.1 Cabin Temperature Control System for Boeing 767

The parameter values used for the cabin temperature control system are specifically for Boeing 767, and they are listed in the Table 4.1.

The inputs to the cabin temperature control system is the cabin air temperature T_{sel} selected by pilot are assumed to be

$$T_{sel} = 10sin(2\pi\frac{1}{60}t) + 75[^{\circ}F]$$

The output of the system is the angle of the mixing valve. The implementation of the standard cabin temperature control system by Simulink is shown in Figure 4.1. As

Table 4.1: Cabin temperature control system model parameters for Boeing 767

Symbol	Description	Value
V_{cab}	Boeing 767 cabin volume	10594.4 $ft^3(319m^3)$
T_{cab}	Typical cabin temperature	75 $F(23.89°C)$
SH	Typical cabin humidity	15 %
P_{cab}	Typical cabin pressure	12.8 psi
w_1	Flow rate of the inlet air	2388 cfm
w_2	Flow rate of the exhaust air	2388 cfm^{**}
m_m	Cabin equipment mass	1506.1 ft^3
m_w	Cabin wall mass	8078.7 ft^3
Q_c	Heat generated from passengers and crew	2000 $\frac{Btu}{lb°R}$
Q_e	Heat generated by lumped equipment	5000 $\frac{Btu}{lb°R}$
UA_m	UA^* between cabin air and internal equipment mass	1.5 $\frac{Btu}{lb°R}$
UA_i	UA^* between cabin	1600 $\frac{Btu}{lb°R}$
UA_o	UA^* between cabin	1300 $\frac{Btu}{lb°R}$
C_m	Specific heat of equipment mass	0.4 $\frac{Btu}{lb°R}$
C_w	Specific heat of walls	3 $\frac{Btu}{lb°R}$
T_m	Cabin equipment temperature	75 $°F$
T_w	Uniform wall temperature	75 $°F$
T_{rec}	Recovery air temperature	20 $°F$

$*$ UA: Overall heat transfer coefficient

$**$ 1128 cfm of overboard exhaust and 1260 cfm of leakage [24]

depicted in the figure, the cabin thermal model described by Figure 3.3 is embedded into the cabin temperature control system described by Figure 3.2.

4.2 FDI performance evaluation of NDEKF based neural network

Normal condition and three fault conditions on the mixing valve actuator, the anticipator, the duct air temperature sensor, and the cabin temperature sensor are considered and simulated on the monitored systems. The construction of neural network for those

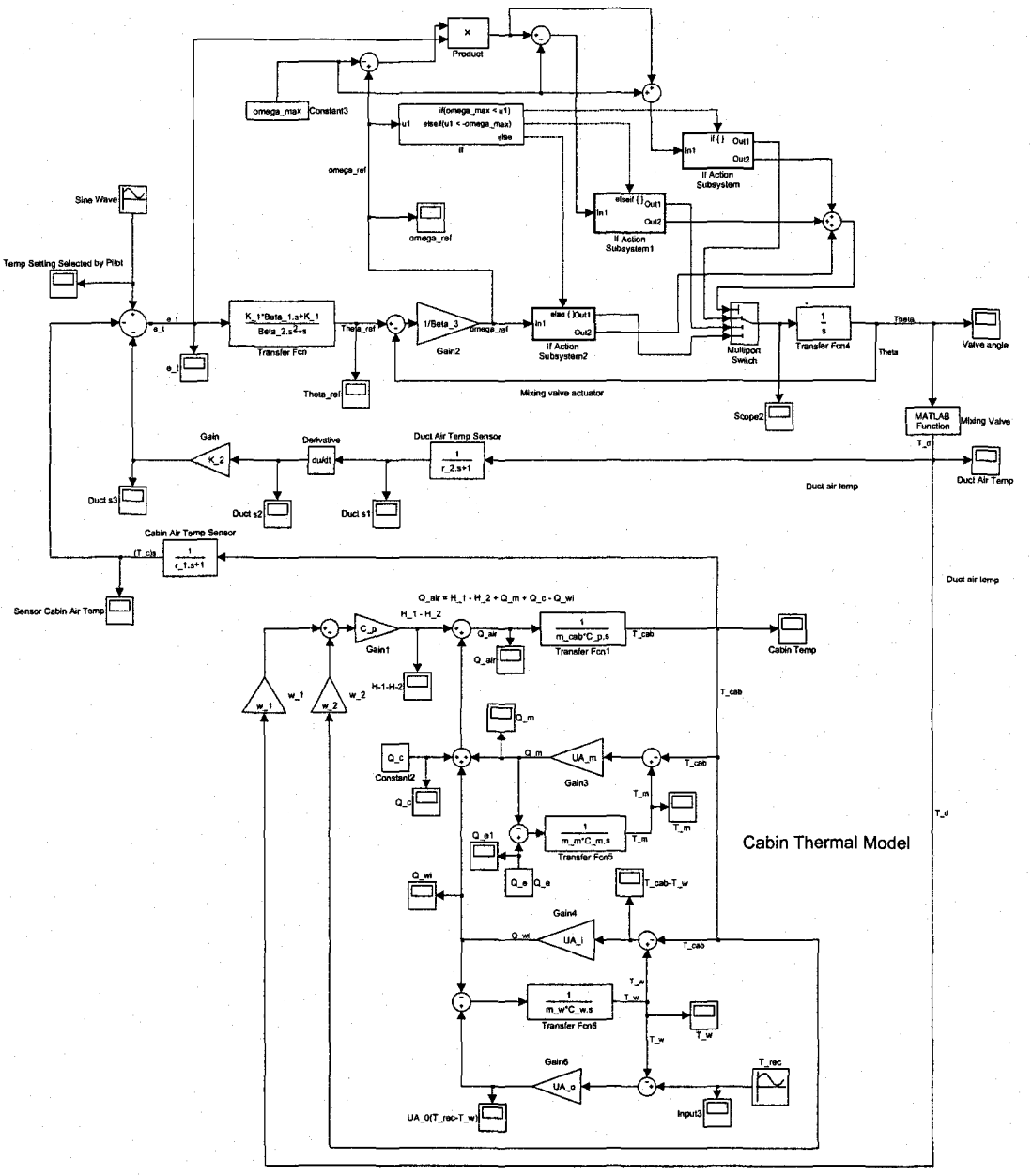


Figure 4.1: The implementation of Standard Cabin Temperature Control System by Simulink.

