
Chapter 8

Implementation

8. Implementation

Mathematical analysis and the ground work for the implementation of embedding Real time control to the Cargo ship and fault tolerant aircraft is already been discussed in the previous chapter. In this chapter, the actual implementation of embedding evolutionary algorithms, specific to these applications with assumptions and constraints, if any, is carried out and results of the same are discussed.

MATLAB programming is used for implementation of these applications.

8.1 Cargo Ship Steering Application

Based on the cargo ship dynamics defined in the section 7.2, Fuzzy Model Reference Learning Controller as well as Evolutionary algorithm based controller as implemented and results of the same are compared with that of different intelligent controllers like PD controller, ANN controller with MLP as well as RBF, Fuzzy controller, are compared.

FMRLC Design

The Design for controlling of directional heading of the cargo ship using an FMRLC is discussed. The inputs to the fuzzy controller, shown in Figure 7.1, are the heading error and change in heading error expressed as

$$e(kT) = \psi_r(kT) - \psi(kT) \quad \dots(8.1)$$

and

$$c(kT) = \frac{e(kT) - e(kT - T)}{T} \quad \dots(8.2)$$

Where $\psi_r(kT)$ is the desired ship heading. The controller output is the rudder angle $\delta(kT)$ of the ship. For the fuzzy controller design, eleven uniformly spaced triangular membership functions are defined for each controller input, as shown in Figure 7.2.

The scaling controller gains for the error, change in error, and the controller output are chosen via the design procedure to be

$$g_e = 1/\pi, \text{ the error } e(kT) \text{ can never be over } 180 \text{ degree.}$$

$g_c = 100$, assuming that ship do not move much faster than 0.01 rad/sec. or 0.57 deg/sec and

$$g_u = 8\pi/18, \text{ as rudder angle } \delta \text{ is limited between } \pm 80 \text{ degree, } g_u = 80\pi/180 = 8\pi/18.$$

The fuzzy sets for the fuzzy controller output are assumed to be symmetric and triangular-shaped with a base width of 0.4, and all centered at zero on the normalized universe of discourse. A first order reference models are implemented.

The input to the fuzzy inverse model includes the error and change in error between the reference model and the ship heading expressed as

$$\psi_e(kT) = \psi_m(kT) - \psi(kT) \quad \dots(8.3)$$

and

$$\psi_c(kT) = \frac{\psi_e(kT) - \psi_e(kT - T)}{T} \quad \dots(8.4)$$

For selecting & tuning of the scaling gains of the inverse model the procedure followed is

1. Select the gain g_{ye} so that $y_e(kT)$ will not saturate the input membership function certainty, near the endpoints. This is a heuristic choice since we cannot know a priori how big $y_e(kT)$ will get; however, intuition about the process can be quite useful in determining the maximum value. Based on this $g_{ye} = 1/\pi$, the error $y_e(kT)$ can never be over 180 degree.
2. Choose the gain g_p to be the same as for the fuzzy controller output gain g_u . i.e. $g_p = 8\pi/18$. Let $g_{yc} = 0$.
3. Apply a step reference input $r(kT)$ that is of a magnitude that may be typical during normal operation.

4. Observe the plant and reference model responses. There are three cases:
 - a. If there are unacceptable oscillations in the plant output response about the reference model response, then increase g_{yc} , as it requires additional derivative action in the learning mechanism to reduce the oscillations. Go to step 3.
 - b. If the plant output is unable to “keep up” with the reference model response, then decrease g_{yc} . Go to step 3.
 - c. If the plant response is acceptable with respect to the reference model response, then the controller design is completed.

After following the above procedures the value of g_{yc} obtained is 5.

For a cargo ship, an increase in the rudder angle $\delta(kT)$ will generally result in a *decrease* in the ship heading angle (refer Figure 7.2). This is the information about the inverse dynamics of the plant that is used in the fuzzy inverse model rules.

As discussed in the section 7.1.3, we are using the form

$$\text{If } \tilde{\psi}_e \text{ is } \tilde{\Psi}_e^i \text{ and } \tilde{\psi}_c \text{ is } \tilde{\Psi}_c^j \text{ then } \tilde{p} \text{ is } \tilde{P}^m$$

If we assume that the center of the output membership function for this rule c_{ij} to emphasize that it is the center associated with the output membership function that has the i^{th} membership function for the $\tilde{\psi}_e$ universe of discourse and the j^{th} membership function for the $\tilde{\psi}_c$ universe of discourse. The rule-base array shown in Table 8.1 is employed for the fuzzy inverse model for the cargo ship. In Table 8.1, $\tilde{\Psi}_e^i$ denotes the i^{th} fuzzy set associated with the error signal ψ_e , and $\tilde{\Psi}_c^j$ denotes the j^{th} fuzzy set associated with the change in error signal ψ_c . Each element in the table represents the center value of symmetric triangular-shaped membership functions c_{ij} with base widths 0.4 for output fuzzy sets P_m on the normalized universe of discourse.

Using the scalar gains referred above, rule base a fuzzy model reference learning controller is implemented and results of the same are compared with traditional PD as well as neuro-fuzzy controllers.

Table 8.1 : Rule base for Fuzzy Inverse Model

C_{ij}		$\tilde{\Psi}_c^j$										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
$\tilde{\Psi}_c^i$	-5	1	1	1	1	1	1	0.8	0.6	0.4	0.2	0
	-4	1	1	1	1	1	0.8	0.6	0.4	0.2	0	-0.2
	-3	1	1	1	1	0.8	0.6	0.4	0.2	0	-0.2	-0.4
	-2	1	1	1	0.8	0.6	0.4	0.2	0	-0.2	-0.4	-0.6
	-1	1	1	0.8	0.6	0.4	0.2	0	-0.2	-0.4	-0.6	-0.8
	0	1	0.8	0.6	0.4	0.2	0	-0.2	-0.4	-0.6	-0.8	-1
	1	0.8	0.6	0.4	0.2	0	-0.2	-0.4	-0.6	-0.8	-1	-1
	2	0.6	0.4	0.2	0	-0.2	-0.4	-0.6	-0.8	-1	-1	-1
	3	0.4	0.2	0	-0.2	-0.4	-0.6	-0.8	-1	-1	-1	-1
	4	0.2	0	-0.2	-0.4	-0.6	-0.8	-1	-1	-1	-1	-1
5	0	-0.2	-0.4	-0.6	-0.8	-1	-1	-1	-1	-1	-1	

Evolutionary Algorithm Design

GA is employed to minimize the heading error and change in heading error between the desired and output. GA implemented has following parameter setting

- No. of Traits : 2
- Evaluation function : ISE (Integrated Squared Error), Minimization
- Termination Option : Maximum Generation
- Max. No. of Generations : 40;
- Selection : Normal selections
- Cross over : Arithmetic Crossover (probability = 0.8)
- Mutation : Uniform Mutations (probability = 0.05)

As discussed in earlier chapters, pure GA will not be suitable option for the real time control application and hence GA is embedded into FMRLC to obtain better real time performance of the application under consideration.

