MODELLING OF TWIN ROTOR MIMO SYSTEM (TRMS)

A PROJECT THESIS SUBMITTED IN THE PARTIAL FUFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL ENGINEERING

BY

ASUTOSH SATAPATHY

(10602067)

RASHMI RANJAN NAYAK

(10602060)

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

MAY 2010

MODELLING OF TWIN ROTOR MIMO SYSTEM (TRMS)

A PROJECT THESIS SUBMITTED IN THE PARTIAL FUFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY IN

ELECTRICAL ENGINEERING

BY

ASUTOSH SATAPATHY

(10602067)

RASHMI RANJAN NAYAK

(10602060)

UNDER THE GUIDANCE OF

PROF. B D SUBUDHI

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

MAY 2010

NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

CERTIFICATE

This is to certify that the thesis entitled " Modelling Of Twin Rotor MIMO System (TRMS)" being submitted by Sri Asutosh Satapathy & Sri Rashmi Ranjan Nayak to the National Institute of Technology, Rourkela (India) , for the partial fulfilment of requirement of the degree of Bachelor in Technology in Electrical Engineering , is an authentic record of research work carried out by them under my supervision and guidance and the work incorporated in this thesis has not been, to the best of my knowledge, submitted to any other University or Institute for the award of a degree or diploma.

Place : Rourkela

Date : 07/05/10

Prof. B. D. SUBUDHI
Department of Electrical Engg.
NIT Rourkela

ACKNOWLEDGEMENT

The authors express their deep sense of gratitude to the thesis supervisor Dr. Bidyadhar Subudhi, Professor & Head, Electrical Engg. Department, for his invaluable encouragement, helpful suggestions and supervision throughout the course of this work. His willingness to devote time and attention amidst numerous responsibilities is gratefully acknowledged. Above all his patience and optimist attitude could lead to carry out this work.

(Asutosh Satapathy)
Roll No.-10602067
Electrical Engg,
National Institute of Tech,
Rourkela.

(Rashmi Ranjan Nayak)
Roll No.-10602060
Electrical Engg,
National Institute of Tech,
Rourkela.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Modeling of a complex air vehicle such as a helicopter is very challenging task. This is because of the high non-linearity, significant cross-coupling between its two axes, complex aerodynamics and the inaccessibility of some of its states and outputs for measurements. It is possible to conceive a similar situation in the laboratory with the help of Twin Rotor MIMO System (TRMS).

While development of the analytical model of the TRMS, various components of the system have been modeled individually and then combined. The various responses of the system models have been compared with that of the real time setup.

The project is aimed at devising a model of the non-linear MIMO system by using Neural Networks. This is because of the efficient modeling approach provided by neural networks for highly non-linear systems. The project utilizes Feedback Instruments manufactured TRMS for capturing the Input-Output parameters i.e. control voltage, yaw & pitch angles, rotor current and position. These data are exploited to train the neural network models. This project also compares the efficiency of the two methods of identification.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background

1.2 System Identification

1.1 BACKGROUND

The TRMS model comprises of a beam pivoted on its base in such y that it rotates freely both in the horizontal and vertical planes. There are two rotors, the main and tail ones at both ends of the beam, which are driven by DC motors. At the pivot a counterbalance arm with a weight at its end is fixed to the beam. The state of the beam is described by four process variables: horizontal and vertical angles which are measured by encoders fitted at the pivot, and two corresponding angular velocities. [1]

There are also two other state variables, the angular velocities of the rotors, which are measured by speed sensors coupled with the driving DC motors. The basic difference in the laboratory set-up and in an actual helicopter is that the aerodynamic force is controlled by changing the angle of attack in a helicopter while in the laboratory set-up the angle of attack is fixed. By varying the speed of the two rotors the aerodynamic force can be controlled in a TRMS Model. As each rotor affects both the position angles, significant cross-coupling can be observed between actions of the rotors. To stabilize the TRMS, the design of the controllers is based on decoupling. The TRMS system has been designed to operate with external, PC-based digital controller. Communication with the position, speed sensors and motors is done by the control computer via a dedicated I/O board and power interface. A real time software operating in the MATLAB/Simulink RTW/RTWT environment controls the I/O board. [2]

1.2 SYSTEM IDENTIFICATION

System identification incorporates the mathematical tools and algorithms which are used for building dynamical models from measured data. Dynamical mathematical model implies that it is a mathematical description of the dynamic behaviour of a system in either the time or the frequency domain.

. System Identification is performed by measuring the behaviour of the system and the process inputs to the system to try and formulate a mathematical relation between them without taking into account the internal processes of the system.

For non-linear systems, system modelling is done by assuming a model structure beforehand and then estimating the model parameters. We can either specialize the model structure for a particular purpose or have a general one that can be used for other devices too. The complexity of the model is determined by the parameter estimation.

CHAPTER 2

DESCRIPTION

2.1 TRMS DESCRIPTION

As shown in Figure 2.1, the TRMS mechanical unit comprises of two rotors positioned on a horizontal beam with a counterbalance at the pivot. The whole unit is attached to the tower which ensures safe helicopter control experiments.

Fig. 2.1 *TRMS Mechanical Unit* [4]

Along with the mechanical unit, the electrical unit which is placed under the tower plays an important role for TRMS control. Its function is to allow measured signals to be transferred to the PC and control signal applications via an I/O card. The mechanical and electrical units provide a complete control system setup presented in Figure 2.2. [4]

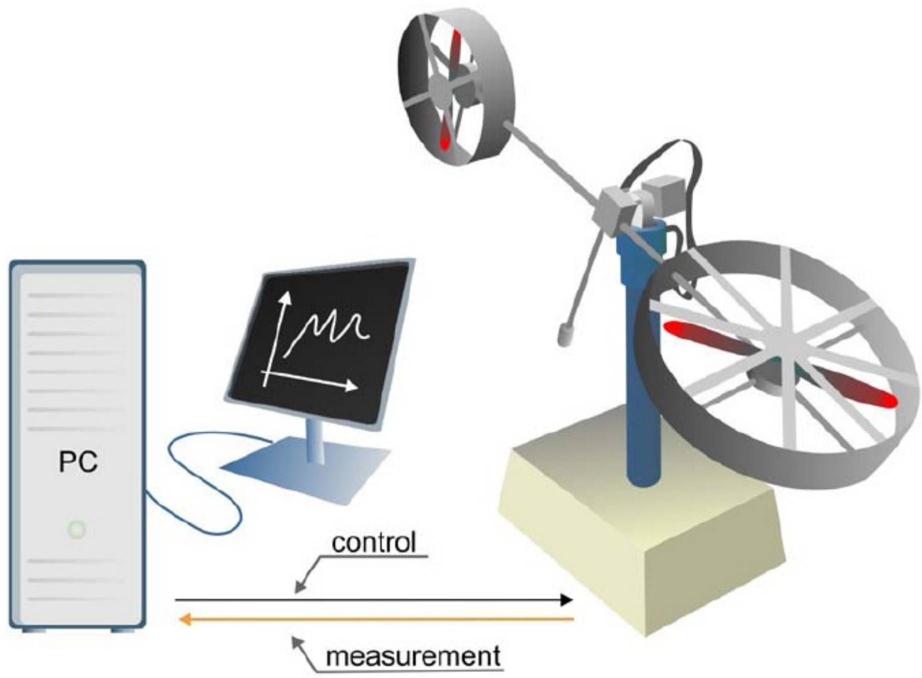

Fig. 2.2 *TRMS Control System* [4]

2.2 TRMS MATHEMATICAL MODEL

The mechanical-electrical model of the TRMS is presented in Figure 3.

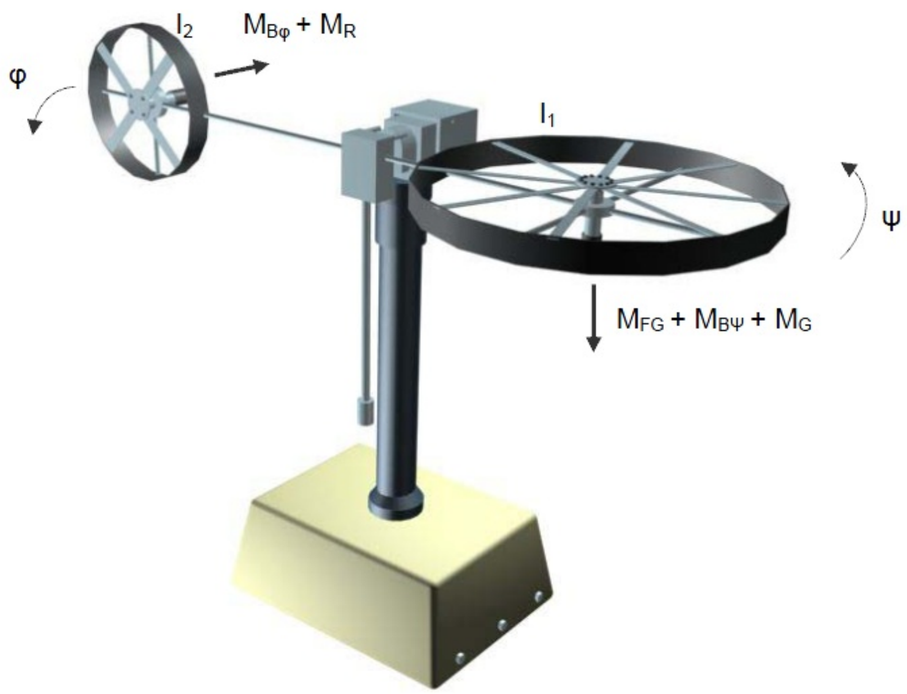

Fig. 2.3 *TRMS phenomenological model* [4]

Usually, phenomenological models tending to be nonlinear, which means that at least one of the states (i – rotor current, θ – position) is an argument of a nonlinear function. So as to present such a model as a transfer function (a form of linear plant dynamics representing a control system), it has to be linearised. As shown in the electrical-mechanical diagram in Figure 2.3 the following non-linear model equations can be derived. [4]

$$I_1 \cdot \ddot{\psi} = M_1 - M_{FG} - M_{B\psi} - M_G, \quad (1)$$

where

$$M_1 = a_1 \cdot \tau_1^2 + b_1 \cdot \tau_1, \quad \text{– nonlinear static characteristic} \quad (2)$$

$$M_{FG} = M_g \cdot \sin\psi, \quad \text{– gravity momentum} \quad (3)$$

$$M_{B\psi} = B_{1\psi} \cdot \dot{\psi} + B_{2\psi} \cdot sign(\dot{\psi}), \quad \text{– friction forces momentum} \quad (4)$$

$$M_G = K_{gy} \cdot M_1 \cdot \dot{\varphi} \cdot \cos\psi. \quad \text{– gyroscopic momentum} \quad (5)$$

The motor momentum is described by an approximated first order transfer function in Laplace Domain:

$$\tau_1 = \frac{k_1}{T_{11}s + T_{10}} \cdot u_1. \quad (6)$$

Equations that refer to the horizontal plane motion are as follows:

$$I_2 \cdot \ddot{\varphi} = M_2 - M_{B\varphi} - M_R \quad (7)$$

where

$$M_2 = a_2 \cdot \tau_2^2 + b_2 \cdot \tau_2, \quad \text{– nonlinear static characteristic} \quad (8)$$

$$M_{B\psi} = B_{1\varphi} \cdot \dot{\psi} + B_{2\varphi} \cdot sign(\dot{\varphi}), \quad \text{– friction forces momentum} \quad (9)$$

M_R is the cross reaction momentum approximated by:

$$M_R = \frac{k_c(T_o s + 1)}{(T_p s + 1)} \cdot \tau_1. \quad (10)$$

Again the DC motor with the electrical circuit is given by:

$$\tau_2 = \frac{k_2}{T_{21}s + T_{20}} \cdot u_2. \quad (11)$$

The phenomenological model parameters having been chosen experimentally, makes the TRMS nonlinear model a semi-phenomenological model. The following table gives the approximate parameter values. [4]

Parameter	Value
I_1 - moment of inertia of verical rotor	$6.8 \cdot 10^{-2}\ kg \cdot m^2$
I_2 - moment of inertia of horizontal rotor	$2 \cdot 10^{-2}\ kg \cdot m^2$
a_1 - static characteristic parameter	0.0135
b_1 - static characteristic parameter	0.0924
a_2 - static characteristic parameter	0.02
b_2 - static characteristic parameter	0.09
M_g - gravity momentum	$0.32\ N \cdot m$
$B_{1\psi}$ - friction momentum function parameter	$6 \cdot 10^{-3} N \cdot m \cdot s/rad$
$B_{2\psi}$ - friction momentum function parameter	$1 \cdot 10^{-3} N \cdot m \cdot s^2/rad$
$B_{1\varphi}$ - friction momentum function parameter	$1 \cdot 10^{-1} N \cdot m \cdot s/rad$
$B_{2\varphi}$ - friction momentum function parameter	$1 \cdot 10^{-2} N \cdot m \cdot s^2/rad$
K_{gy} - gyroscopic momentum parameter	$0.05\ s/rad$
k_1 - motor 1 gain	1.1
k_2 - motor 2 gain	0.8
T_{11} - motor 1 denominator parameter	1.1
T_{10} - motor 1 denominator parameter	1
T_{21} - motor 2 denominator parameter	1
T_{20} - motor 2 denominator parameter	1
T_p - cross reaction momentum parameter	2
T_0 - cross reaction momentum parameter	3.5
k_c - cross reaction momentum gain	-0.2

Table 2.1 *TRMS Model Parameters* [4]

The limits of the control signal are set to [–2.5V- +2.5V].