Table 4.2: The parameter values of components in normal and fault conditions (1: Cabin Air Temperature Sensor, 2: Duct Air Temperature Sensor, 3: Anticipator, 4: Mixing Valve Angle).

	r_1 [1]	r_2 [2]	K_2 [3]	β_3 [4]
NM	0.1	0.1	1.6	0.9
$F1$	2.6	3.2	0.3	3.7
$F2$	5.4	6.7	0.8	2.2
$F3$	6.7	8.5	2.4	4.6

conditions are based on MLFFNN with NDEKF, where the network architecture is 100-20-10-100 as introduced in Chapter 2. The function used in the neural network nodes in the two hidden layers is a Bipolar activation function, and the one used at the output node is a linear activation function. The activation function and its derivative, which is used for computation of derivative matrix during NDEKF procedure, are the following:

$$f(x) = \frac{e^x - 1}{e^x + 1}, \quad \dot{f}(x) = \frac{2e^x}{(e^x + 1)^2} = \frac{1}{2}(1 - f^2(x))$$

The cockpit's setting for the cabin air temperature (i.e., T_{sel}) and their corresponding highly nonlinear outputs from the cabin temperature control system model in a normal condition and three fault conditions are used as training sets for neural network. The criterion of an MSE value for the training (i.e., identification) of the network is set to be 0.1.

An example of the failure scenario for each of the four components used for FDI performance evaluation is depicted in Figure 4.2. Each condition lasts for one time period, and the scenario is fed by two different input sequences; the first sequence from time 0 to 4, and the other sequence from time 4 to 8, where inputs are as following:

$$T_{sel} = \begin{cases} 10sin(2\pi(\frac{1}{60})t) + 75, & \text{for time 0-4} \\ 9.5sin(2\pi(\frac{1}{60})t) + 0.5sin(2\pi(\frac{1}{6})t) + 75, & \text{for time 4-8} \end{cases}$$

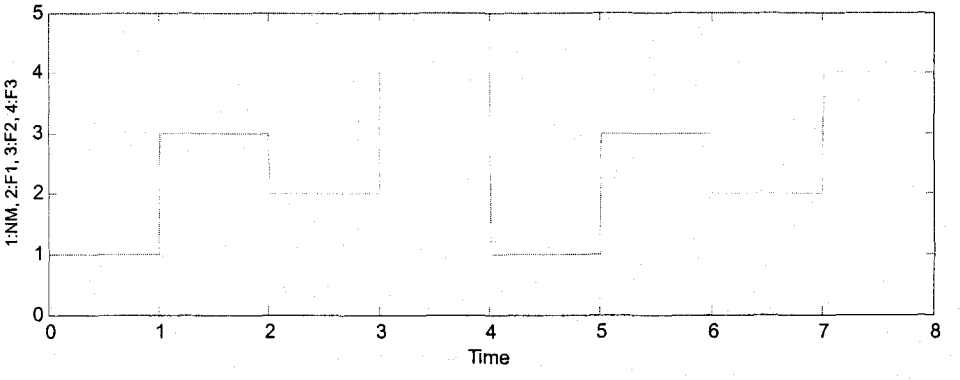


Figure 4.2: Failure sequence for cabin temperature control system

The inputs are made to be different so that the simulation results show the robustness of the neural network against different inputs.

Figure 4.3-4.6 show the responses and the neural network outputs of the mixing valve actuator, the anticipator, the duct air temperature sensor, and the cabin temperature sensor respectively in the four conditions for the cockpit's cabin air temperature settings T_{sel} above. The figures are in the order of normal condition (NM), failure condition 1 (F1), failure condition 2 (F2), and failure condition 3 (F3) from the top, and the parameter values of components in normal and fault conditions are depicted in the Table 4.2.

As can be seen from the figures, the neural network outputs for all the conditions are generated at each time interval and compared with outputs from components. The condition where the difference between component outputs and the neural network outputs are the least indicates the current condition of a component.

Figure 4.7-4.10 shows the component outputs and neural network outputs in frequency domain, where the four conditions are shown for eight time intervals from top to bottom.

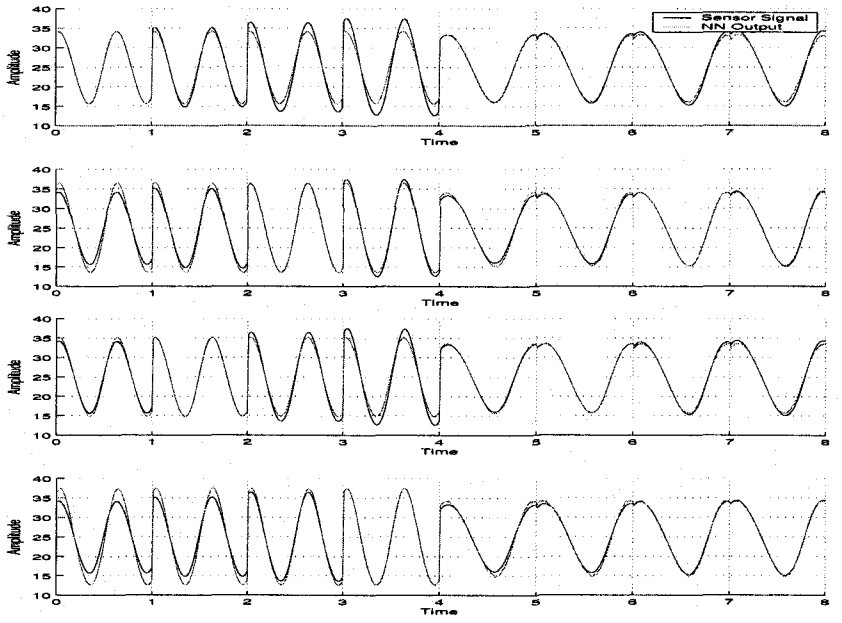


Figure 4.3: Responses of the mixing valve angle from the component and neural network in time domain.