8.2 Simulation

To design the PD controller response surface methodology is used, where range of gains K_p & K_d are defined as [-5, -0.5] and [-500, -100] respectively. There are 20 grid points identified for each, meaning that total 400 points identified on the surface. The cost function, representing the minimization of squared error, is used to evaluate all those points and the best three are

suggested for implementation. The grid formed and the surface is shown in figure 8.1 and 8.2, respectively.

The best values of gains K_p & K_d obtained using this are given below:

K_p	-3.3421	-2.8684	-3.1053	-2.9345
K_d	-500.000	-500.0000	-500.0000	436.6634

Using $K_p = -3.1053$ & $K_d = -500.00$, to implement the PD controller, the response obtained is as given in the Figure 8.3, assuming that heading sensor noise and wind disturbance is zero.

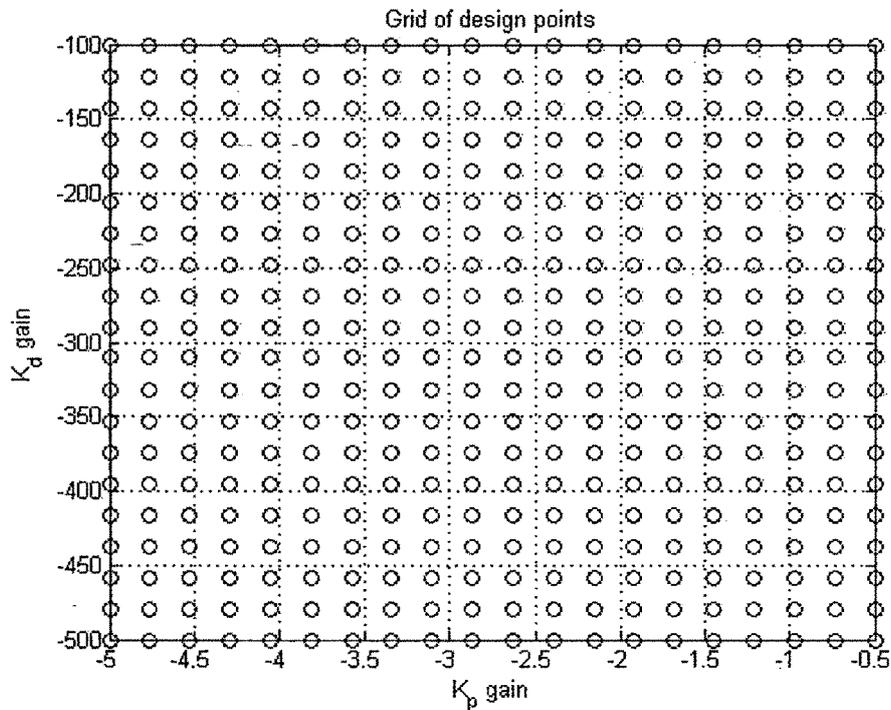


Figure 8.1: Grid points to search for best K_p & K_d

Figure 8.4 show the response of the tanker ship with the effect of the heading sensor noise is introduced as random value in the range of $[-0.01 +0.01]$ degree.