2.3 TRMS SYSTEM SCHEMATIC

The TRMS is a Multiple Input Multiple Output (MIMO) plant. The simplified schematic of the TRMS is presented in Figure 2.4.

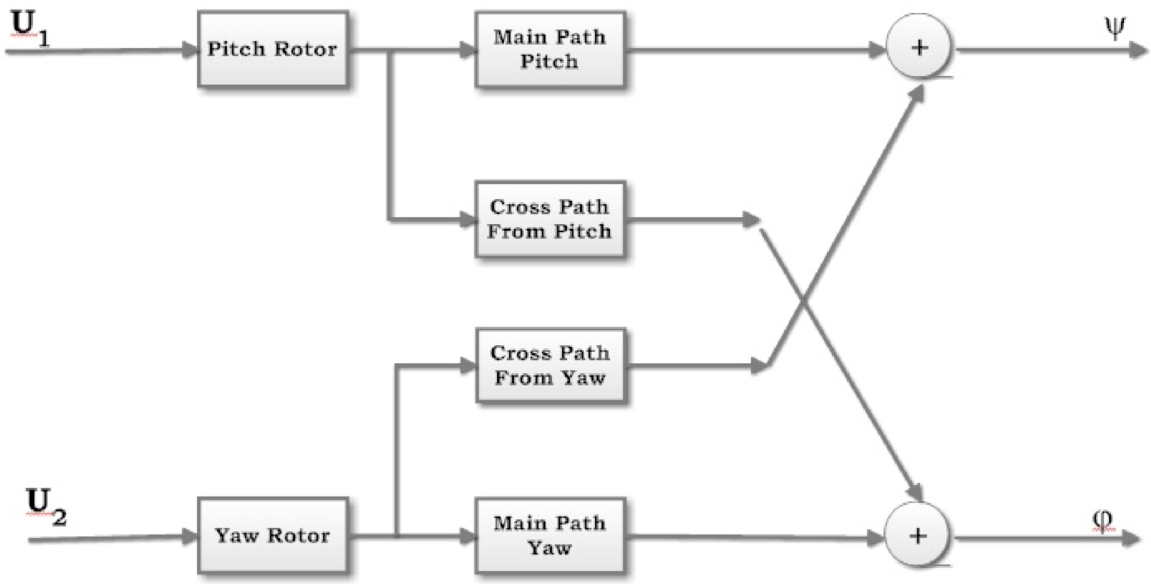

Fig. 2.4 *TRMS Simplified System Schematic*

The TRMS has two control inputs- U_1 and U_2. As it can be seen from Figure 2.4 the dynamics of cross couplings between the rotors are the key features of the TRMS. The position state variable of the beams is measured with the help of incremental encoders, which provides for a relative position signal. Setting proper initial conditions is hence important every time the Real-Time TRMS simulation is run. [4]

CHAPTER 3

MODEL IDENTIFICATION

3.1 TRMS MODEL IDENTIFICATION

As we already know from the previous section, there is significant cross-coupling between rotors in the nonlinear MIMO plant (shown in Figure 2.4. The model can be treated as two linear rotor models with two linear couplings in-between in order to keep the identification simple. Therefore four linear models have to be identified: two for the main dynamics path from U_1 to ψ and U_2 to φ and cross coupling dynamics paths from U1 to φ and U2 to ψ.

There are a few important things that has to be kept in mind when carrying out an identification experiment:

3.1.1 Stability Problem:

For an unstable plant the identification has to be carried out with a working controller, which introduces more problems that will be discussed in later sections. The identification is much simpler if the plant is stable and therefore we do not have to work with a controller.

3.1.2 Structure Choice:

Structure choice happens to be a very important aspect of the identification. It depends on the choice of the numerator and denominator order of the transfer function for linear models. It is applicable for both continuous and discrete systems. The structures are also divided in terms of the error term description: ARX, ARMAX, OE and BJ in the cases of the discrete model.

3.1.3 Sampling Time:

The sampling time choice is very important for both the identification and the control. It can neither be too short nor can it be too long. Because of the quantization effect introduced by the AD, the identification quality might be influenced by very short sampling time. Furthermore for smaller sampling time the software and hardware has to be faster and more memory is required. However elimination of aliasing effects will be allowed for short sampling time and thus introduction of anti aliasing filters will not be required. Inclusion of all of the dynamics will not be allowed for long sampling times.

3.1.4 Excitation Signal:

The choice is pretty much simple for the linear models the excitation. White noise is used quite frequently by designers. But it is often not allowed for industrial applications. White noise holds very broad frequency content so identification of the whole dynamics of the plant can be done easily. Thus white noise is quite attractive. Several sinusoids with different frequency levels can be added to produce a desired excitation signal if the dynamics are not too complex.

3.1.5 Identification Method:

The "Least Mean Square (LMS)" method and the "Instrumental Variable" method are the commonly used methods. The LMS method is the very much popular and is applied in Matlab. The error between the model and the plant output is minimized using this method. The optimal model parameters, for which the square of the error is minimal is the result of the identification. [4]

CHAPTER 4

IDENTIFICATION EXPERIMENTS

4.1 MAIN PATH PITCH ROTOR IDENTIFICATION

Introduction:

This model describes the relation between the control voltage U1 and the angle ψ. Generally all the real time simulations are carried out using a sampling time of Ts = 0.001 [s]. But since plant dynamic response is relatively slow, the identification of the discrete model is carried out with the sampling time of Ts = 0.1[s].

The identification experiment is carried out using the model called MainPitch_Ident.mdl in the Matlab Toolbox. The function of the model is to excite the TRMS and record its response. This excitation signal comprises of several sinusoids. Two signals are collected in the form of vectors and are available in the Workspace.

Task:

The identification experiment was conducted and data was collected. The model was identified using the Matlab identification interface. The graphs generated of the transient response, step response analysis, frequency response, pole and zeros map, and model residuals give a clear idea of the quality of the response.

Results:

The following results were obtained in the experiment conducted.