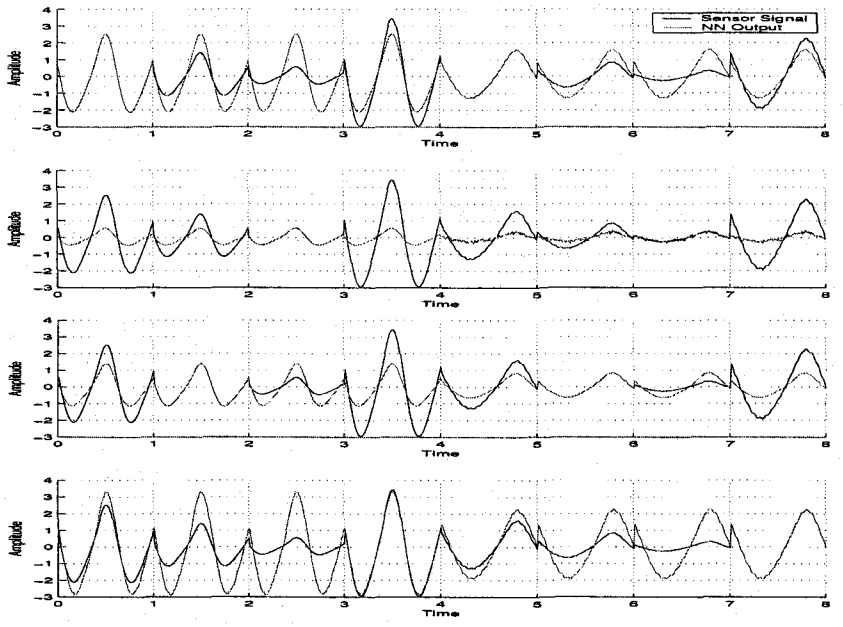


Figure 4.4: Responses of the anticipator from the component and neural network in time domain.

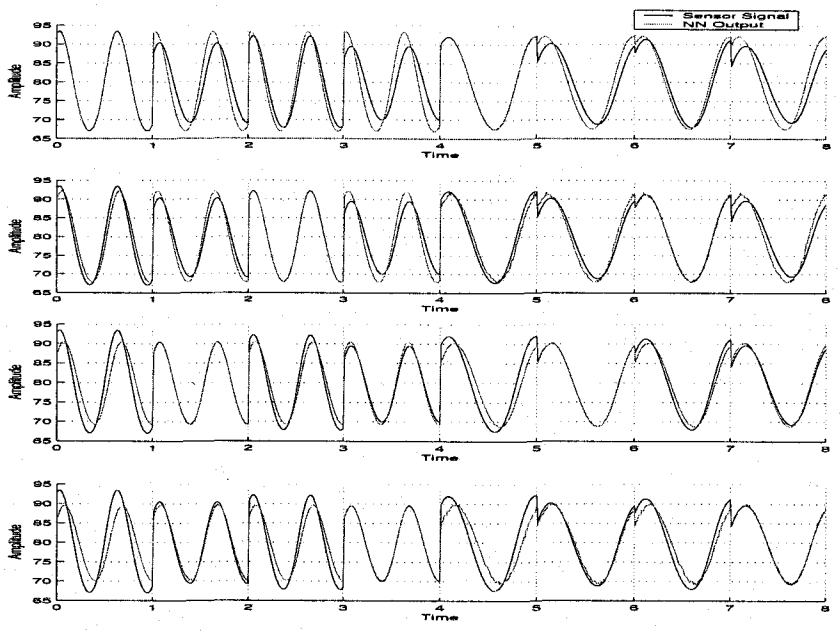


Figure 4.5: Responses of the duct air temperature sensor from the component and neural network in time domain.

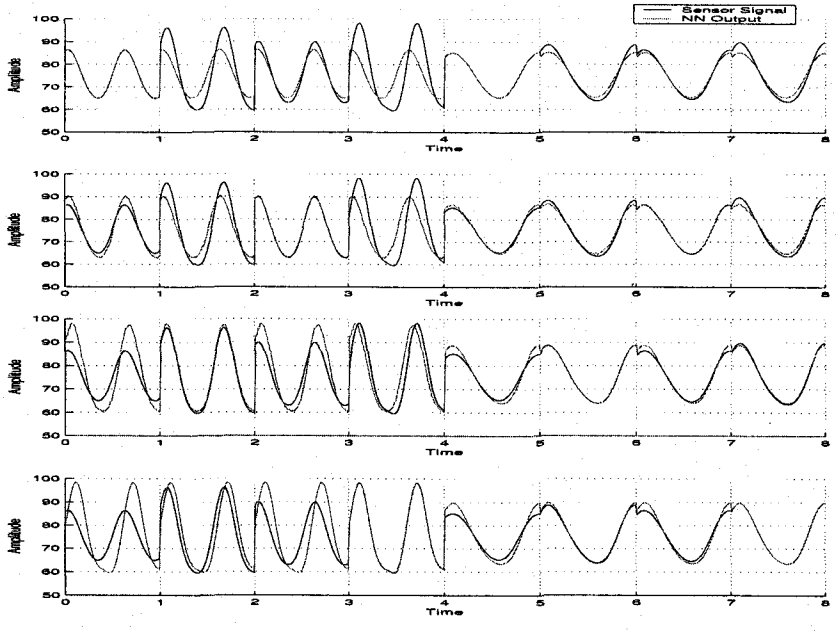


Figure 4.6: Responses of the cabin air temperature sensor from the component and neural network in time domain.