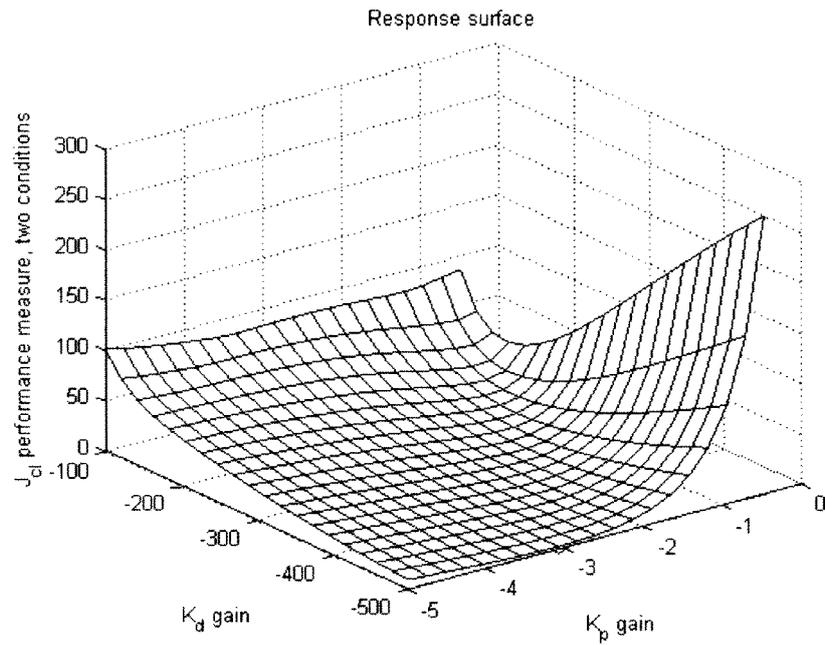


Figure 8.2: Response Surface for selection of best Kp & Kd

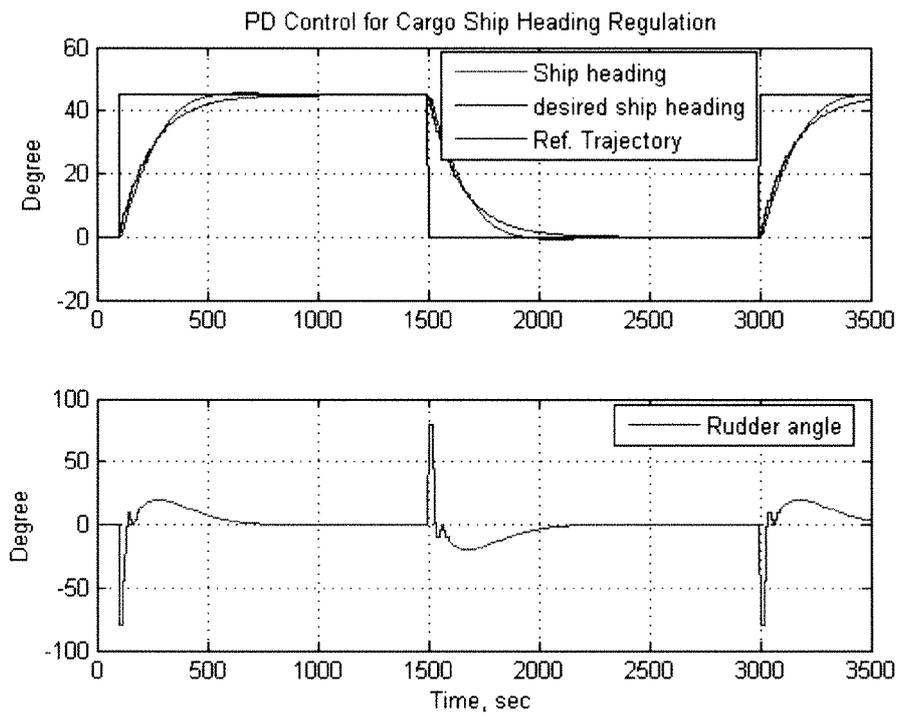


Figure 8.3 PD Control of Cargo ship $K_p=-3.1053$, $K_d = -500.00$, without Noise

The Fuzzy controller with the cargo ship steering control is also implemented and results of the same are shown in the figure 8.5, whereas figure 8.6 shows the Fuzzy controller input – output mapping. The scalar gains used are $g_e=2/\pi, g_c=300, g_u=4*\pi/18$. Figure 8.7 shows the response of Fuzzy controller with scalar gains $g_e=2/\pi, g_c=300, g_u=8*\pi/18$ whereas Figure 8.8 shows the response of Fuzzy controller with scalar gains $g_e=1/\pi, g_c=100, g_u=8*\pi/18$. Referring figure 8.5, figure 8.7 and figure 8.8, we can clear understand the effect of scalar gains on the fuzzy controller, which are going to be used while implementing FMRLC.

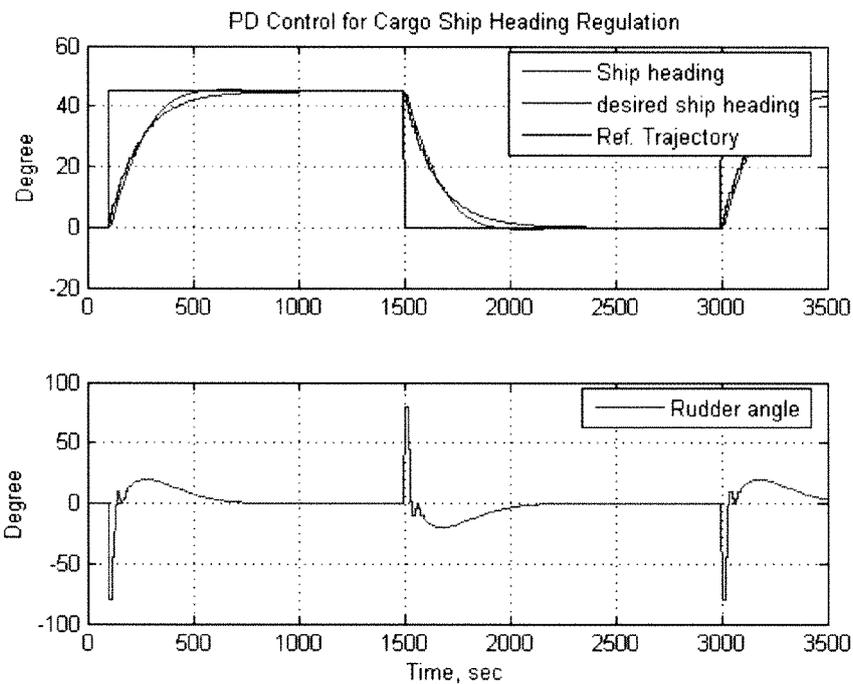


Figure 8.4 PD Control of Cargo ship $K_p=-3.1053, K_d = -500.00$, with Noise

Figure 8.9 show the response of cargo ship implemented using Artificial Neural Network with Multilayer Perceptron (MLP). The neural network is defined using 3 hidden layers and three nodes in each layer. Back propagation algorithm is used to train the network. Figure 8.10 shows the input output mapping of the neural network implementation.