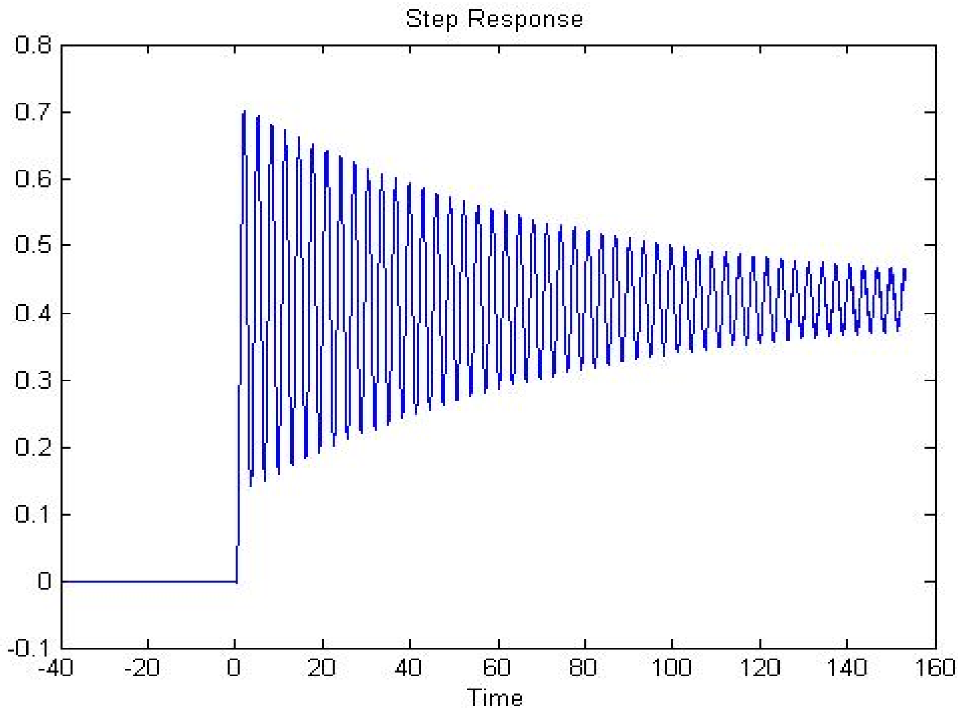

Fig. 4.1 *Step Response*

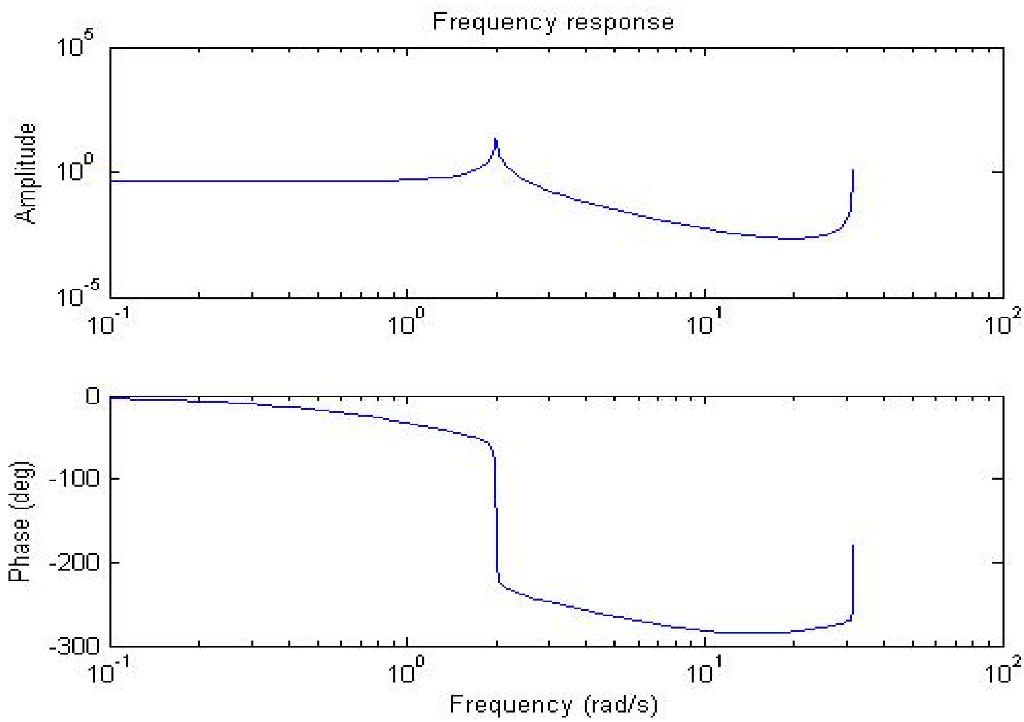

Fig. 4.2 *Frequency Response*

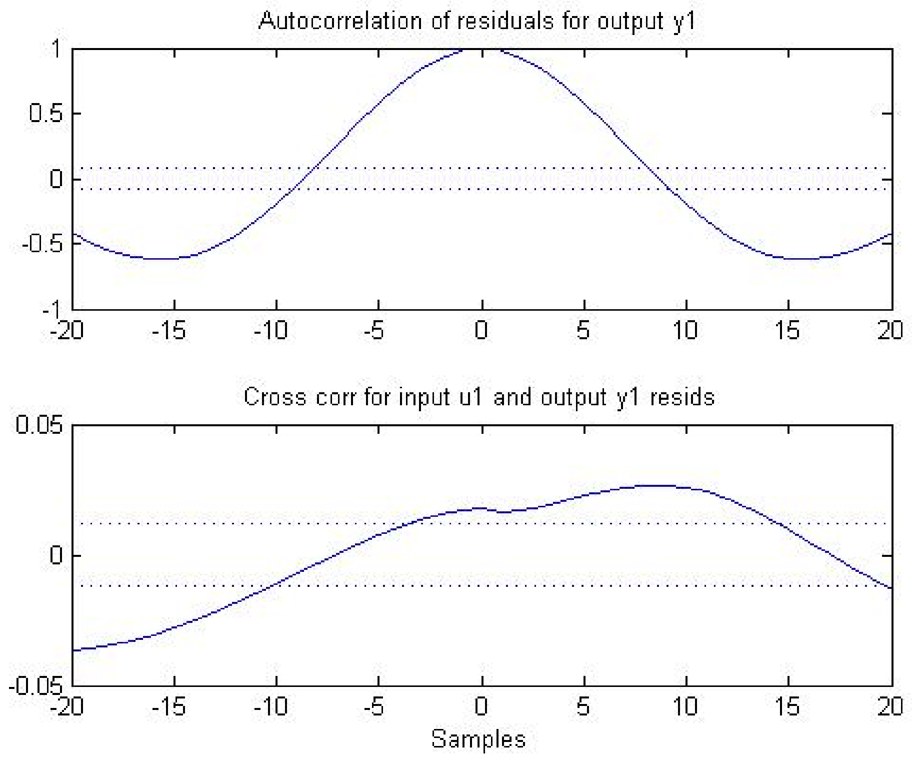

Fig. 4.3 *Autocorrelation of Residuals for Output y_1*

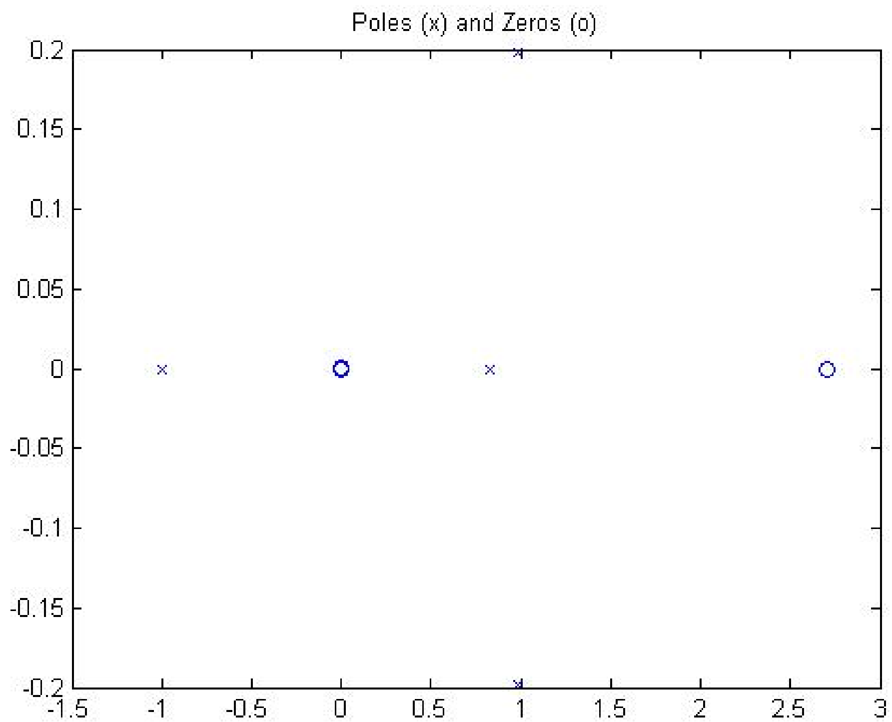

Fig. 4.4 *Poles & Zeroes*

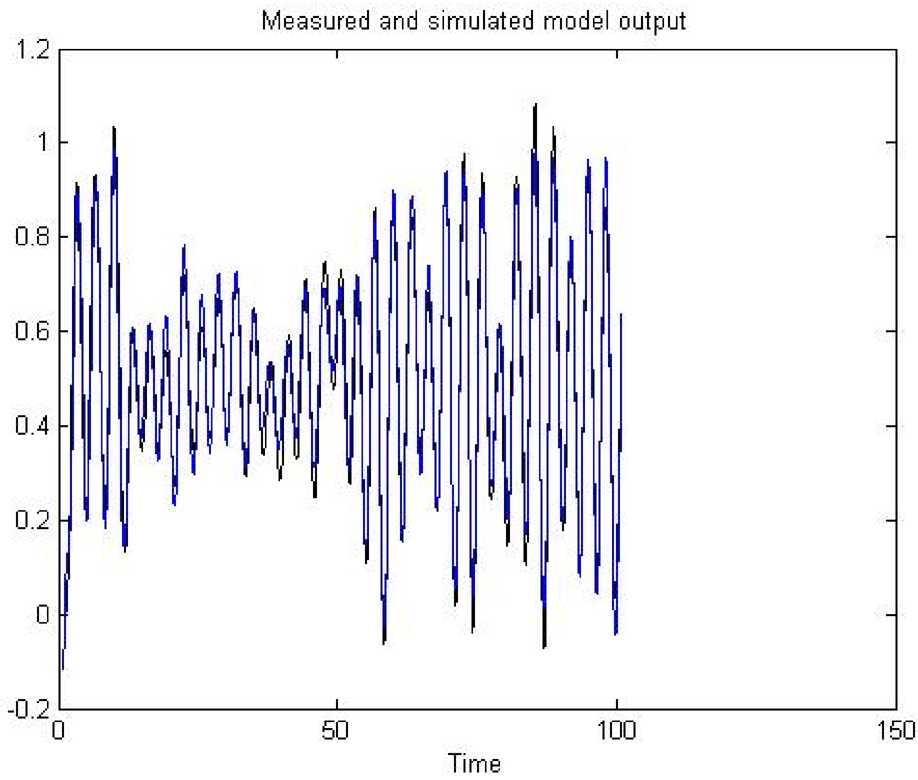

Fig. 4.5 *Measured & Simulated Model Output*

Discrete Transfer function thus obtained:

$$\frac{-0.003293\,z + 0.008894}{z^4 - 1.788\,z^3 - 0.1683\,z^2 + 1.8\,z - 0.8305}$$

Continuous Transfer function thus obtained:

$$\frac{0.02169\,s^4 \quad 0.8584\,s^3 \quad 19.89\,s^2 \quad 31.57\,s + 3039}{s^5 + 1.842\,s^4 + 990.9\,s^3 + 1855\,s^2 + 3977\,s + 7252}$$

Sampling time: 0.1 [s].

Discussions:

Figure 4.1 shows the Step Response of the Main Path Pitch Rotor Identification. It basically gives knowledge of how the system behaves in time when the inputs change from zero to one in a relatively short span of time. In this case the system is stable because it settles down to give a steady output by reaching another steady state in a short span of time. Fig 4.2 shows the Frequency Response of the system, which is the measure of a system's output spectrum with respect to its input signal. In this case a Bode Plot has been drawn to plot the magnitude (measured in dB) and the phase (measured in radians) versus frequency. Fig 4.3 shows the Autocorrelation of the output, which means that it is the cross-correlation of the output signal with itself observed as a function of a time lag with itself. Fig 4.4 is the Poles and Zeroes map which shows the position and number of poles and zeroes of the transfer function. If any of the position of these poles or zeroes were to be changed, then it would have great implications on the Step Response of the system. Fig 4.5 is the comparison between the measured and simulated output which is a comparison between the control and real time experiments. While the control experiments have been done in perfect setup, external factors come into picture in case of the real time experiment. In this case, the graphs being nearly similar, the real time experimental results have little error.

4.2 MAIN PATH YAW ROTOR IDENTIFICATION

Task:

The identification experiment was conducted using the MainYaw_Ident.mdl in Matlab and data was collected. The model was identified using the Matlab identification interface. The graphs generated of the transient response, step response analysis, frequency response, pole and zeros map, and model residuals give a clear idea of the quality of the response.

Results:

The following results were obtained in the experiment conducted.