Table 4.3: Training error between desired outputs and the neural network outputs for cabin air temperature sensor.

	Inputs 1 (Time 1 to 249)	Inputs 2 (Time 250 to 500)
NM	0.014518	0.0093919
F1	0.0061599	0.0031889
F2	0.048701	0.024153
F3	0.033858	0.024203

Figure 4.11 shows the 3-dimensional representation of frequency domain analysis for normal condition as an example. The FFT amplitude of the neural network output for all the conditions are simultaneously compared with the plant outputs. When streams of sensor signals come from a sensor mounted on or in a plant, the data at each time division is collected and their magnitude is computed through FFT for each time interval. The number of data available in a time should reasonably be large to avoid false alarm. In the simulated example, each time period has 100 time domain data.

With the MSE between the actual plant output and the neural network output in time domain and frequency domain respectively, MMSE successfully shows the condition at each time interval as represented in Table 4.4.

The experiment with EKF and standard BP showed that NDEKF converged in much less iterations to meet the criterion (MSE of 0.1) as expected. It took 2 iterations for NDEKF, 62 iterations for EKF, and 147 iterations for standard BP.

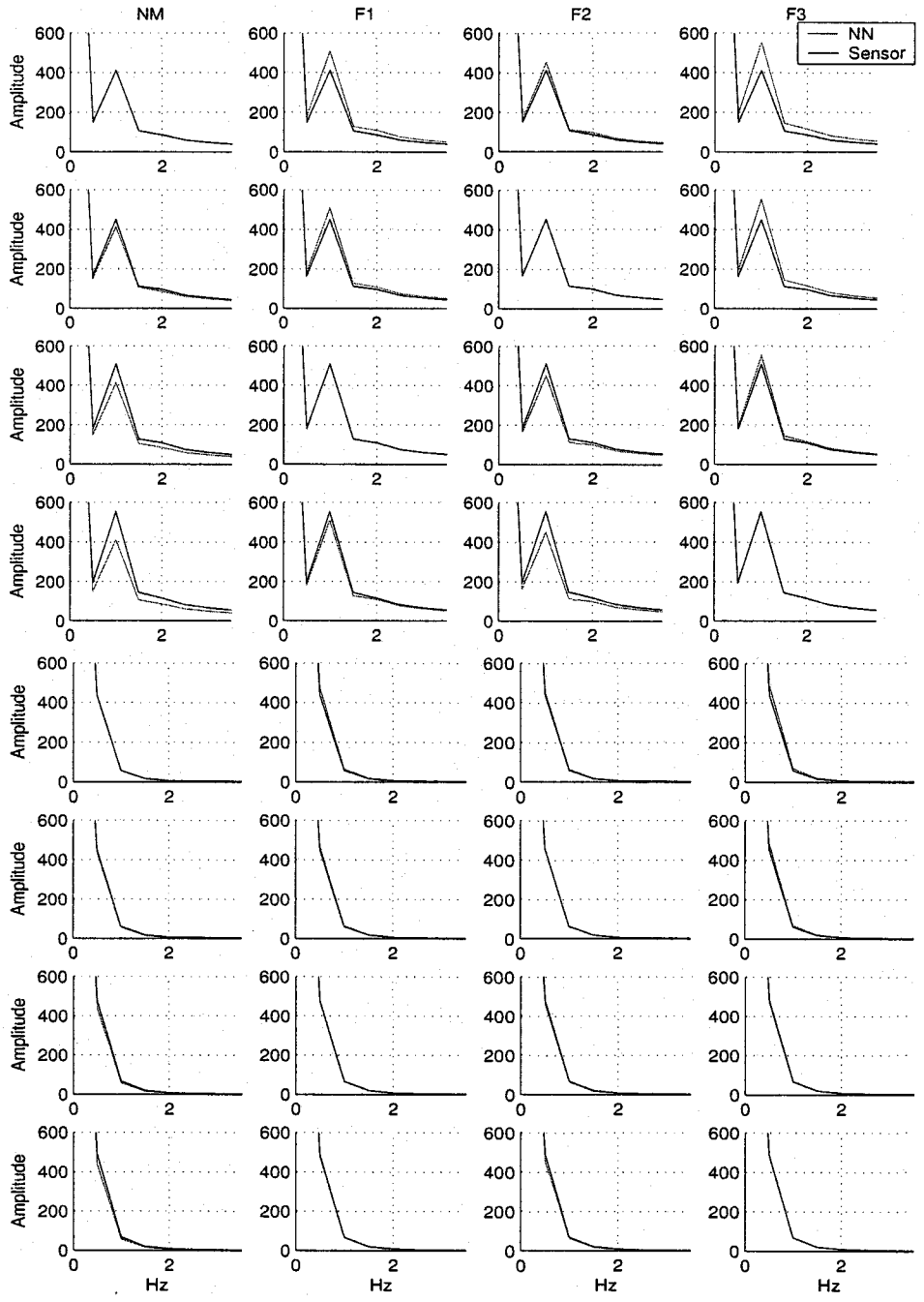


Figure 4.7: Responses of the mixing valve angle from the component and neural network in frequency domain.