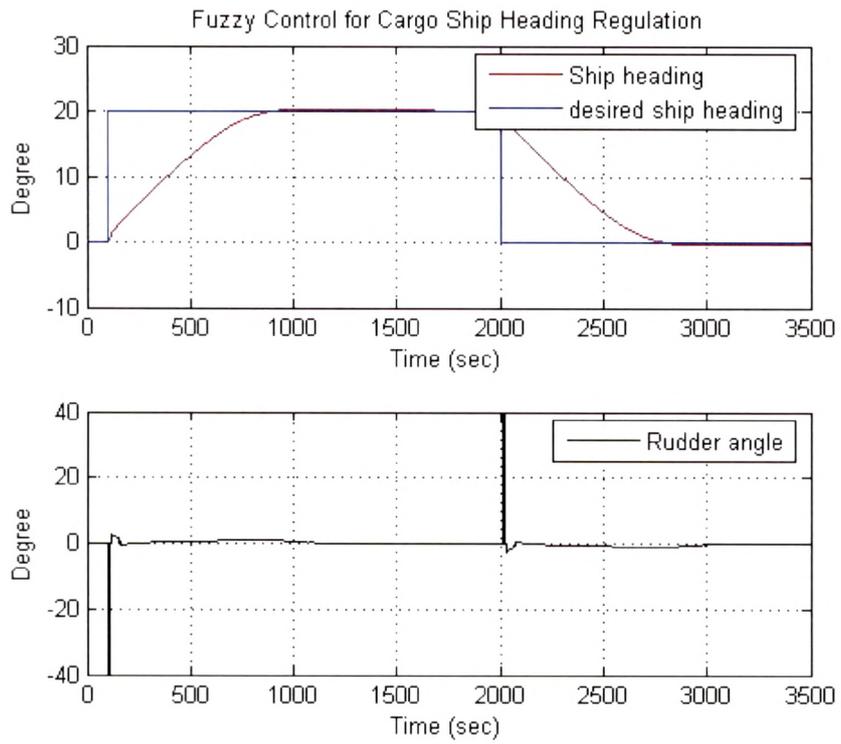


Figure 8.5: Fuzzy controller, with scalar $g_c=2/\pi, g_c=300, g_u=4*\pi/18$

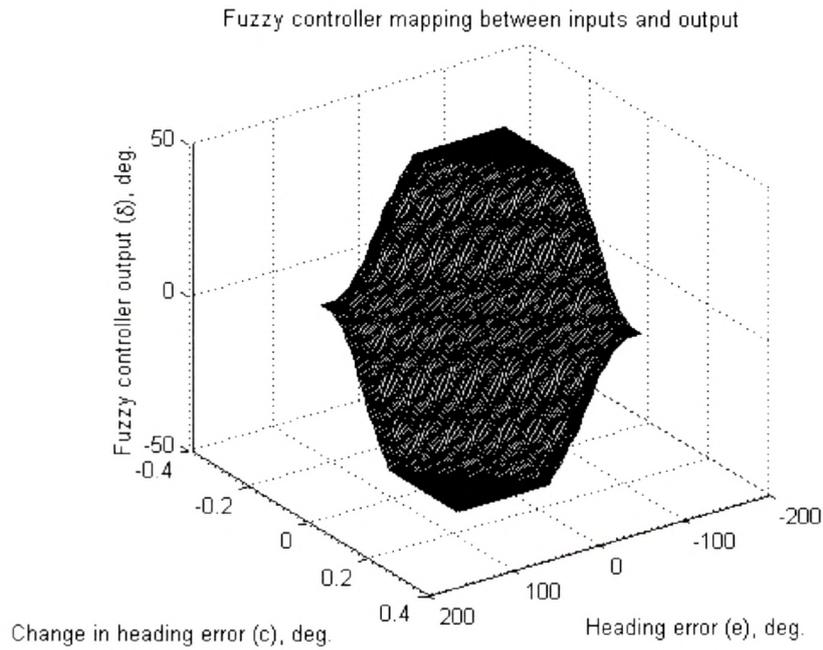


Figure 8.6: Fuzzy controller input output mapping

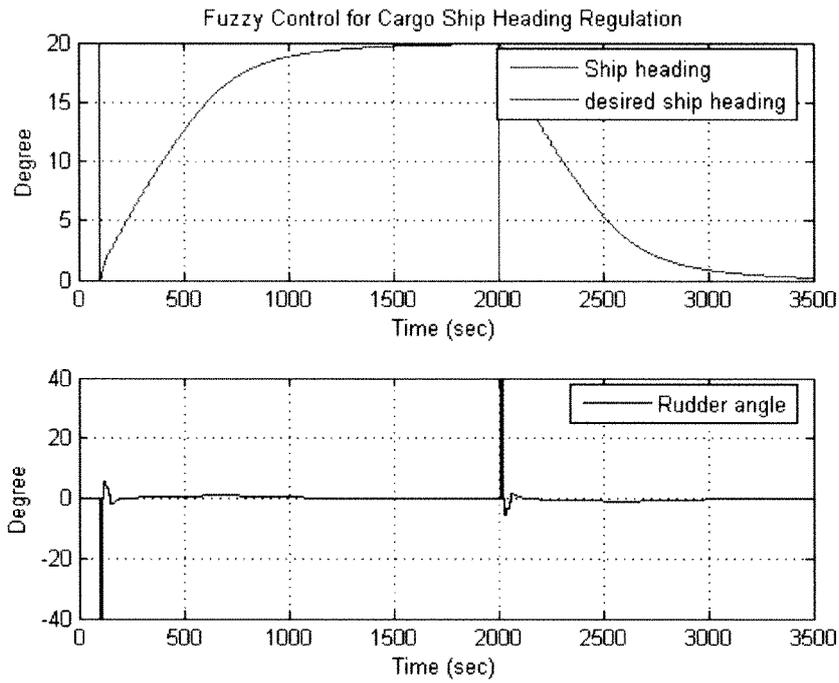


Figure 8.7: Fuzzy controller, with scalar $g_e=2/\pi, g_c=300, g_u=8*\pi/18$

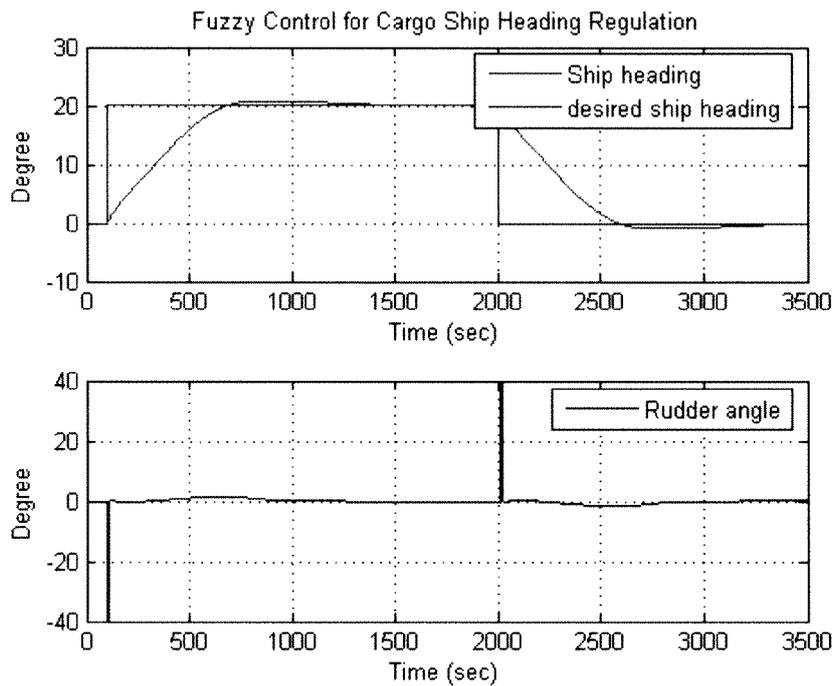


Figure 8.8: Fuzzy controller, with scalar $g_e=1/\pi, g_c=100, g_u=8*\pi/18$

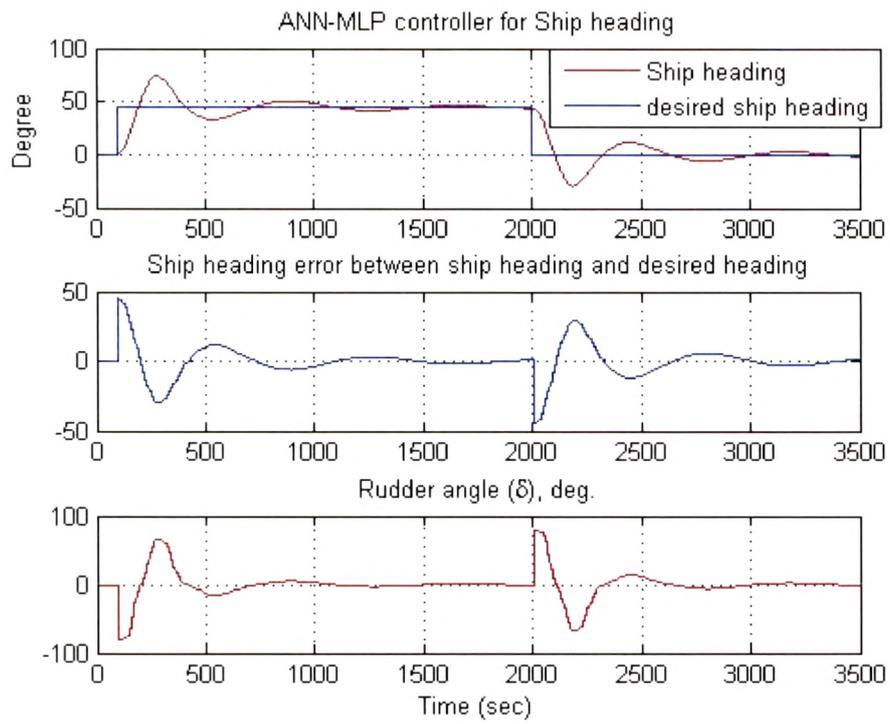


Figure 8.9: ANN with MLP for ship heading

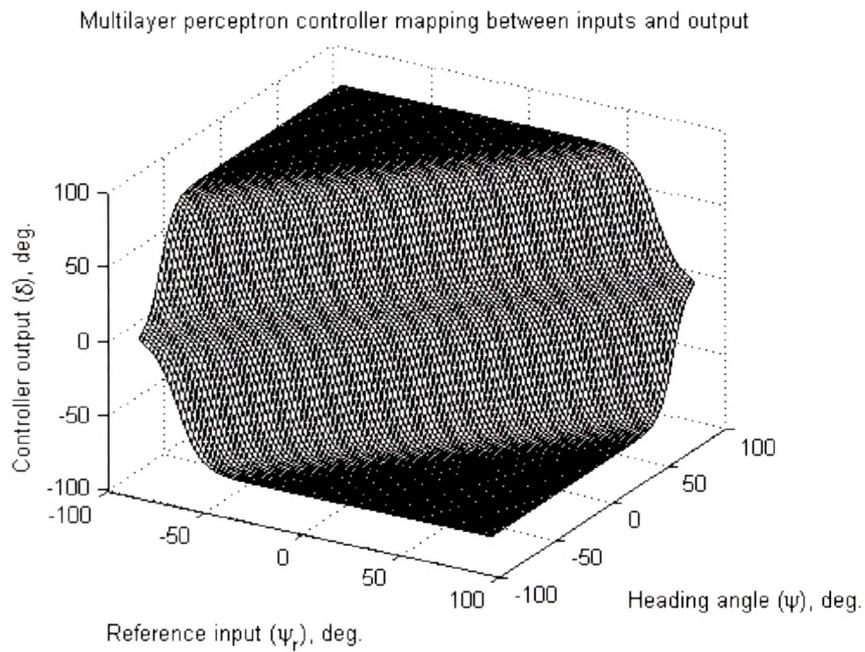


Figure 8.10: MLP mapping between inputs and output

Figure 8.11 shows the implementation of ship heading regulation with the use of Radial Basis function Neural network. Figure 8.12 shows the grid of the receptive field unit centres, figure 8.13 to show the radial basis function output and Figure 8.14 shows the mapping of input-output.

Figure 8.15 show the response of actual implementation of FMRLC design. The scalar gains used in the fuzzy controller are $g_c=1/\pi, g_c=100, g_u=8*\pi/18$ as the first approximation. The fuzzy inverse model also have the scalar gains as defined earlier i.e. $g_{yc}=1/\pi, g_{yc}=10, g_p=0.4$. The first order reference model is used. The rules base of the fuzzy controller obtained after learning is as given in table 8.2.