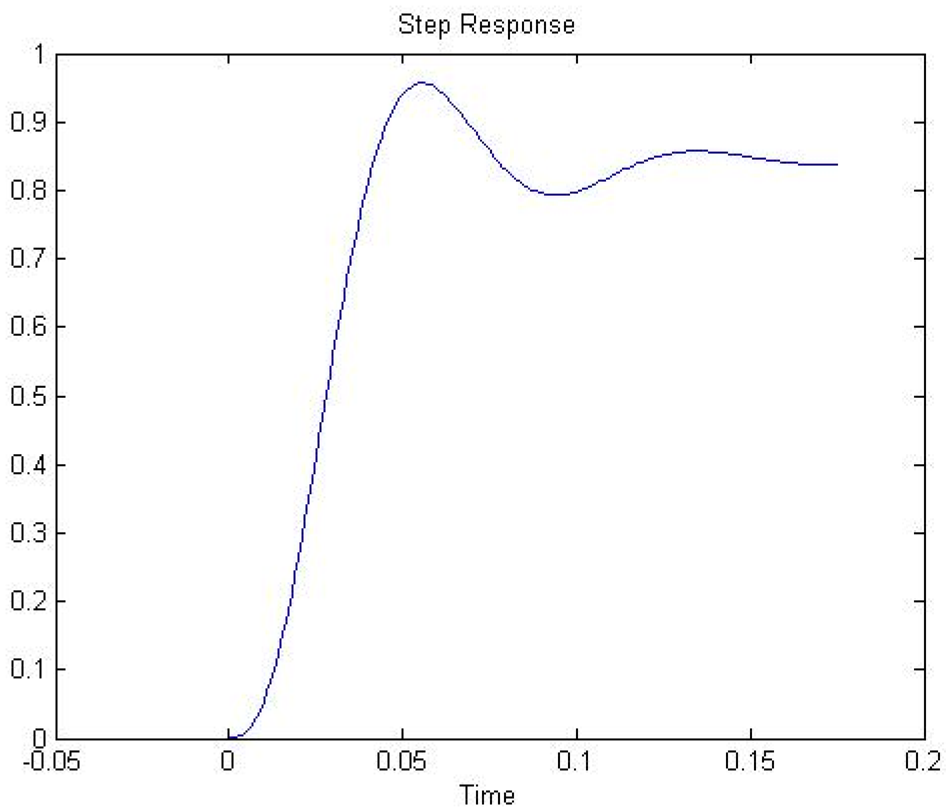

Fig. 4.6 *Step Response*

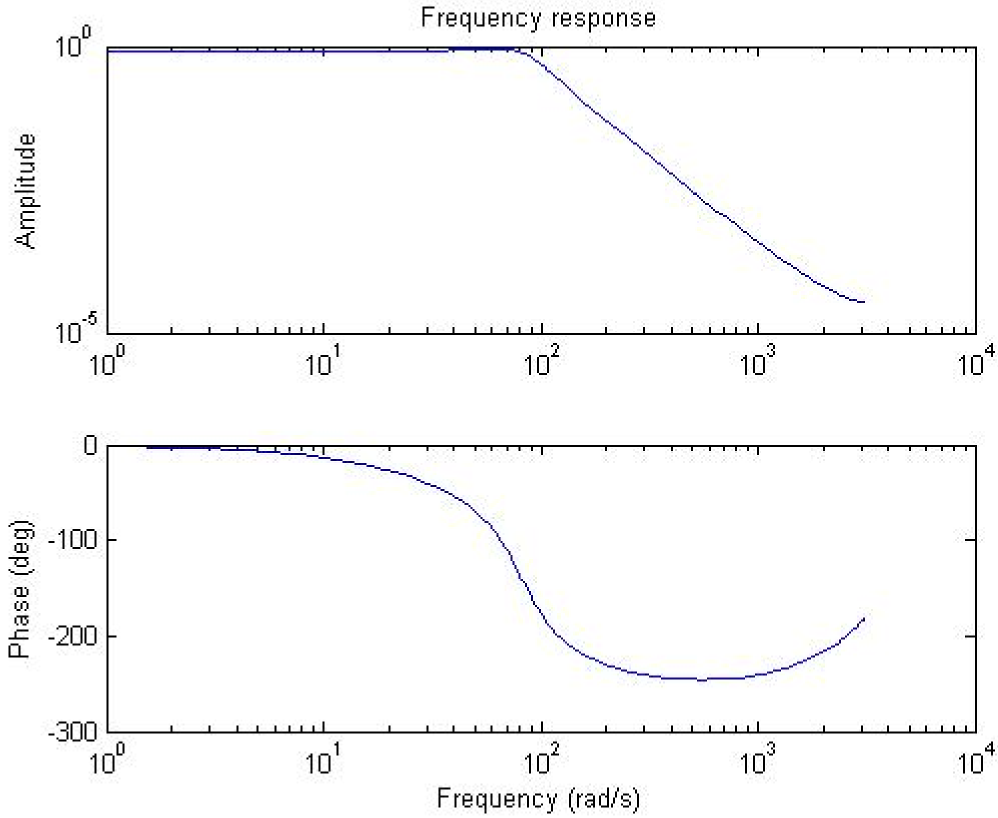

Fig. 4.7 *Frequency Response*

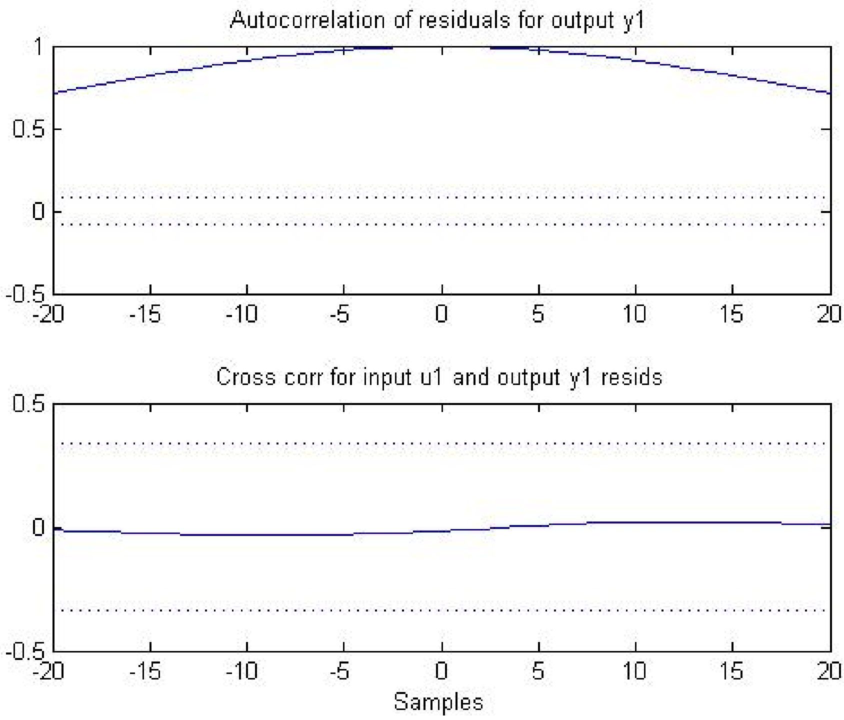

Fig. 4.8 *Autocorrelation of Residuals for Output y_1*

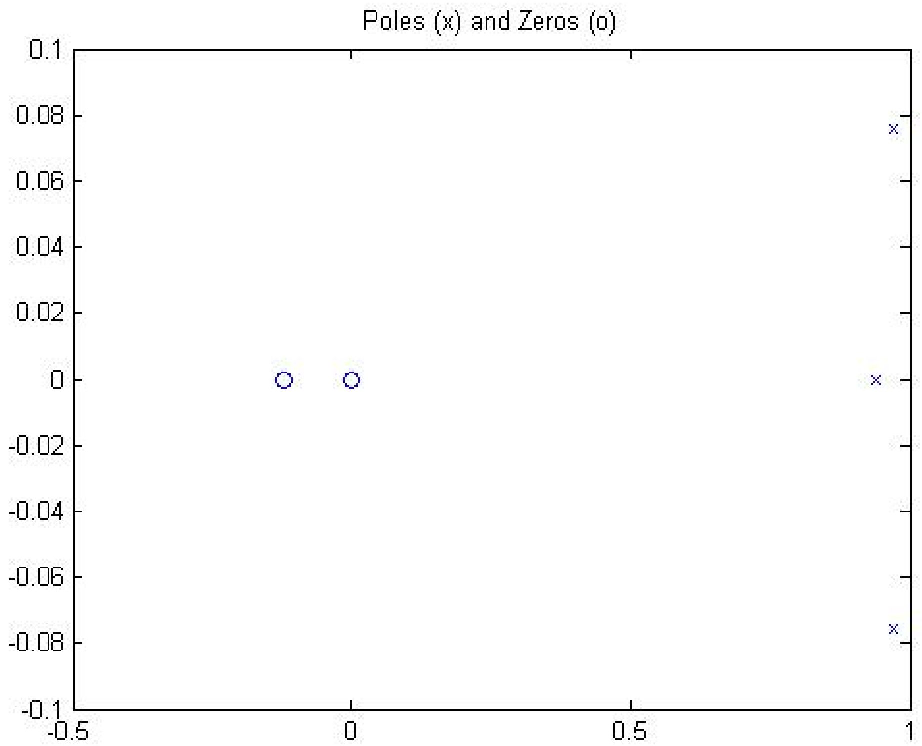

Fig. 4.9 *Poles & Zeroes*

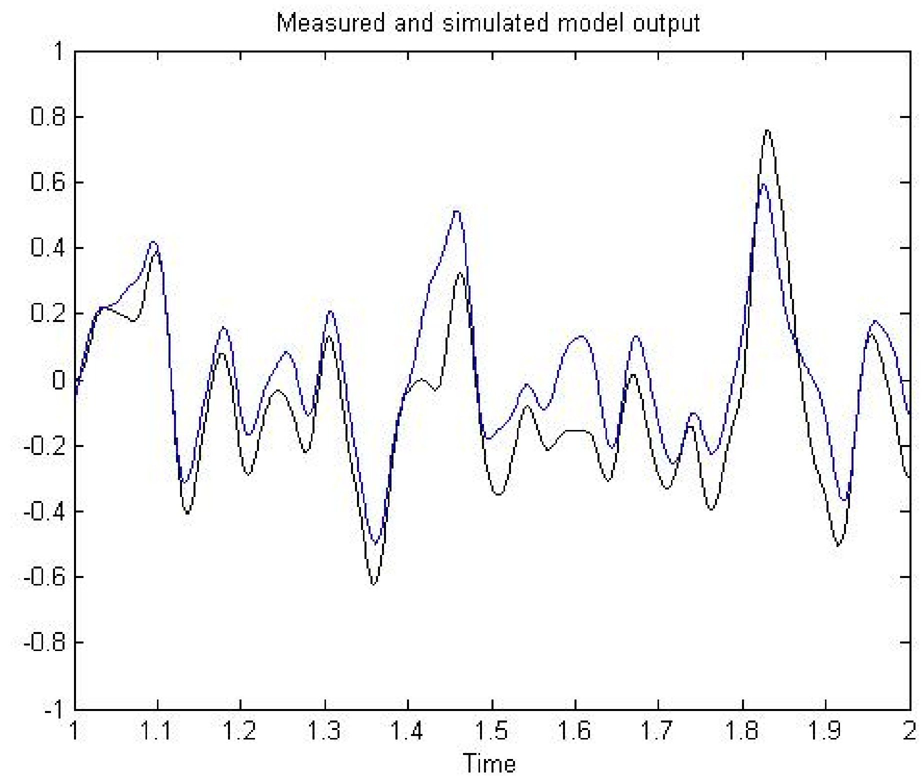

Fig. 4.10 *Measured & Simulated Model Output*

Discrete Transfer function thus obtained:

$$\frac{0.0003014\, z + 3.68e-005}{z^3 - 2.88\, z^2 + 2.771\, z - 0.89}$$

Continuous Transfer function thus obtained:

$$\frac{0.09701\, s^2 + 316.1\, s + 3.586e^{005}}{s^3 + 116.5\, s^2 + 1.02e004\, s + 4.26e^{005}}$$

Sampling time: 0.001 [s].

Discussions:

Fig 4.6 shows the Step Response of the Main Path Yaw Rotor Identification. It basically gives knowledge of how the system behaves in time when the inputs change from zero to one in a relatively short span of time. In this case the system is stable because it settles down to give a steady output by reaching another steady state in a short span of time. Fig 4.7 shows the Frequency Response of the system, which is the measure of a system's output spectrum with respect to its input signal. In this case a Bode Plot has been drawn to plot the magnitude (measured in dB) and the phase (measured in radians) versus frequency. Fig 4.8 shows the Autocorrelation of the output, which means that it, is the cross-correlation of the output signal with itself observed as a function of a time lag with itself. Fig 4.9 is the Poles and Zeroes map which shows the position and number of poles and zeroes of the transfer function. If any of the position of these poles or zeroes were to be changed, then it would have great implications on the Step Response of the system. Fig 4.10 is the comparison between the measured and simulated output which is a comparison between the control and real time experiments. While the control experiments have been done in perfect setup, external factors come into picture in case of the real time experiment. In this case, the graphs vary at some points; hence the real time experimental results have some error.

4.3 CROSS PATH PITCH ROTOR IDENTIFICATION

Introduction:

The identification experiment was carried out using the model called CrossPitch_Ident.mdl. This model excites the TRMS with U_1 and records its response φ – yaw angle. The excitation signal is composed of several sinusoids. Two signals are collected in the form of vectors and are available in Workspace.

Task:

The identification experiment was conducted and data was collected. The model was identified using the Matlab identification interface. The graphs generated of the transient response, step response analysis, frequency response, pole and zeros map, and model residuals give a clear idea of the quality of the response.

Results:

The following results were obtained in the experiment conducted.