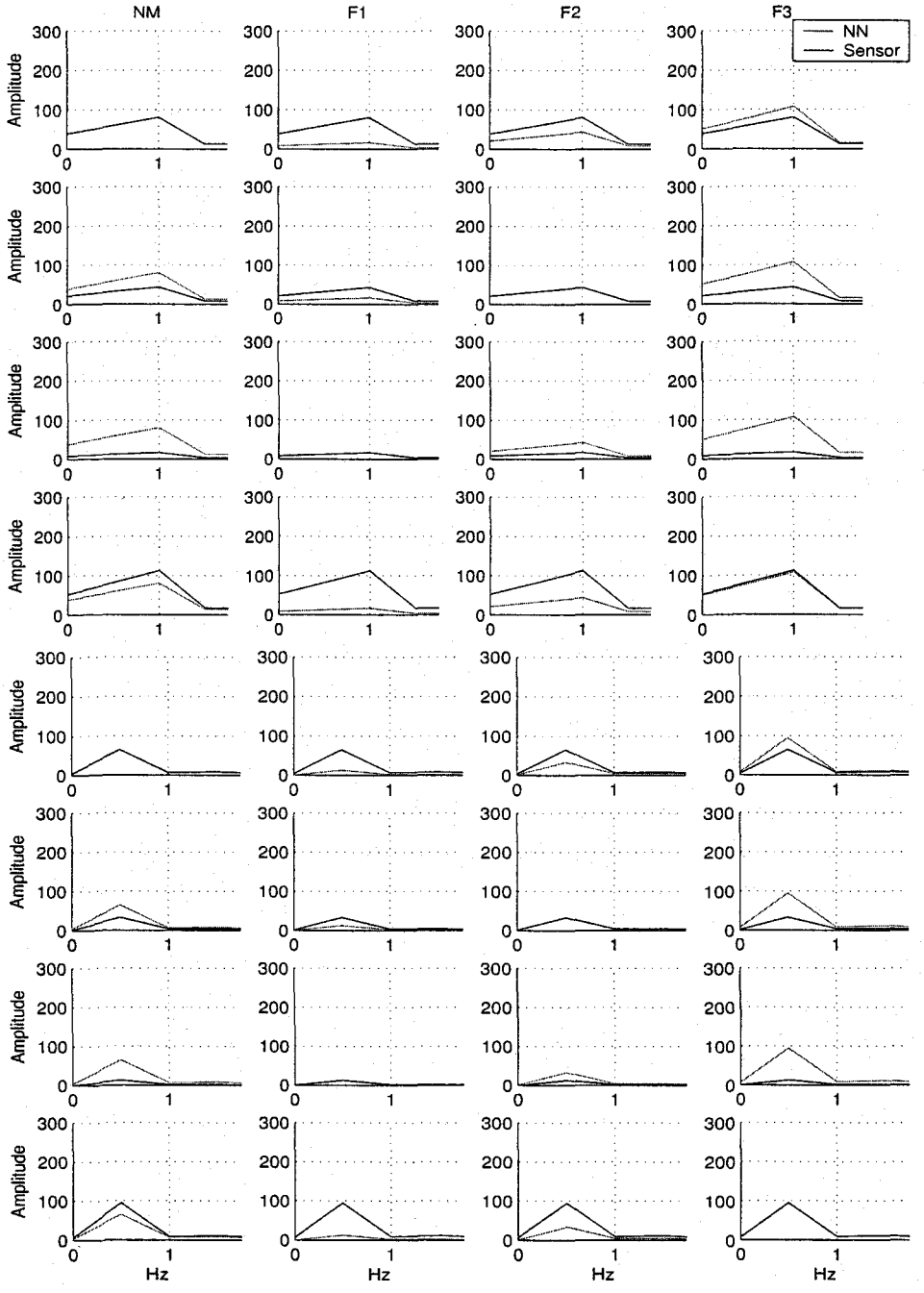


Figure 4.8: Responses of the anticipator from the component and neural network in frequency domain.