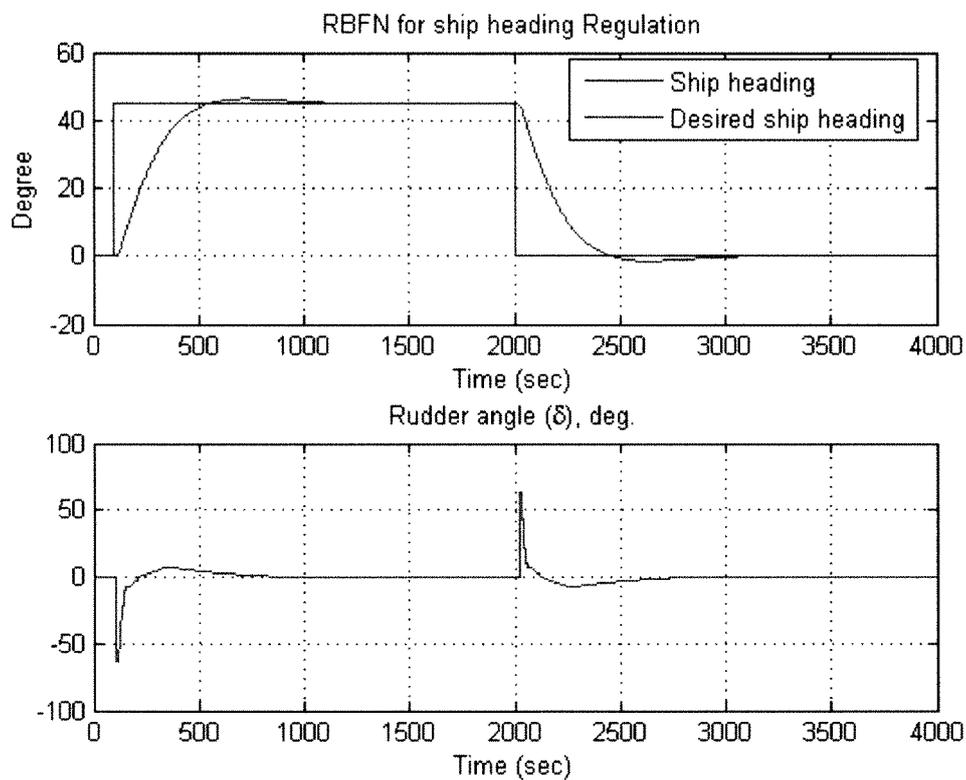


Figure 8.11: Response of RBFN for ship heading Regulation

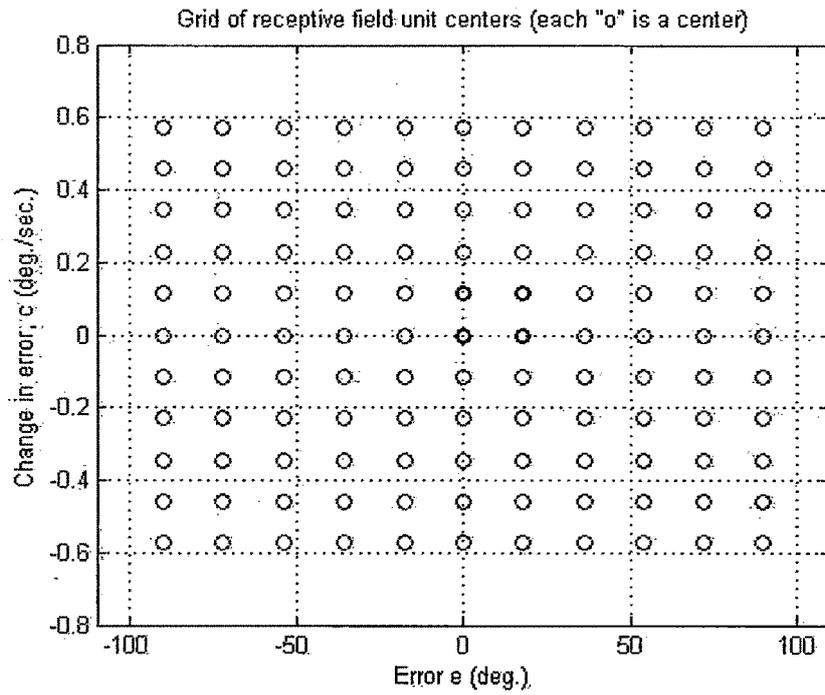


Figure 8.12: Grid of Receptive unit Centres for RBFN

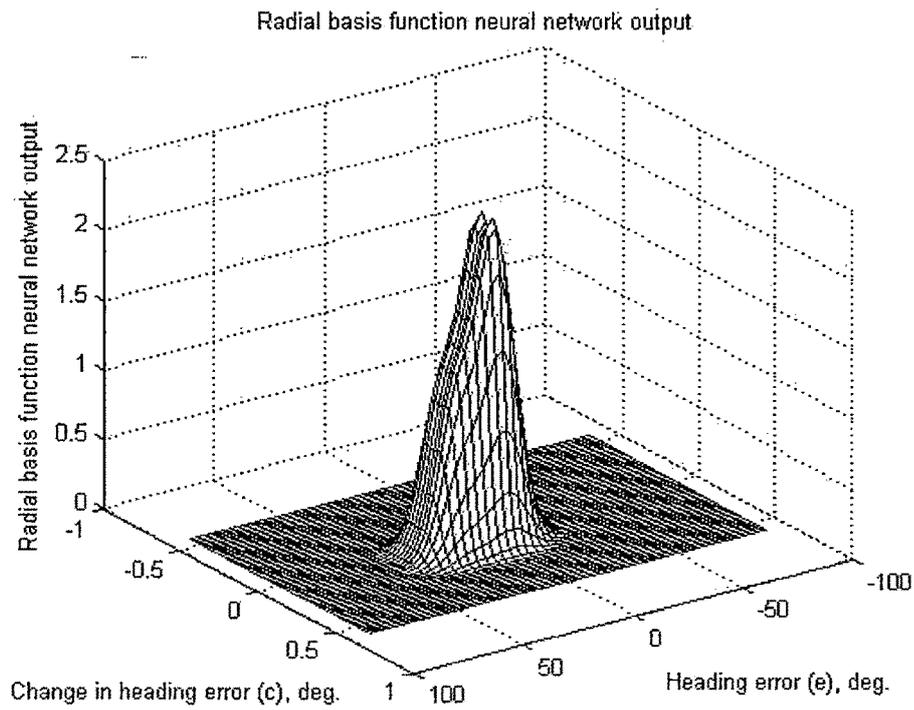


Figure 8.13: Radial Basis Function NN output at grid centers

Radial basis function neural network controller mapping between inputs and output