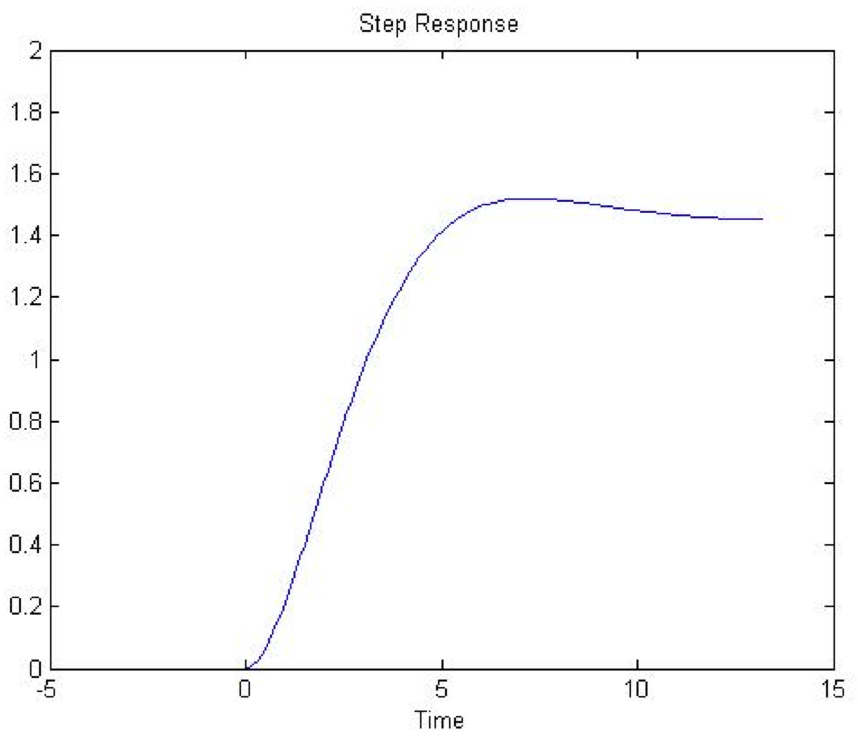

Fig. 4.11 *Step Response*

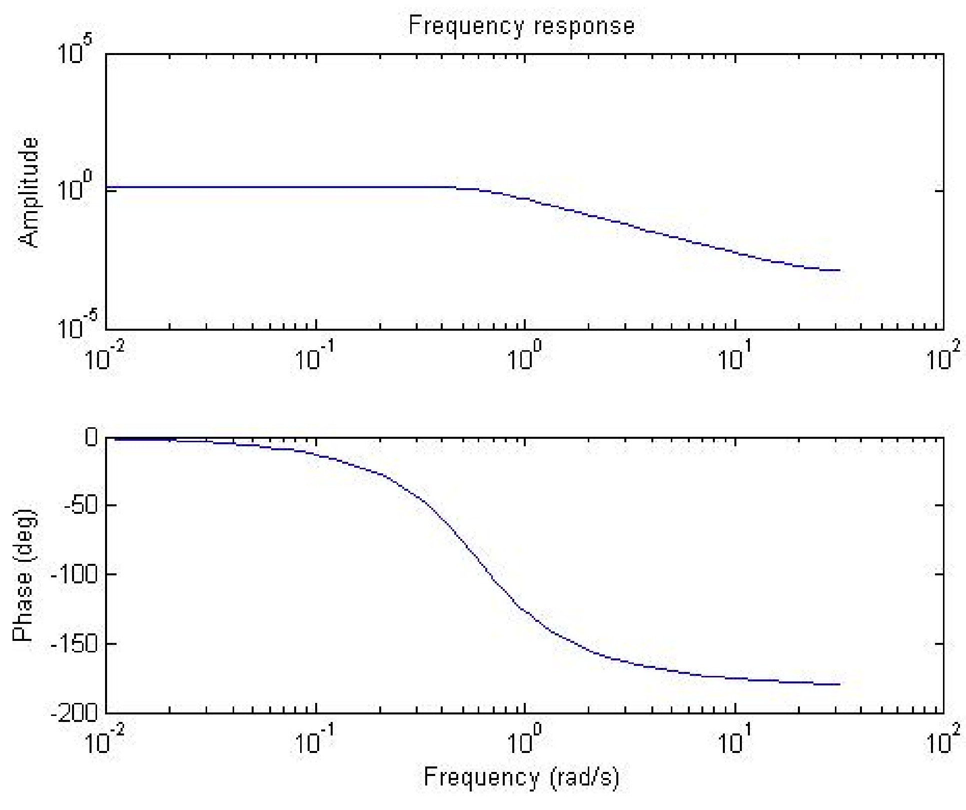

Fig. 4.12 *Frequency Response*

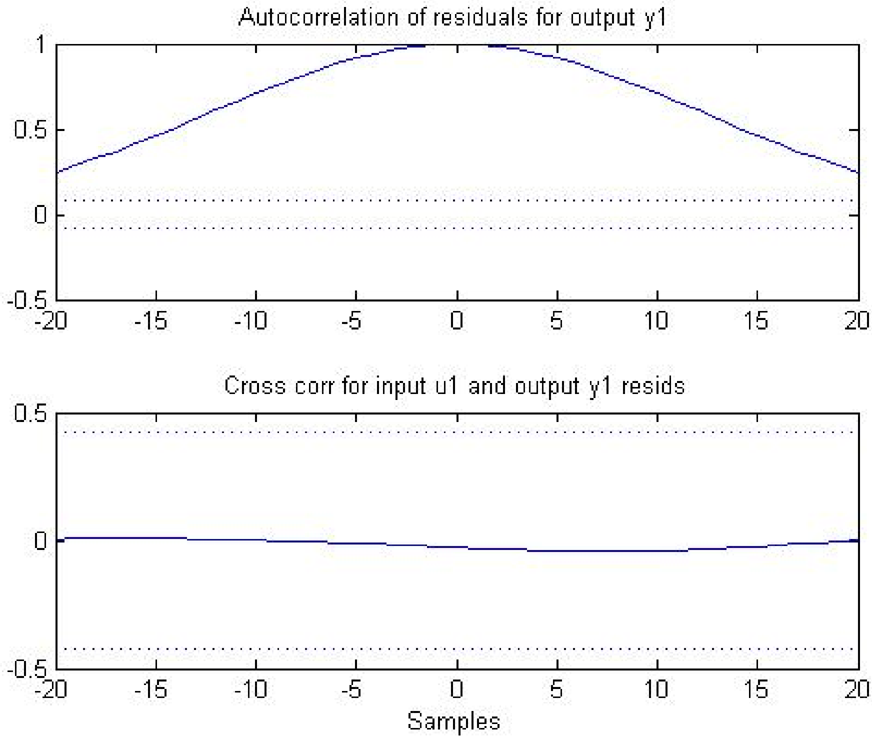

Fig. 4.13 *Autocorrelation of Residuals for Output y_1*

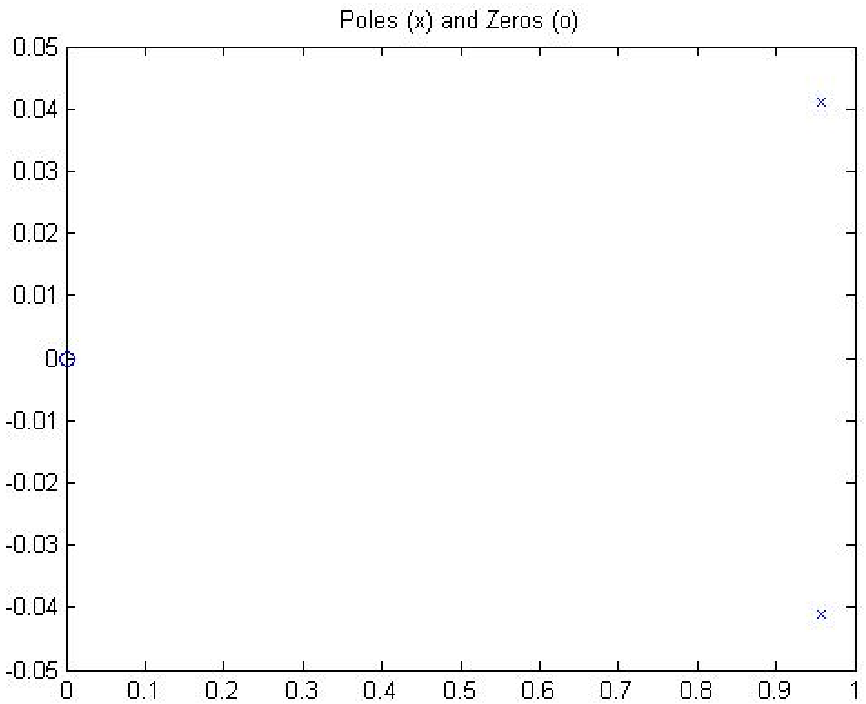

Fig. 4.14 *Poles & Zeroes*

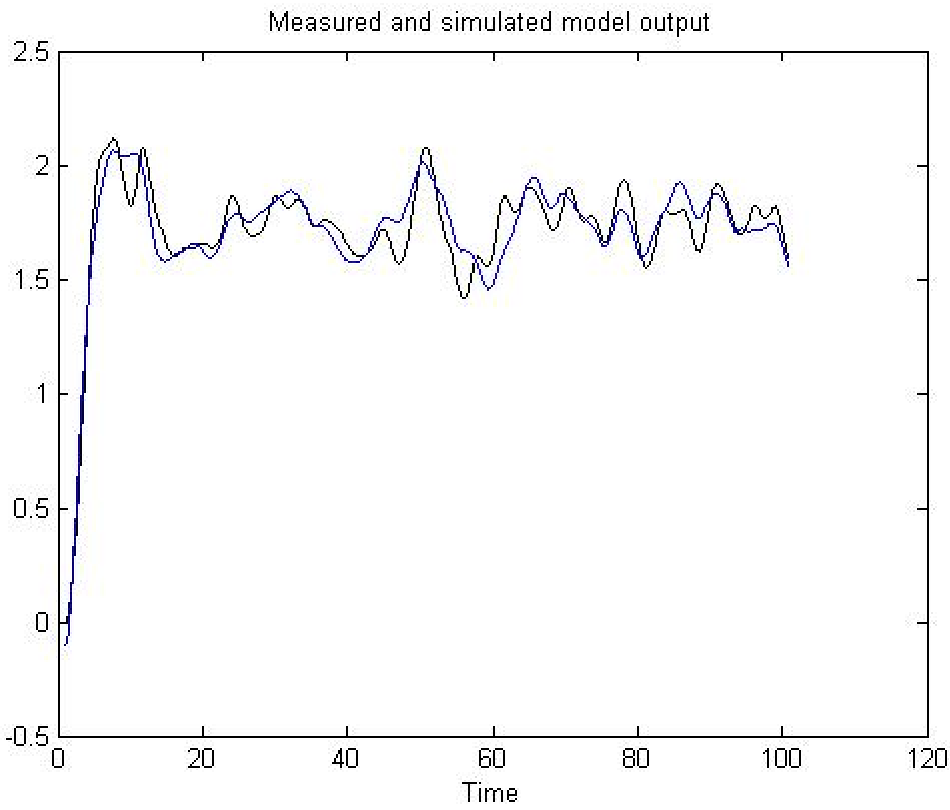

Fig. 4.15 *Measured & Simulated Model Output*

Discrete Transfer function thus obtained:

$$\frac{0.005052}{z^2 - 1.916\,z + 0.9191}$$

Continuous Transfer function thus obtained:

$$\frac{0.02599\,s + 0.527}{s^2 + 0.8434\,s + 0.3627}$$

Sampling time: 0.1 [s].

Discussions:

Fig 4.11 shows the Step Response of the Main Path Yaw Rotor Identification. It basically gives knowledge of how the system behaves in time when the inputs change from zero to one in a relatively short span of time. In this case the system is stable because it settles down to give a steady output by reaching another steady state in a short span of time. Fig 4.12 shows the Frequency Response of the system, which is the measure of a system's output spectrum with respect to its input signal. In this case a Bode Plot has been drawn to plot the magnitude (measured in dB) and the phase (measured in radians) versus frequency. Fig 4.13 shows the Autocorrelation of the output, which means that it, is the cross-correlation of the output signal with itself observed as a function of a time lag with itself. Fig 4.14 is the Poles and Zeroes map which shows the position and number of poles and zeroes of the transfer function. If any of the position of these poles or zeroes were to be changed, then it would have great implications on the Step Response of the system. In this case there are no zeroes. Fig 4.15 is the comparison between the measured and simulated output which is a comparison between the control and real time experiments. While the control experiments have been done in perfect setup, external factors come into picture in case of the real time experiment. In this case, the graphs vary at very few points; hence the real time experimental results have some or little error.

4.4 CROSS PATH YAW ROTOR IDENTIFICATION

Introduction:

The identification experiment was carried out using the model called CrossYaw_Ident.mdl. This model excites the TRMS with U_2 and records its response ψ – pitch angle. The excitation signal is composed of several sinusoids. Two signals are collected in the form of vectors and are available in Workspace.

Task:

The identification experiment was conducted and data was collected. The model was identified using the Matlab identification interface. The graphs generated of the transient response, step response analysis, frequency response, pole and zeros map, and model residuals give a clear idea of the quality of the response.

Results:

The following results were obtained in the experiment conducted.

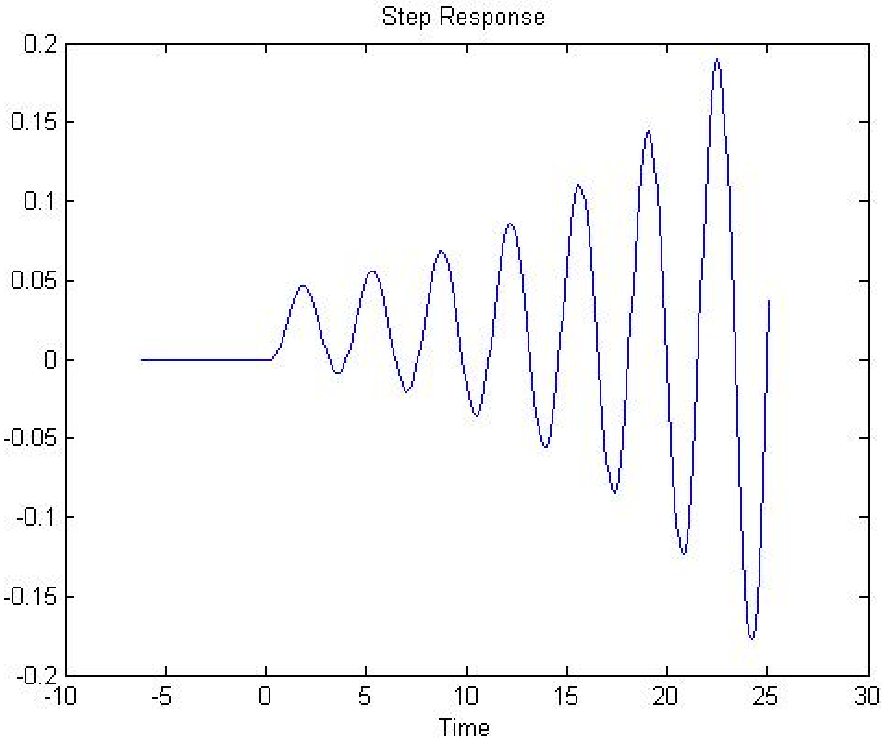

Fig. 4.16 *Step Response*

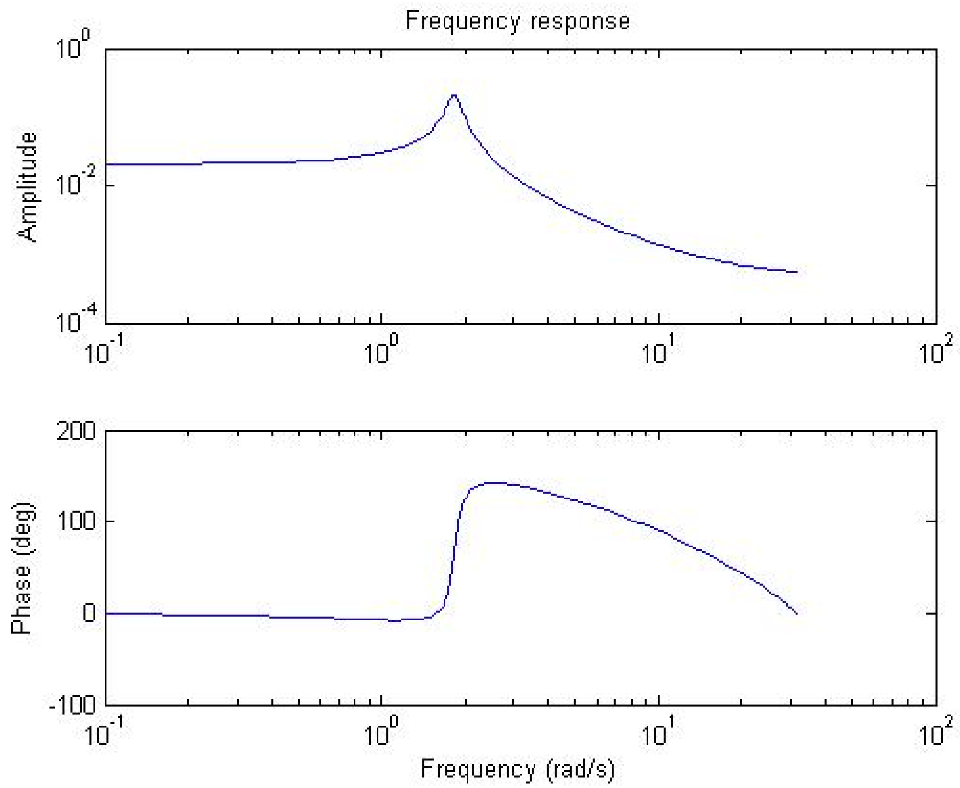

Fig. 4.17 *Frequency Response*

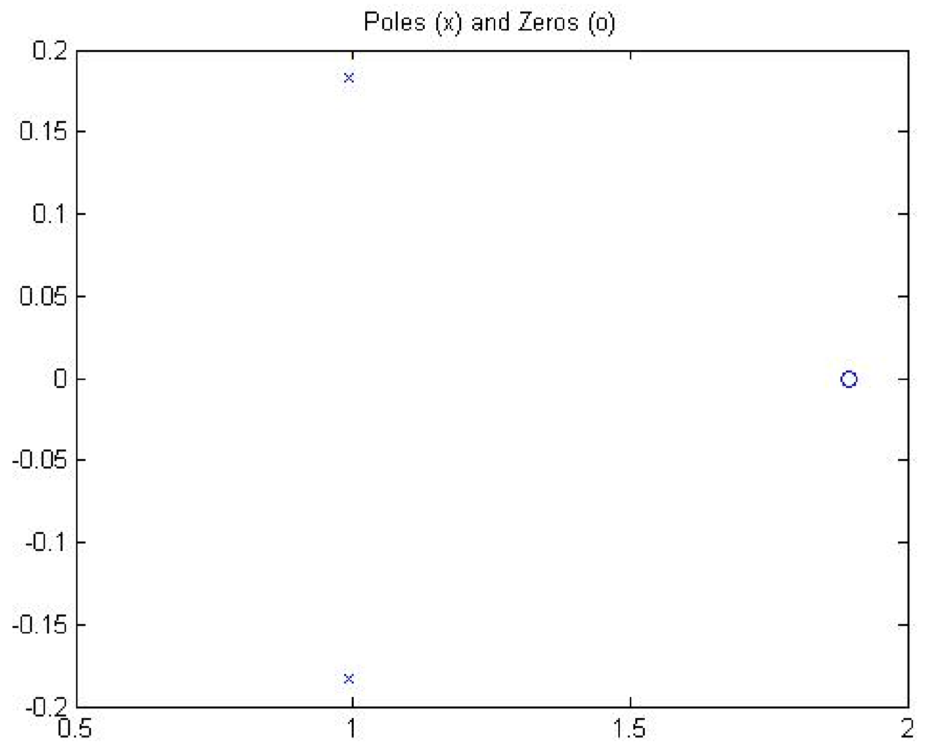

Fig. 4.18 *Poles & Zeroes*

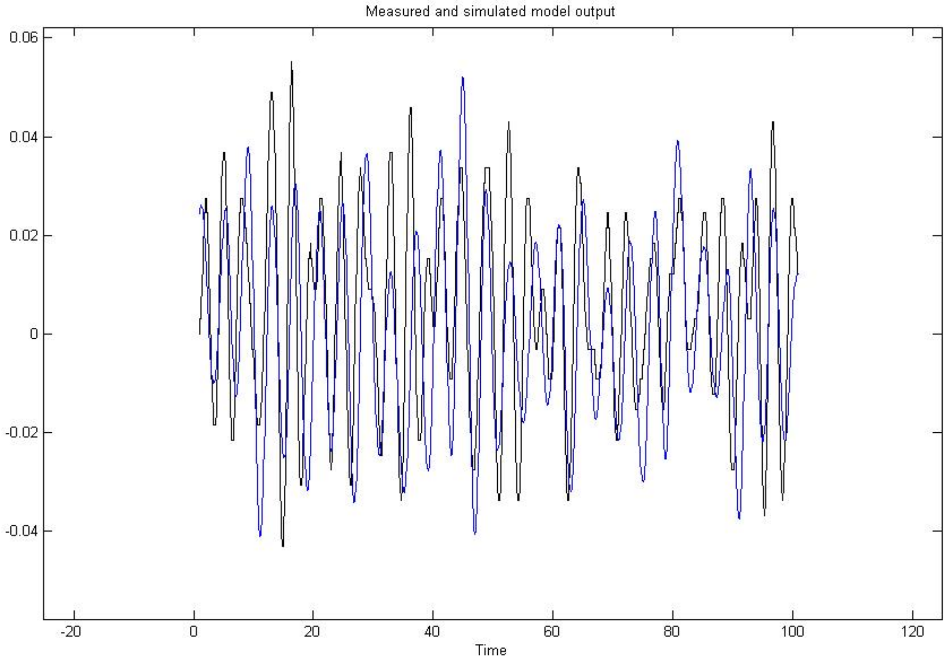

Fig. 4.19 *Measured & Simulated Model Output*

Discrete Transfer function thus obtained:

$$\frac{-0.0007785\,z + 0.001475}{z^2 - 1.985\,z + 1.019}$$

Continuous Transfer function thus obtained:

$$\frac{-0.01122\,s + 0.06921}{s^2 - 0.1844\,s + 3.348}$$

Sampling time: 0.1 [s].