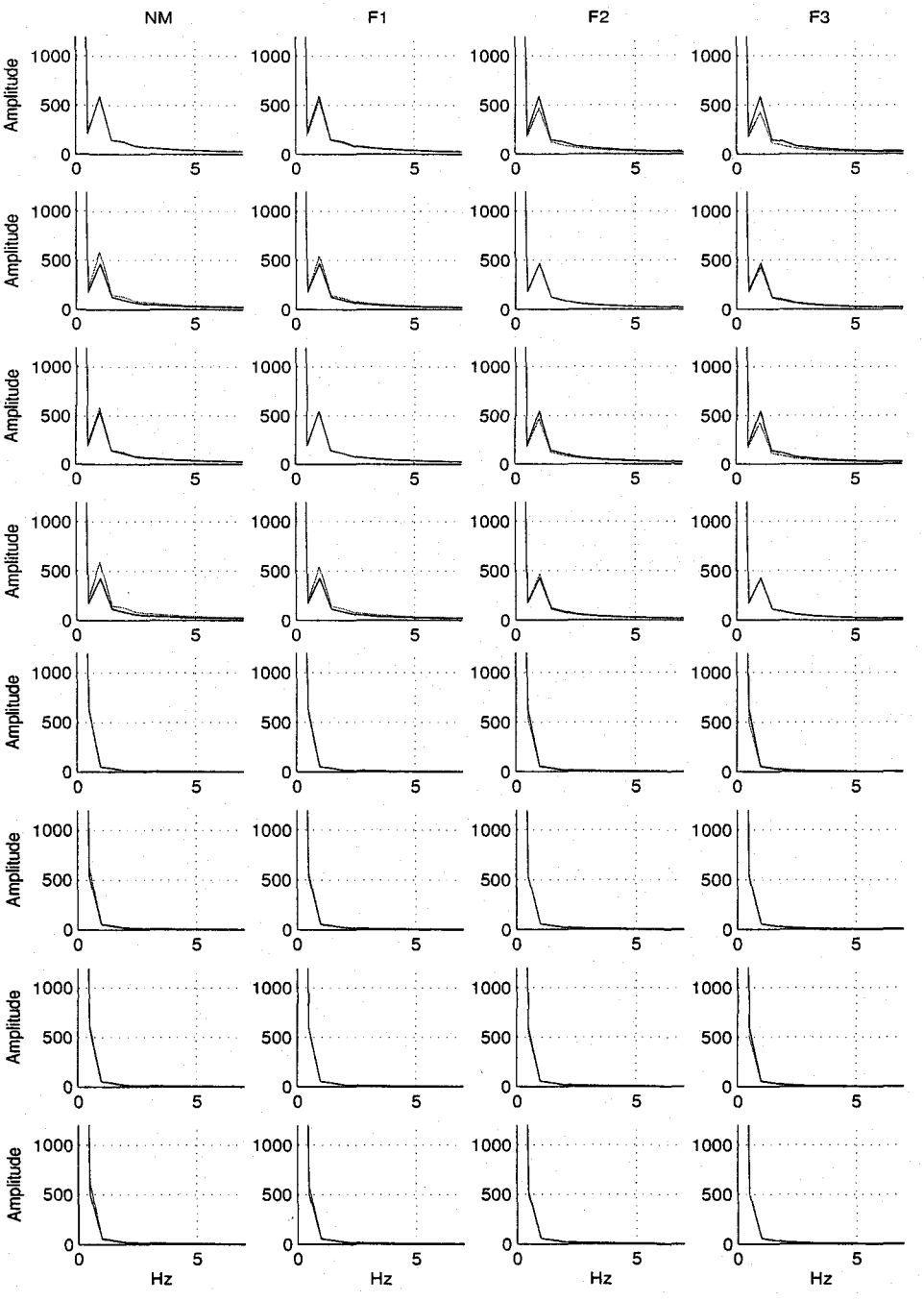


Figure 4.9: Responses of the duct air temperature sensor from the component and neural network in frequency domain.

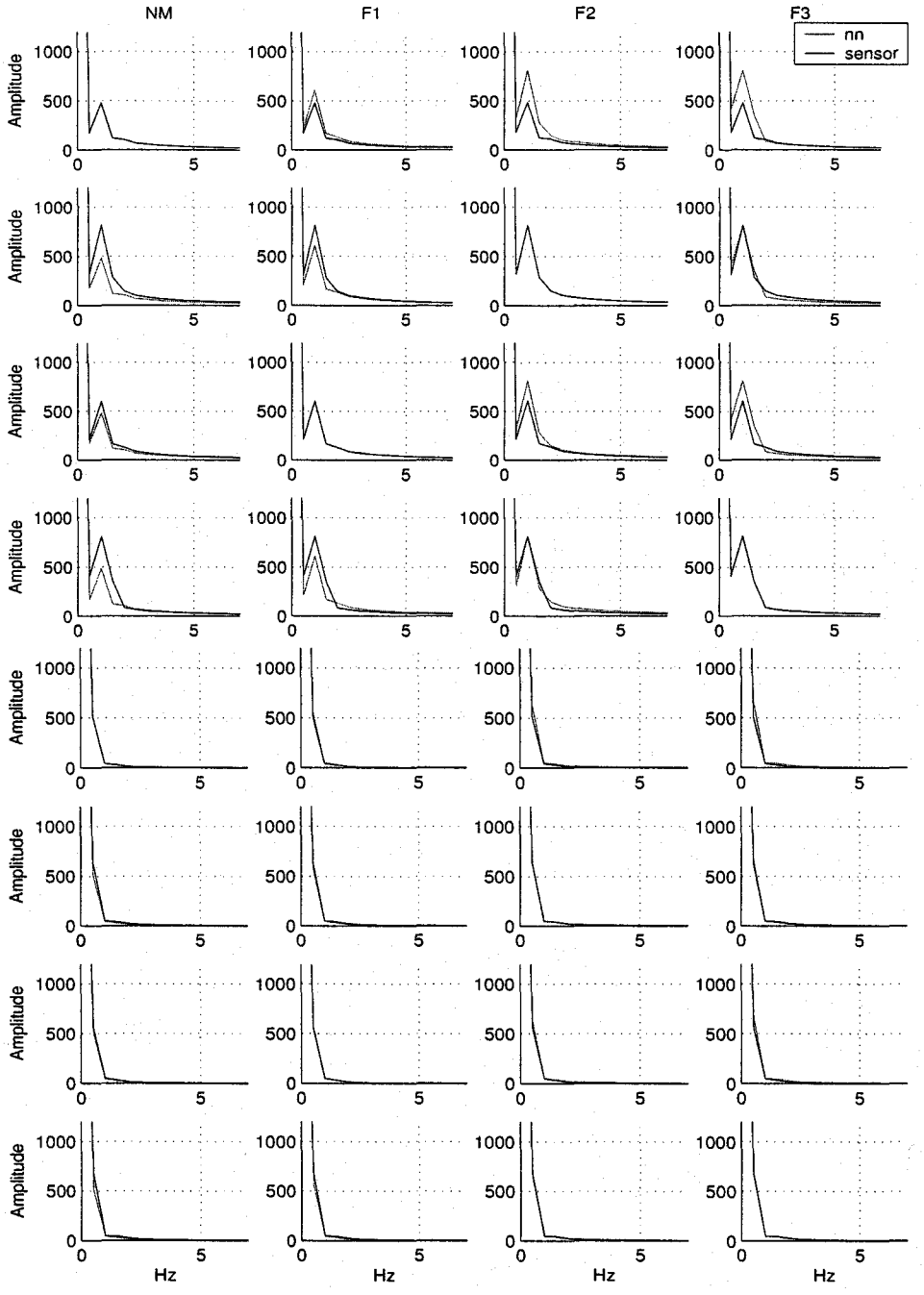


Figure 4.10: Responses of the cabin air temperature sensor from the component and neural network in frequency domain.