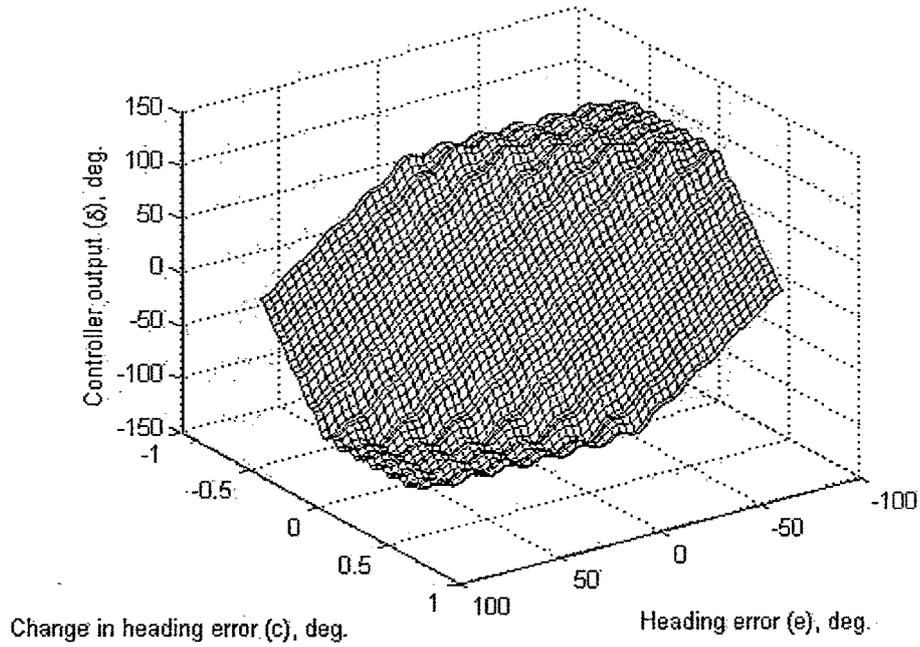


Figure 8.14: RBFN controller mapping of input-output

Table 8.2 : Rule Base of Fuzzy controller, after leaning – FMRLC

	-5	-4	-3	-2	-1	0	1	2	3	4	5
-5	1.40	1.40	1.40	1.40	1.40	1.40	1.12	0.84	0.42	0.14	0.00
-4	1.40	1.40	1.40	1.40	1.40	1.12	0.84	0.42	0.14	0.00	-0.14
-3	1.40	1.40	1.40	1.40	1.12	0.84	0.42	0.14	0.00	-0.14	-0.42
-2	1.44	1.40	1.40	1.12	0.84	0.46	0.36	0.19	-0.14	-0.42	-0.84
-1	1.44	1.40	1.12	0.92	0.90	0.76	0.33	-0.04	-0.42	-0.84	-1.12
0	1.40	1.12	0.84	0.55	0.62	0.07	-0.50	-0.50	-0.84	-1.12	-1.40
1	1.12	0.84	0.42	-0.06	-0.46	-0.86	-0.88	-0.84	-1.12	-1.40	-1.48
2	0.84	0.42	0.14	-0.25	-0.60	-0.62	-0.84	-1.12	-1.40	-1.40	-1.48
3	0.42	0.14	0.00	-0.14	-0.42	-0.84	-1.12	-1.40	-1.40	-1.40	-1.40
4	0.14	0.00	-0.14	-0.42	-0.84	-1.12	-1.40	-1.40	-1.40	-1.40	-1.40
5	0.00	-0.14	-0.42	-0.84	-1.12	-1.40	-1.40	-1.40	-1.40	-1.40	-1.40

Figure 8.16 shows response of fuzzy inverse model, error in heading & change in heading between ship heading and desired ship heading, figure 8.17 shows error in heading & change in heading between output and reference model. Figure 8.18 show the nonlinear mapping of input and output.

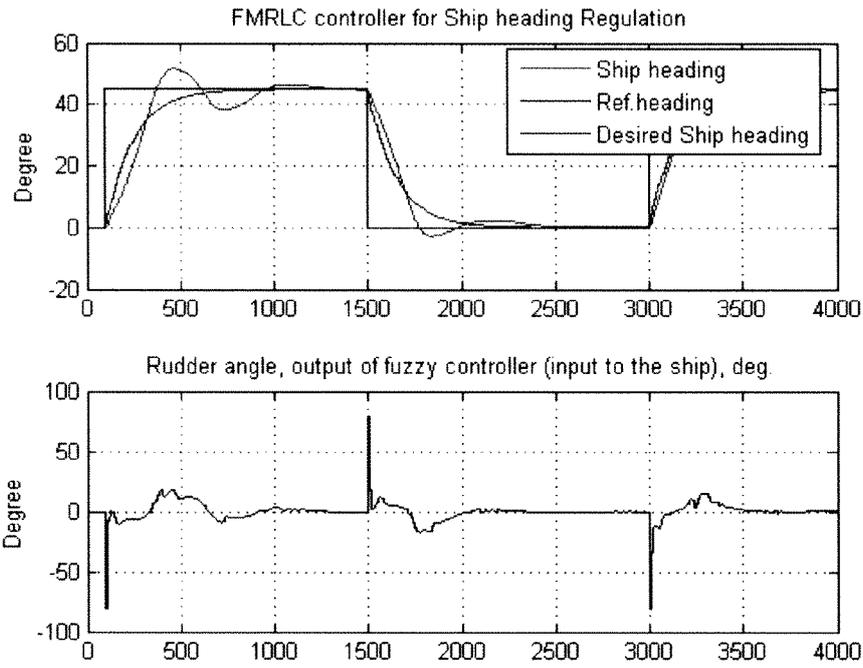


Figure 8.15: FMRLC with scalar gains $g_c=1/\pi, g_v=100, g_0=8*\pi/18$, without Noise