Discussions:

Fig 4.16 shows the Step Response of the Main Path Yaw Rotor Identification. It basically gives knowledge of how the system behaves in time when the inputs change from zero to one in a relatively short span of time. In this case the system is somewhat unstable because of the high non linearity and cross-couplings of the axes. Fig 4.17 shows the Frequency Response of the system, which is the measure of a system's output spectrum with respect to its input signal. In this case a Bode Plot has been drawn to plot the magnitude (measured in dB) and the phase (measured in radians) versus frequency. Fig 4.18 shows the Autocorrelation of the output, which means that it, is the cross-correlation of the output signal with itself observed as a function of a time lag with itself. Fig 4.19 is the Poles and Zeroes map which shows the position and number of poles and zeroes of the transfer function. If any of the position of these poles or zeroes were to be changed, then it would have great implications on the Step Response of the system. In this case there are no zeroes. Fig 4.20 is the comparison between the measured and simulated output which is a comparison between the control and real time experiments. While the control experiments have been done in perfect setup, external factors come into picture in case of the real time experiment. In this case, the graphs vary at some points; hence the real time experimental results have error.

CHAPTER 5

IDENTIFICATION USING NEURAL NETWORK MODELS

5.1 INTODUCTION TO SYSTEM IDENTIFICATION USING NEURAL NETWORK MODELS

Neural Networks (NNs) comprise of networks of neurons, for e.g. as in human brains. Artificial Neurons are physical devices or mathematical constructs, which are often crude approximations of the neurons found in a brain. Artificial Neural Networks (ANNs) are networks of Artificial Neurons; physical devices or simulated on computers and behave as approximations to the parts of a real brain. Practically an ANN is a parallel computational system comprising of simple processing elements inter-connected in a particular way to perform a specific task.

These powerful computational devices are extremely efficient due to massive parallelism. As they have the capability to learn and generalize from training data, therefore there is no need for tedious programming and lengthy calculations. Extremely fault tolerant and noise tolerant, they possess the ability to cope with situations which normal symbolic systems have difficulty to deal with.

Taking assumptions, a discrete-time multivariable non-linear control system with m outputs and r inputs can be represented by the multi variable NARMAX model:

$$y(t) = f\{y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u), e(t-1), \ldots, e(t-n_e)\} + e(t)$$

where y(t), u(t) and e(t) are the system output, input and noise vectors, respectively; n_y, n_u and n_e are the maximum lags in the output, input and noise respectively; *e(t)* is a zero-mean independent sequence; and $f(\bullet)$ is some vector-valued non-linear function. [3]

The input-output relationship is dependent upon the non-linear function f(•). In reality, f(•) is generally very complex and knowledge of the form of this function is often not available. The solution is to approximate f(•) using some known simpler function, and in the present study we consider using neural networks to approximate non-linear systems governed by the model

$$y(t) = f(y(t - 1), \ldots , y(t - n_y) , u(t - 1), \ldots , u(t - n_u) + e(t)$$

It can be noticed that the later equation is a slightly simplified version of the former one because only additive uncorrelated noise is considered. [3]

Neural networks used for function approximation purposes are feed forward type networks, typically with one or more hidden layers between the inputs and outputs. Every layer comprises of some computing units known as nodes. The network inputs are passed on to each node in the first layer. Then the outputs of the first layer nodes are passed to the second layer, and so on. Hence the network outputs the outputs of the nodes lying in the final layer. Generally all the nodes in a layer are completely connected to the nodes in adjacent layers, but there is no inter-connection within a layer. The I/O relationship of each hidden node is determined by the connection weights W_i, a threshold parameter Il and the node activation function a(•), as follows: [3]

$$y = a(\Sigma\, W_i X_i + \mu)$$

The objective of this process is to carry out system identification of the TRMS by using neural networks. The input (u) of interest is the control voltage applied to the TRMS while the output (y) is the yaw or pitch angle as the case maybe. In the identification framework, we assume that the TRMS can be represented in discrete input-output form by the identification structure:

$$y(k)= a_1(k-1)+a_2y(k-2)\ldots+a_ny(k-n)+a_1u(k-1)+a_2u(k-2)\ldots+a_nu(k-n)+\epsilon(k)$$

In our experiment, firstly a set of data was collected experimentally. Then the output of the system was corrupted by Gaussian white noise with SNR (Signal-to-Noise Ratio) 25dB. Then system was trained using 500 input-output data pairs and result was obtained.

5.2 MAIN PATH PITCH ROTOR IDENTIFICATION

Program Code:

```
% Adding Gaussian white noise to system output
yn=awgn(y,25);
% y - system output
% yn - system output corrupted by noise
figure(1)
plot(1:N,y,1:N,yn,'r');
for k=1:(N-2)
    input(1,k)=yn(k+1);
    input(2,k)=yn(k);
    input(3,k)=u(k+1);
    input(4,k)=u(k);
    target(k)=yn(k+2);
end
% Creating a feedforward NN
net=newff(minmax(input),[1],{'purelin'});
net.trainparam.goal=1e-3;
% Train the NN
net=train(net,input,target);
% Trained NN's weights and biases
net.IW{1}
net.b{1}
% Output of trained NN for training input
```

```
ytraincap=sim(net,input);
ytraincap=[0 0 ytraincap];
figure(2)
plot(1:N,ytraincap,'r',1:N,y);
% Testing-input generation (step)
u1=ones(1,N);
y1(1)=0;y1(2)=0;y1(3)=0;y1(4)=0;
for k=5:N
    y1(k)=0.8305*y1(k-4)-1.8*y1(k-3)+0.1683*y1(k-
2)+1.788*y1(k-1)+0.008894*u1(k-4)-0.003293*u1(k-3);
end
for k=1:(N-2)
    input1(1,k)=y1(k+1);
    input1(2,k)=y1(k);
    input1(3,k)=u1(k+1);
    input1(4,k)=u1(k);
    target1(k)=y1(k+2);
end
% Testing NN
ycap=sim(net,input1);
ycap=[0 0 ycap];
% y1 - system output
% ycap - NN output
% u1 - test input
figure(3)
plot(1:N,u1,1:N,ycap,'r',1:N,y1,'g');
```

Results:

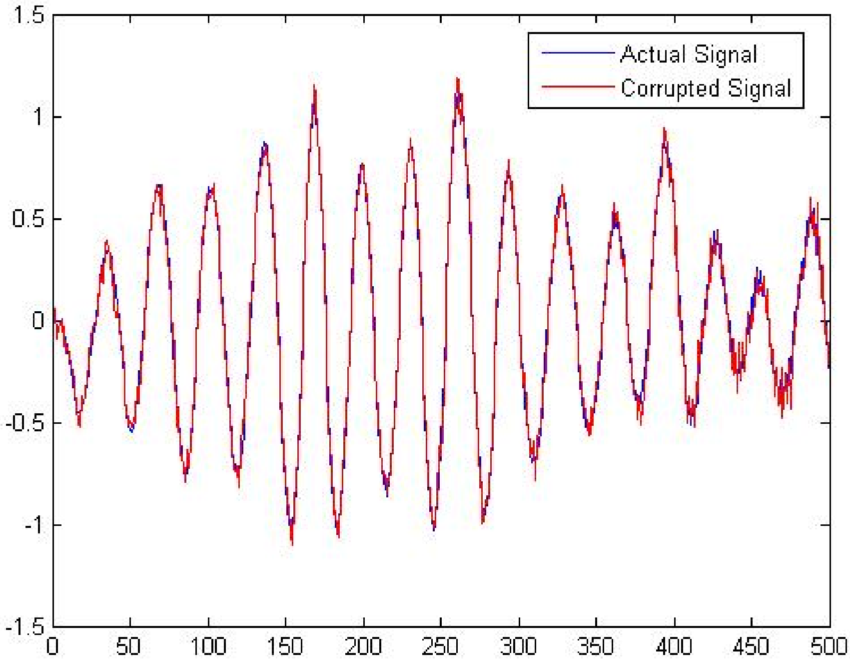

Fig. 5.1 *Actual Vs Corrupted Signal*

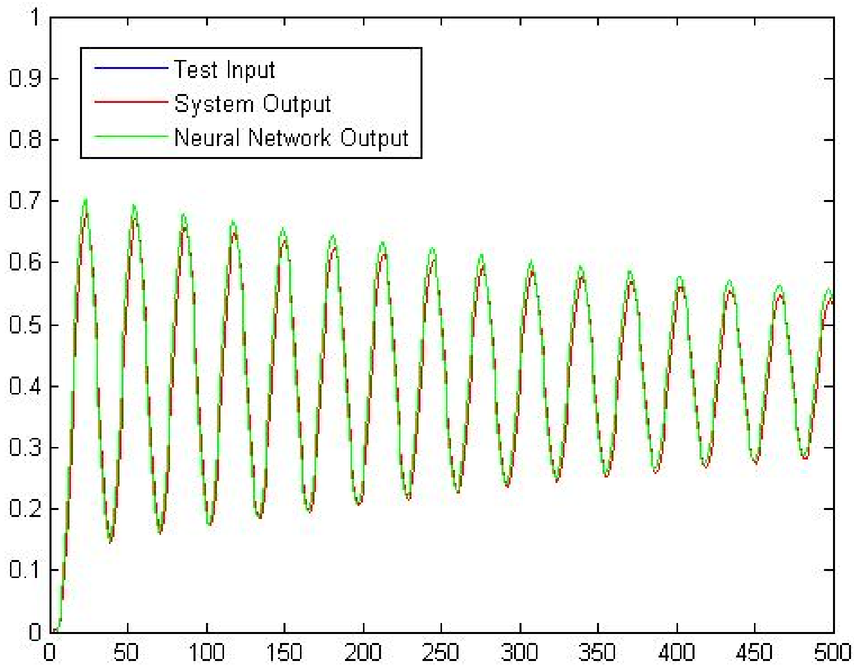

Fig. 5.2 *System Output Vs NN Output*

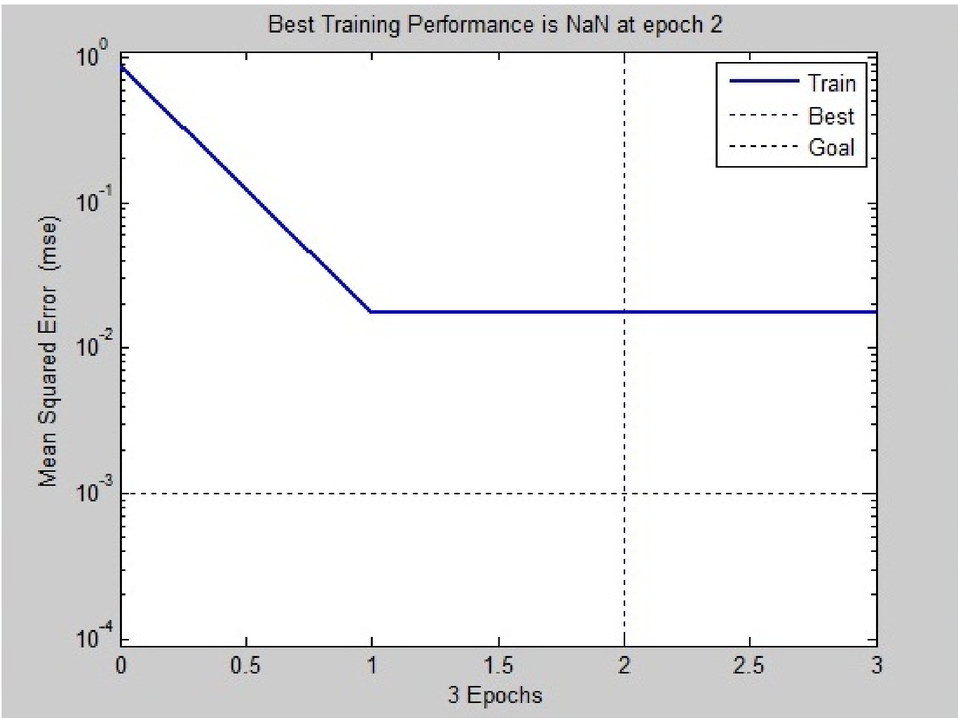

Fig. 5.3 *Best Training Performance*

<u>Discussion:</u>

Fig 5.1 shows the comparison between the actual signal and the corrupted signal. The corrupted signal has Gaussian white Noise with SNR 25dB. Fig 5.2 shows the comparison between the system output and the Neural Network Output. Fig 5.3 shows the Best Training Performance plot.