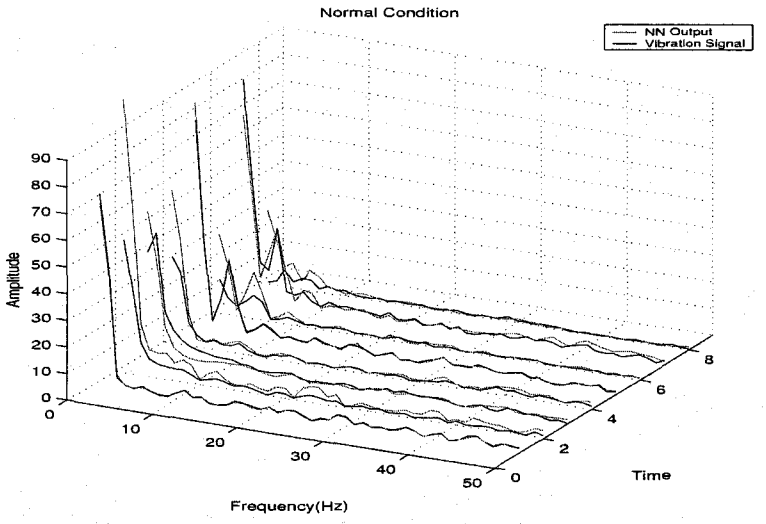
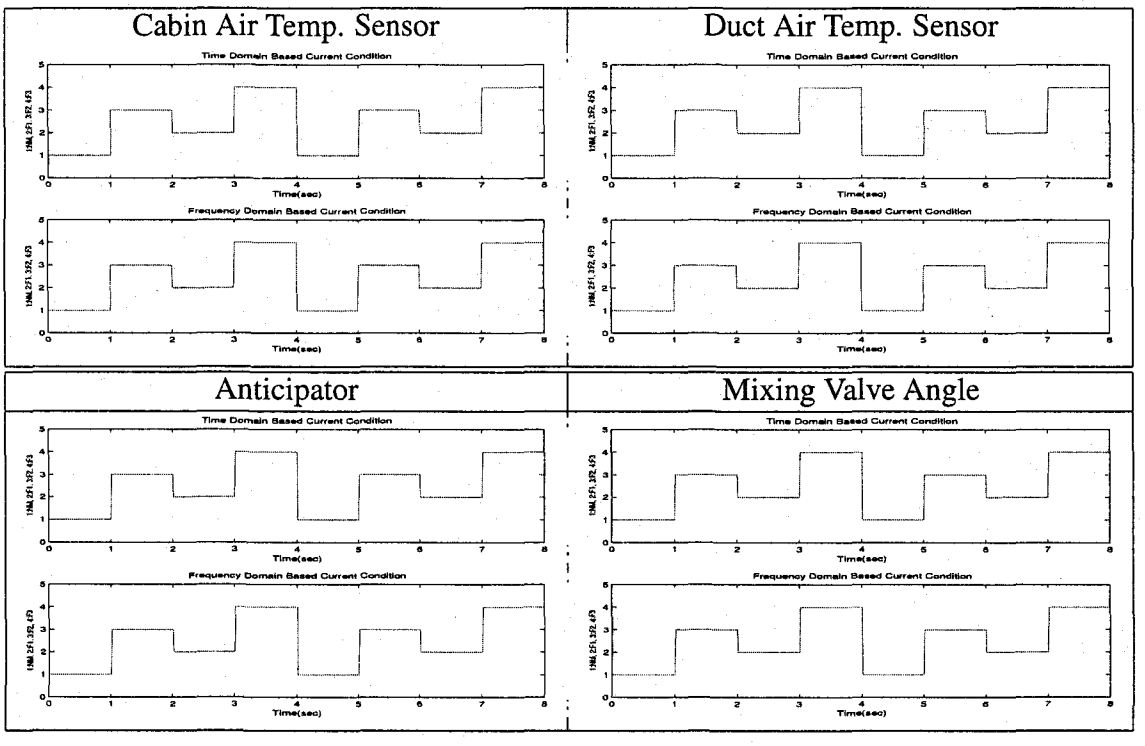


Figure 4.11: The 3-dimensional representation of frequency domain analysis.

Table 4.4: The current conditions of four components from time 1 to 8 based on time and frequency domains.



Chapter 5

Conclusions and Further Research

5.1 Discussion

In this thesis, the advantages of the neural network with the NDEKF method with its application in diagnosing through system identification technique both in time and frequency domain are introduced. The use of the NDEKF substantially reduces the computational complexity and memory requirements, in which EKF has disadvantages. The proposed FDI techniques are also effective when suitable process models are not available or when it is too complicated to be extracted from the data. The results show the capability of diagnosing in both time domain and frequency domain, where the choice of the domain is component dependent as each has its own advantages. The introduced techniques were demonstrated on the cabin temperature control system with parameters specifically determined for Boeing 767 aircraft as a real application example.

This task was developed by means of a number of intermediate stages to be achieved:

1. To present a general framework and background idea for neural network and Kalman filtering algorithm.

2. To show a node-decoupled extended Kalman filtering (NDEKF) algorithm with the its idea and the effectiveness in the reduction of computer complexity and memory requirements.

3. To introduce the failure detection and identification scheme through the neural network based system identification in both time and frequency domain.

4. To present the mechanism of how a standard cabin temperature control system for current aircrafts.

5. To show the feasibility of the proposed machine condition monitoring system through simulation.

The results presented in the simulation indicate that these goals have been met and that the overall objective of the thesis has been achieved.

5.2 Further Research

The proposed machine condition monitoring system have been used to monitor the current condition of a system. However, the proposed technique can also be applied and combined as prognosis schemes that are based on Kalman filtering based [20], wavelet neural network [21], and the nonlinear combination of neural network, Kalman filtering, and wavelet network [22].

In this thesis, the cabin temperature control system model is used as environmental control system (ECS) application example, but the proposed technique can also be applied to other models such as pressure regulators, flow controllers, and cabin pressure controllers.

Bibliography

[1] Haykin, Simon; *Kalman Filtering and Neural Network*, John Wiley & Sons, Inc., New York, 2001.

[2] Murtuza, Syed; Chorian, Steven; *"Node Decoupled Extended Kalman Filter Based Learning Algorithm For Neural Networks"*, IEEE International Symposium on Intelligent Control, August, 1994

[3] Maybeck, Peter; *Stochastic models, estimation, and control; Vol. 1*, Academic Press 1979

[4] Narendra, K.S.; Parthasarathy, K.; *"Identification and Control of Dynamical Systems Using Neural Networks"*, IEEE Transactions on Neural Networks Volume: 1 Issue 1, Mar 1990

[5] Welch, Greg; Bishop Gary; *"An Introduction to the Kalman Filter"*, Siggraph 2001, University of North Carolina at Chapel Hill

[6] Ruchti, T.L.; Brown, R.H.; Garside, J.J.; *"Kalman based artificial neural network training algorithms for nonlinear system identification"*, Intelligent Control, 1993., Proceedings of the 1993 IEEE International Symposium on, 25-27 Aug 1993 Page(s): 582 -587

[7] Dash, P.K.; Liew, A.C.; Ramakrishna, G.; *"Power-demand forecasting using a neural network with an adaptive learning algorithm"* Generation, Transmission and Distribution, IEE Proceedings- , Volume: 142 , Issue: 6 , Nov. 1995 Pages:560 - 568

[8] Saad, E.W.; Prokhorov, D.V.; Wunsch, D.C.; *"Advanced neural network training methods for low false alarm stock trend prediction"* Neural Networks, 1996., IEEE International Conference on , Volume: 4 , 3-6 June 1996 Pages:2021 - 2026 vol.4

[9] Frelicot, C.; Dubuisson, B.; *"K-step ahead prediction in fuzzy decision space-application to prognosis"* Fuzzy Systems, 1992., IEEE International Conference on , 8-12 March 1992 Pages:669 - 676

[10] Iwao, O.; *"Dynamic prediction of traffic volume through Kalman filtering theory"* Transportation Research, Vol.18B, no.2,1984,pp.1-11

[11] Puskorius, G.V.; Feldkamp, L.A.; *"Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks"*, Neural Networks, IEEE Transactions on, Volume: 5 Issue: 2, Mar 1994 Page(s): 279 -297

[12] Puskorius, G.V.; Feldkamp, L.A.; *"Model reference adaptive control with recurrent networks trained by the dynamic DEKF algorithm"*, Neural Networks, 1992. IJCNN., International Joint Conference on, Volume: 2, 7-11 Jun 1992 Page(s): 106 -113 vol.2

[13] Iiguni, Y.; Sakai, H.; Tokumaru, H.; *"A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter"*, Signal Processing, IEEE Transactions on, Volume: 40 Issue: 4, Apr 1992 Page(s): 959 -966

[14] Jin, L.; Nikiforuk, P.N.; Gupta, M.M.; *"Decoupled recursive estimation training and trainable degree of feedforward neural networks"* Neural Networks, 1992. IJCNN., International Joint Conference on , Volume: 1, 7-11 Jun 1992 Page(s): 894 -900 vol.1

[15] Li, Bo; Chow, M., Tipsuwan, Y.; Hung, J.; *"Neural-Network-Based Motor Rolling Bearing Fault Diagnosis"* Industrial Electronics, IEEE Transactions on, Volume: 47 Issue: 5, October 2000 Page(s): 1060 -1069

[16] Purkait, P.; Chakravorti, S.; *"Time and Frequency Domain Analyses Based Expert System for Impulse Fault Diagnosis in Transformers"* Dielectrics and Electrical Insulation, IEEE Transactions on, Volume: 9 No. 3, June 2000 Page(s): 433 -445

[17] Dallet, D.; Saliba, E.; *"Time-Frequency Analysis for Multistage A/D Converter Diagnosis"* Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE , Volume: 1 , 24-26 May 1999

[18] Betta, G.; Liguori, C.; Paolillo, A.; Pietrosanto, A.; *"A DSP-based FFT-analyzer for the fault diagnosis of rotating machine based on vibration analysis"* Instrumentation and Measurement, IEEE Transactions on , Volume: 51 , Issue: 6 , Dec. 2002 Pages:1316 - 1322

[19] Gillespie, B.W.; Atlas, L.E.; *"Optimizing time-frequency kernels for classification"* Signal Processing, IEEE Transactions on, Volume: 49 , Issue: 3 , March 2001 Pages:485 - 496

[20] Iwao, Okutani; *"Dynamic prediction of traffic volume through Kalman filtering theory"* Transportation Research, vol. 18B, no.2, 1984, Pages: 1-11

[21] Vachtsevanos,G.; Wang, P.; *"Fault Prognosis using Dynamic Wavelet Neural Networks."* AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference , 20-23 Aug. 2001 Pages:857 - 870

[22] Li, Sheng; *"Nonlinear Combination of Travel-time Prediction Model Based on Wavelet Network"* Intelligent Transportation Systems, IEEE International conference on, 2002 Pages: 741-746

[23] Boeing, Co.; *"Environmental Control System Library User Guide"* March 1999 Pages: 7.32-8.7

[24] Li, Sheng; *"The Airlinere Cabin Environment and the Health of Passengers and Crew"* National Academy of Sciences, National Academy Press. Pages: 53-61

[25] Puskorius, G.V.; Feldkamp, L.A.; *"Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on , Volume: i , 8-14 July 1991 "* National Academy of Sciences, National Academy Press. Pages: 771 - 777