5.3 MAIN PATH YAW ROTOR IDENTIFICATION

Program Code:

```
% Adding Gaussian white noise to system output
yn=awgn(y,1);
% y - system output
% yn - system output corrupted by noise
figure(1)
plot(1:N,y,1:N,yn,'r');
for k=1:(N-2)
    input(1,k)=yn(k+1);
    input(2,k)=yn(k);
    input(3,k)=u(k+1);
    input(4,k)=u(k);
    target(k)=yn(k+2);
end
% Creating a feedforward NN
net=newff(minmax(input),[1],{'purelin'});
net.trainparam.goal=1e-3;
% Train the NN
net=train(net,input,target);
% Trained NN's weights and biases
net.IW{1}
net.b{1}
% Output of trained NN for training input
```

```
ytraincap=sim(net,input);

ytraincap=[0 0 ytraincap];

figure(2)

plot(1:N,ytraincap,'r',1:N,y);

% Testing-input generation (step)

u1=ones(1,N);

y1(1)=0;y1(2)=0;y1(3)=0;

for k=4:N

    y1(k)=0.89*y1(k-3)-2.771*y1(k-2)+2.88*y1(k-

1)+0.0003014*u1(k-2)+0.0248*u1(k-3);

end

for k=1:(N-2)

    input1(1,k)=y1(k+1);

    input1(2,k)=y1(k);

    input1(3,k)=u1(k+1);

    input1(4,k)=u1(k);

    target1(k)=y1(k+2);

end

% Testing NN

ycap=sim(net,input1);

ycap=[0 0 ycap];

% y1 - system output

% ycap - NN output

% u1 - test input

figure(3)

plot(1:N,u1,1:N,ycap,'r',1:N,y1,'g');
```

Results:

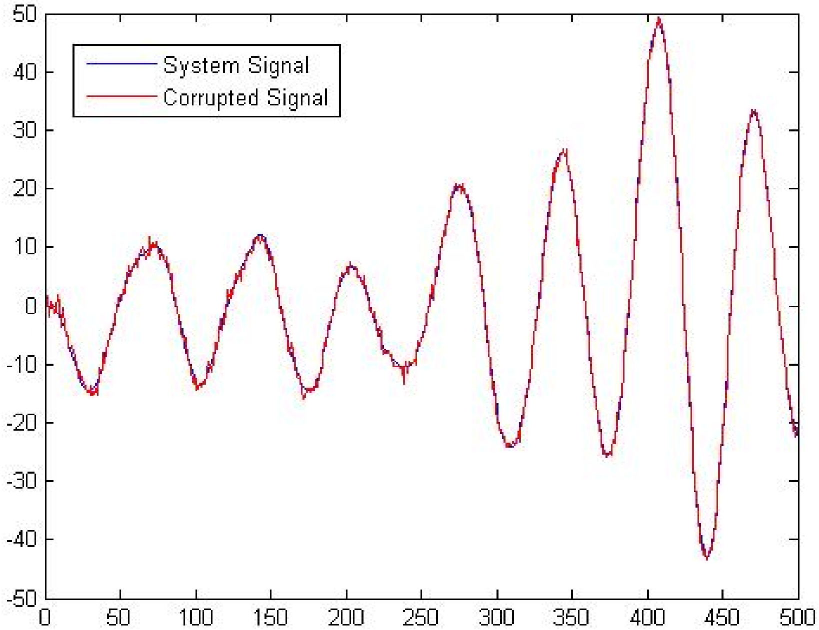

Fig. 5.4 *Actual Vs Corrupted Signal*

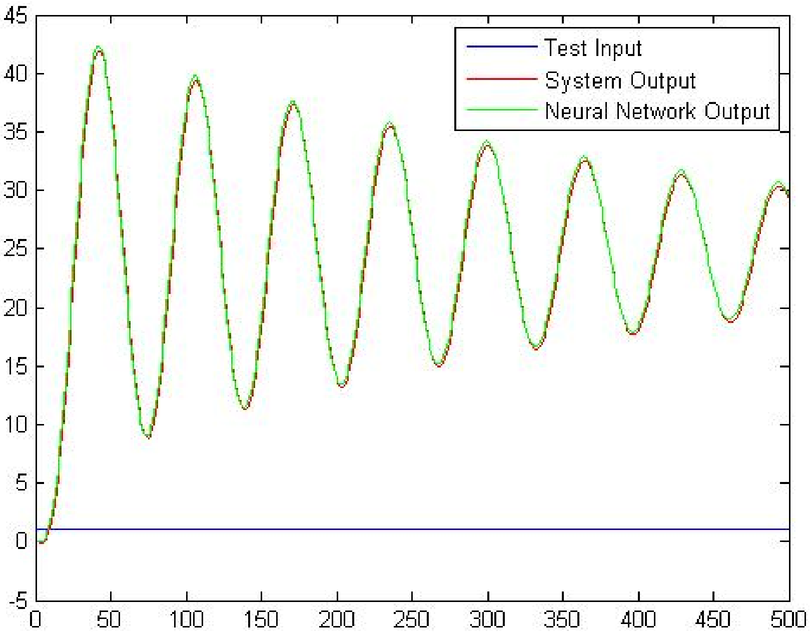

Fig. 5.5 *System Output Vs NN Output*

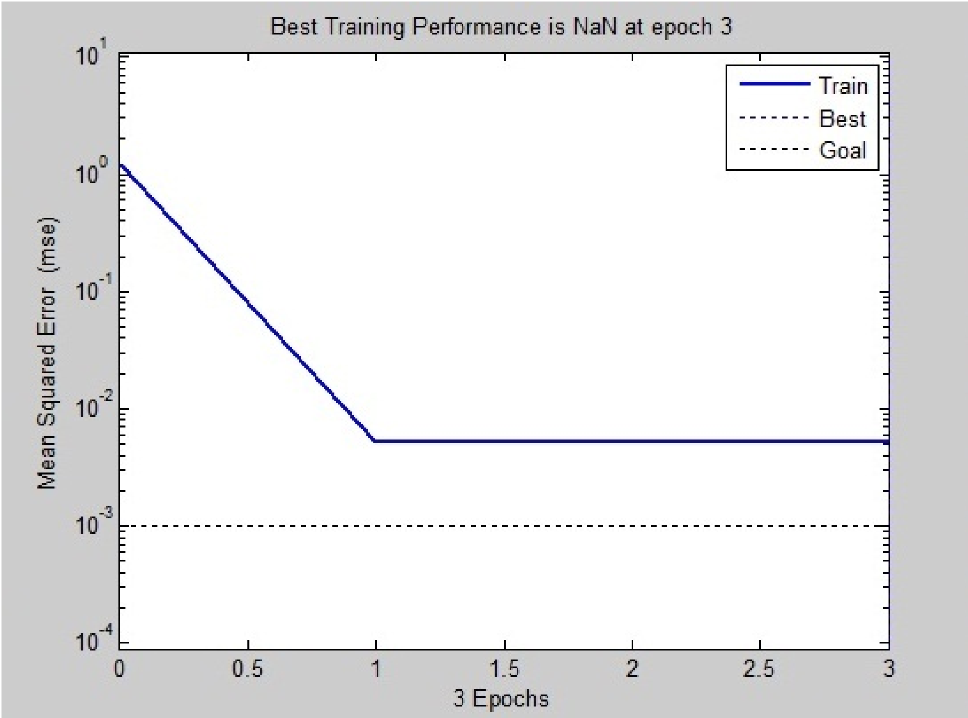

Fig. 5.6 *Best Training Performance*

Discussion:

Fig 5.4 shows the comparison between the actual signal and the corrupted signal. The corrupted signal has Gaussian white Noise with SNR 25dB. Fig 5.5 shows the comparison between the system output and the Neural Network Output. Fig 5.6 shows the Best Training Performance plot.

5.4 CROSS PATH PITCH ROTOR IDENTIFICATION

Program Code:

```
% Adding Gaussian white noise to system output
yn=awgn(y,25);
% y - system output
% yn - system output corrupted by noise
figure(1)
plot(1:N,y,1:N,yn,'r');
for k=1:(N-2)
    input(1,k)=yn(k+1);
    input(2,k)=yn(k);
    input(3,k)=u(k+1);
    input(4,k)=u(k);
    target(k)=yn(k+2);
end
% Creating a feedforward NN
net=newff(minmax(input),[1],{'purelin'});
net.trainparam.goal=1e-3;
% Train the NN
net=train(net,input,target);
% Trained NN's weights and biases
net.IW{1}
net.b{1}
% Output of trained NN for training input
```

```
ytraincap=sim(net,input);
ytraincap=[0 0 ytraincap];
figure(2)
plot(1:N,ytraincap,'r',1:N,y);
% Testing-input generation (step)
u1=ones(1,N);
y1(1)=0;y1(2)=0;
for k=3:N
    y1(k)=-0.9191*y1(k-2)+1.916*y1(k-1)+0.005052*u1(k-2);
end
for k=1:(N-2)
    input1(1,k)=y1(k+1);
    input1(2,k)=y1(k);
    input1(3,k)=u1(k+1);
    input1(4,k)=u1(k);
    target1(k)=y1(k+2);
end
% Testing NN
ycap=sim(net,input1);
ycap=[0 0 ycap];
% y1 - system output
% ycap - NN output
% u1 - test input
figure(3)
plot(1:N,u1,1:N,ycap,'r',1:N,y1,'g');
```

Results:

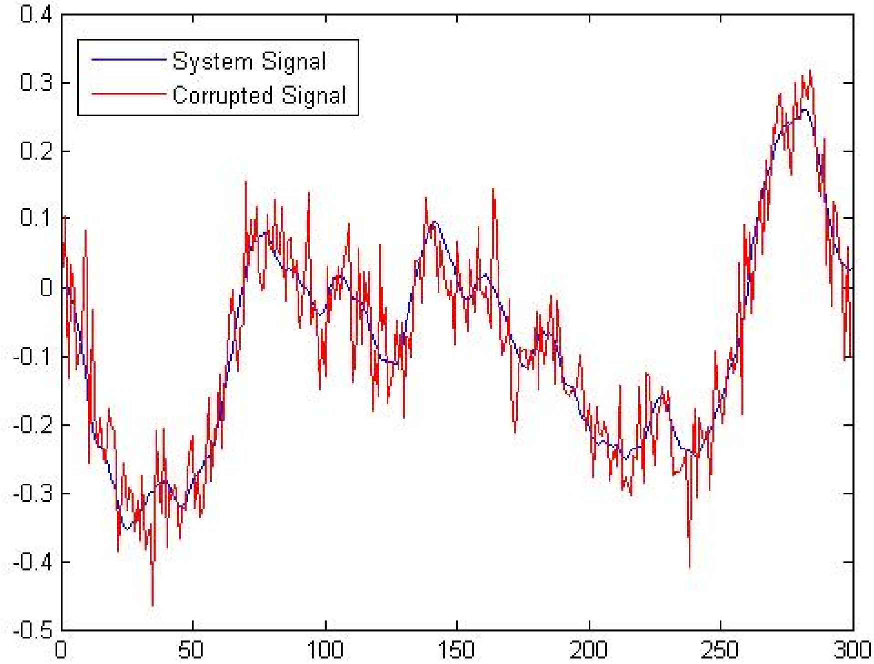

Fig. 5.7 *Actual Vs Corrupted Signal*

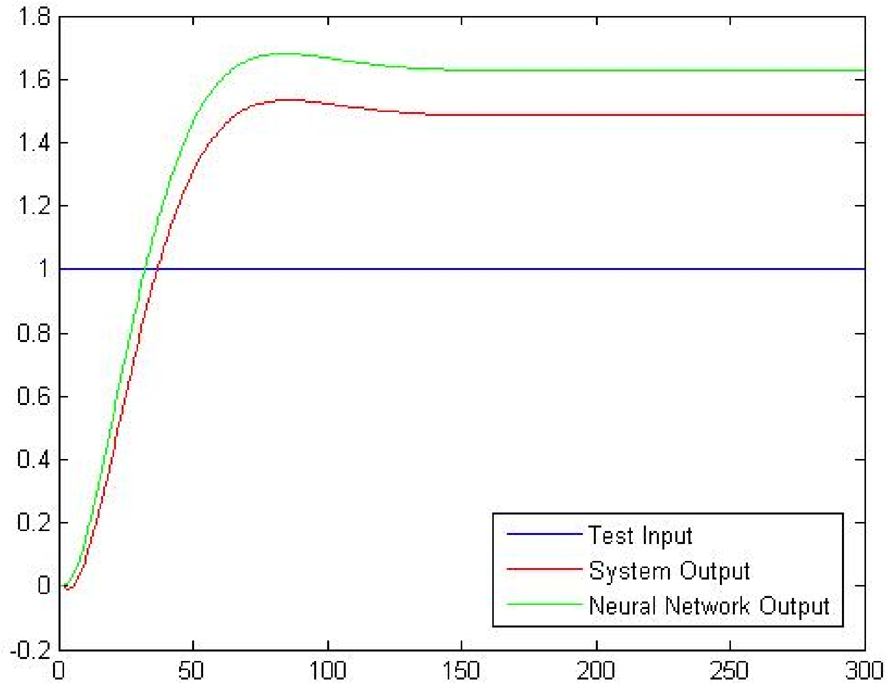

Fig. 5.8 *System Output Vs NN Output*

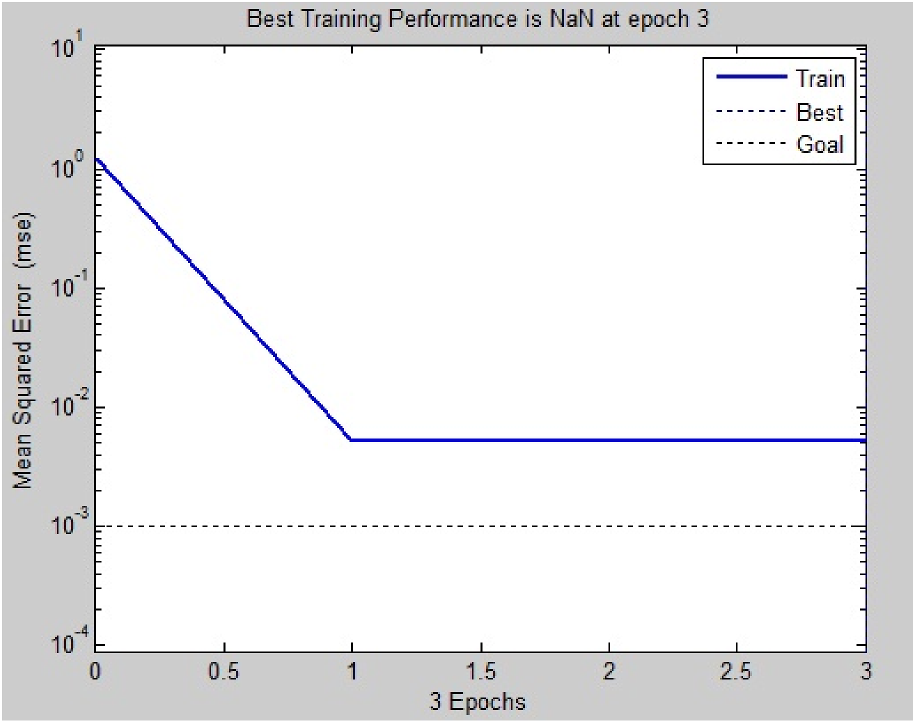

Fig. 5.9 *Best Training Performance*

<u>Discussion:</u>

Fig 5.7 shows the comparison between the actual signal and the corrupted signal. The corrupted signal has Gaussian white Noise with SNR 25dB. Fig 5.8 shows the comparison between the system output and the Neural Network Output. Fig 5.9 shows the Best Training Performance plot.

5.5 CROSS PATH YAW ROTOR IDENTIFICATION

Program Code:

```
% Adding Gaussian white noise to system output
yn=awgn(y,25);
% y - system output
% yn - system output corrupted by noise
figure(1)
plot(1:N,y,1:N,yn,'r');
for k=1:(N-2)
    input(1,k)=yn(k+1);
    input(2,k)=yn(k);
    input(3,k)=u(k+1);
    input(4,k)=u(k);
    target(k)=yn(k+2);
end
% Creating a feedforward NN
net=newff(minmax(input),[1],{'purelin'});
net.trainparam.goal=1e-3;
% Train the NN
net=train(net,input,target);
% Trained NN's weights and biases
net.IW{1}
net.b{1}
% Output of trained NN for training input
```

```
ytraincap=sim(net,input);
ytraincap=[0 0 ytraincap];
figure(2)
plot(1:N,ytraincap,'r',1:N,y);
% Testing-input generation (step)
u1=ones(1,N);
y1(1)=0;y1(2)=0;
for k=3:N
    y1(k)=-1.019*y1(k-2)+1.985*y1(k-1)+0.001475*u1(k-2)-
0.0007785*u1(k-1);
end
for k=1:(N-2)
    input1(1,k)=y1(k+1);
    input1(2,k)=y1(k);
    input1(3,k)=u1(k+1);
    input1(4,k)=u1(k);
    target1(k)=y1(k+2);
end
% Testing NN
ycap=sim(net,input1);
ycap=[0 0 ycap];
% y1 - system output
% ycap - NN output
% u1 - test input
figure(3)
plot(1:N,u1,1:N,ycap,'r',1:N,y1,'g');
```

Results:

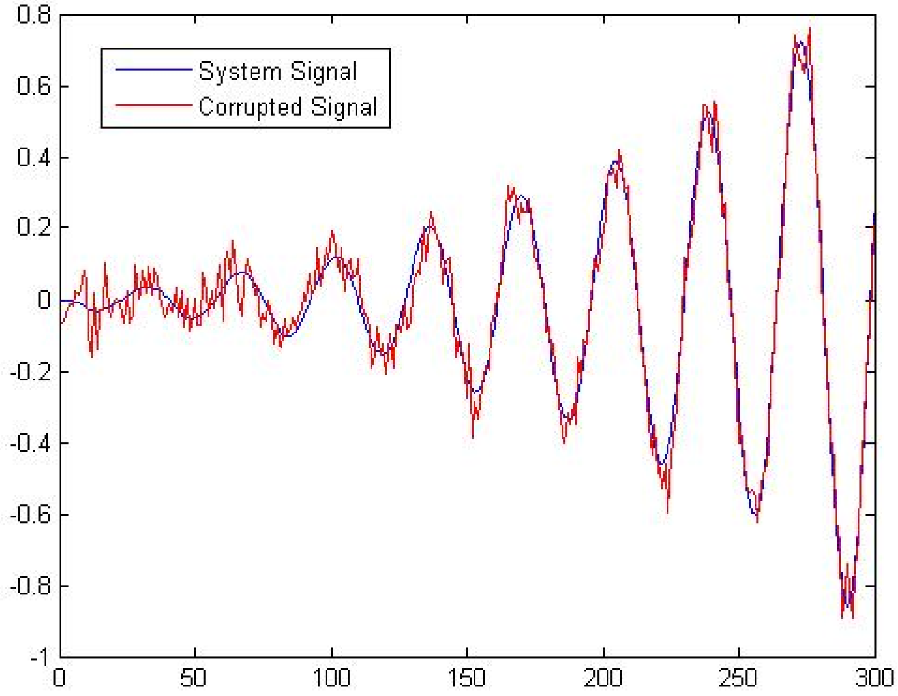

Fig. 5.10 *Actual Vs Corrupted Signal*

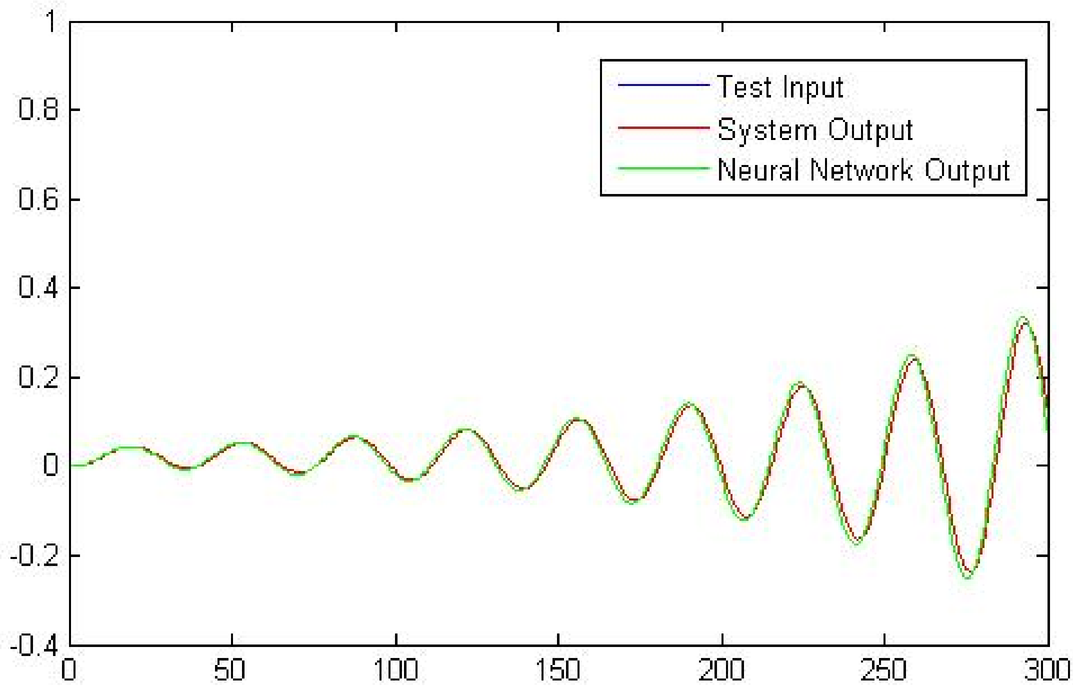

Fig. 5.11 *System Output Vs NN Output*

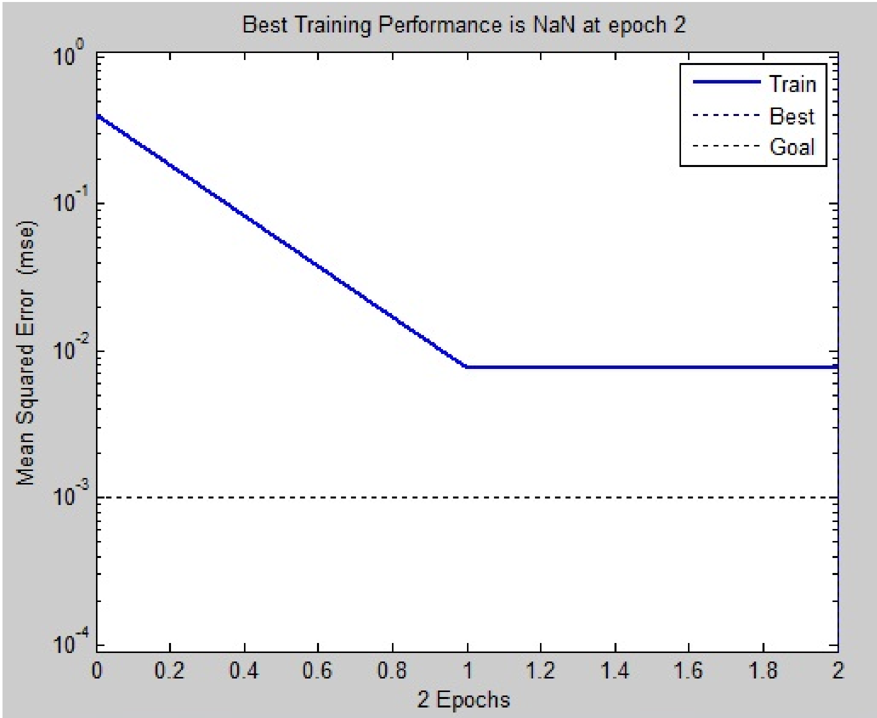

Fig. 5.12 *Best Training Performance*

Discussion:

Fig 5.10 shows the comparison between the actual signal and the corrupted signal. The corrupted signal has Gaussian white Noise with SNR 25dB. Fig 5.11 shows the comparison between the system output and the Neural Network Output. Fig 5.12 shows the Best Training Performance plot.

CONCLUSION

Modelling of physical systems are essential in the design of a controller for its analysis and future applications. In this investigation the system identification of an experimental system, Twin Rotor MIMO System, Feedback Instruments Ltd, using both analytical and neural network based methods has been developed. While development of the analytical model of the TRMS, various components of the system have been modelled individually and then combined. The various responses of the system models have been compared with that of the real time setup.

Neural network's ability to model complex non-linear MIMO system has been demonstrated. We know that neural networks provide an excellent platform to approximate any complex non-linear system with reasonable accuracy. In the project we therefore demonstrate the generation of input-output data pair using the laboratory model and use them for modelling using neural networks.

In our neural network modelling the Levenberg–Marquardt algorithm (LMA) is used which provides a numerical solution to the problem of minimizing the approximation function on this nonlinear system model, over a space of parameters of the function. These minimization problems arise especially in least squares curve fitting and nonlinear programming. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. On the other hand, for well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as GNA improved with trust region approach.

We have compared both the methods of identification i.e. the analytical and neural network. Although both the methods have given us quite accurate results, the neural network approach provides for a better identification as it is extremely fault tolerant and noise tolerant and possess the ability to cope with situations which normal symbolic systems have difficulty to deal with.

REFERENCES

1. *Rahideh, M.H. Shaheed and H.J.C. Huijberts. "Dynamic Modelling of the TRMS using Analytical and Empirical approaches", Control Engineering Practice, Volume 16, Issue 3, March 2008, Pages 241-259.*
2. *A.Rahideh, H.M. Shaheed and A. H. Bajodah."Adaptive non-linear model inversion control of a twin rotor multi-input multi-output system using artificial intelligence", Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, Volume 221, Number 3 / 2007, Pages 343-351.*
3. *S. Chen, S.A. Billings and P.M. Grant, Non-linear system identification using neural networks, INT. J. CONTROL, 1990, VOL. 51, No.6, Pages 1191-1214.*
4. *Feedback Instruments Ltd, Twin Rotor MIMO System Control Experiments, 33-949S, Laboratory Manual.*
5. *M. Gopal, Digital Control and State Variable Methods: Conventional and Intelligent Control Systems, Tata McGraw-Hill Publishing Company Limited.*