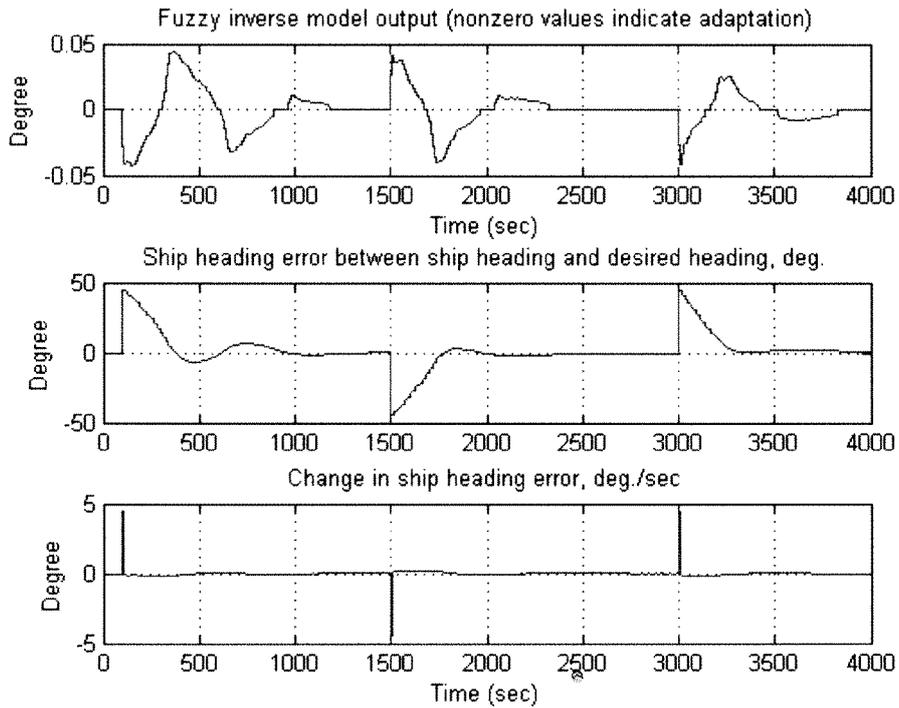


Figure 8.16: FMRLC Fuzzy Inverse model response & Heading Errors without noise

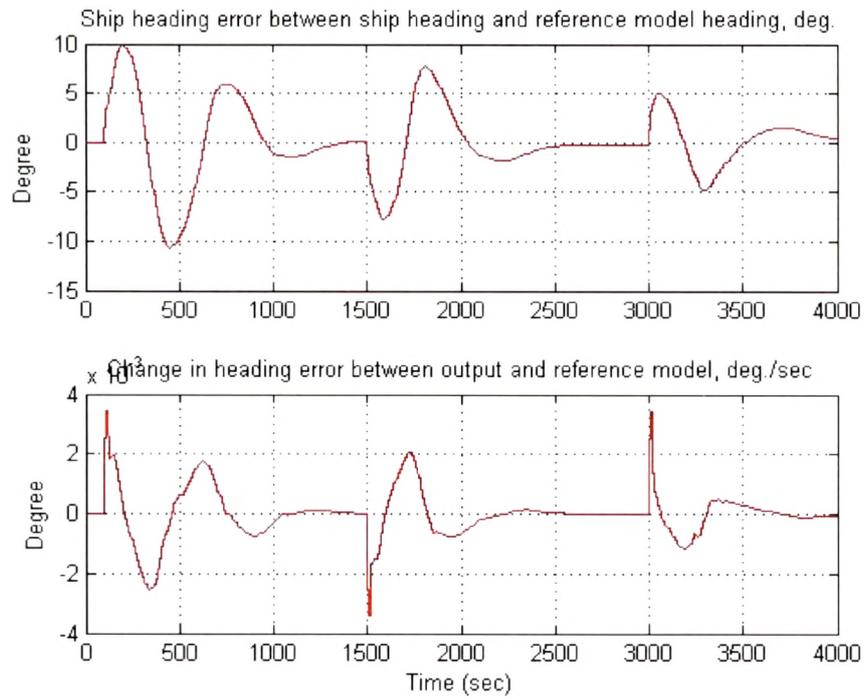


Figure 8.17: FMRLC : Heading and change in heading error, without noise

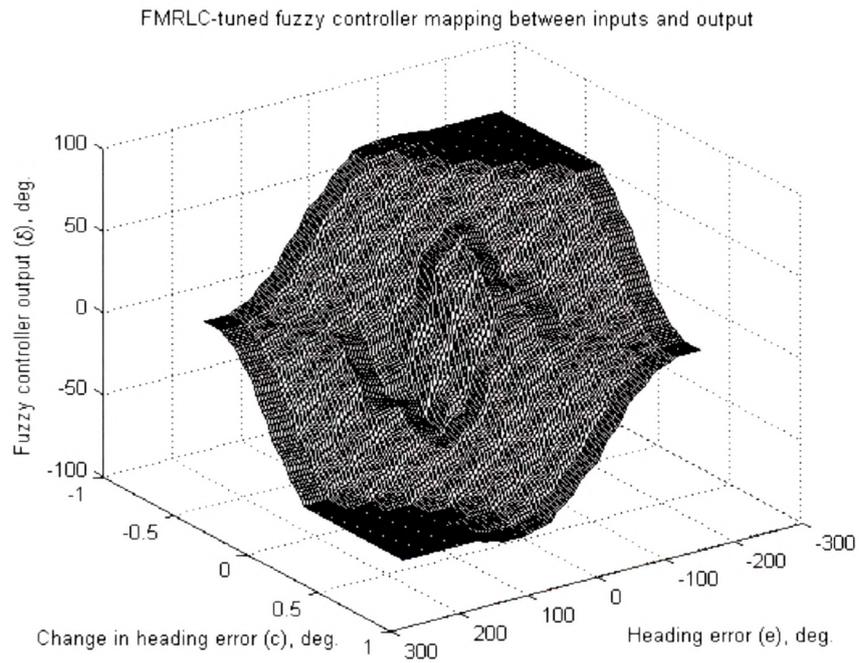


Figure 8.18: FMRLC controller mapping between input and output.

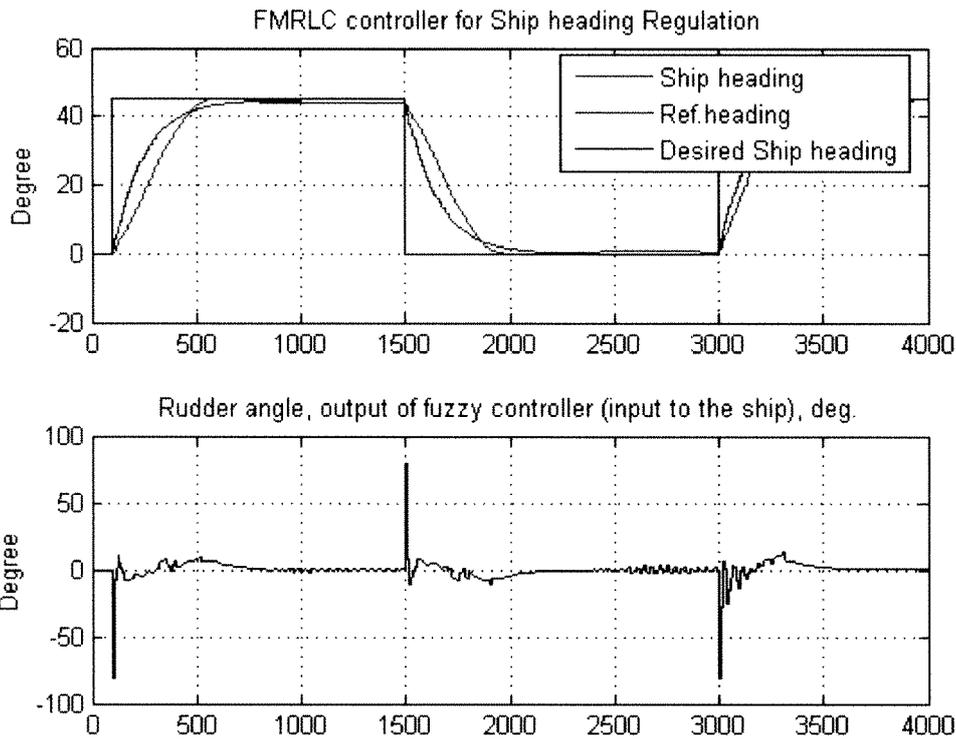


Figure 8.19 : FMRLC with scalar gains $g_c=2/\pi, g_c=250, g_u=8*\pi/18$, without Noise

Figure 8.19, Figure 8.20 and Figure 8.21 show the response of FMRLC with scalar gains for fuzzy controller as $g_c=2/\pi$, $g_c=250$, and $g_u=8*\pi/18$ obtained with tuning using simple fuzzy controller with first order reference model without considering heading sensor noise or wind disturbances.

Results of the FMRLC implementation with Heading sensor noise & Wind disturbance are shown in Figure 8.22, Figure 8.23 and Figure 8.24, with scalar gains for fuzzy controller as $g_c=2/\pi$, $g_c=250$, and $g_u=8*\pi/18$ obtained with tuning using simple fuzzy controller with first order reference model. Uniform distributed random noise is considered in the ship reading in the range ± 0.01 degree. Wind disturbance considered to be as an additive sine disturbance to the rudder input. It is of amplitude of 0.5 deg and its period is 1000 sec.

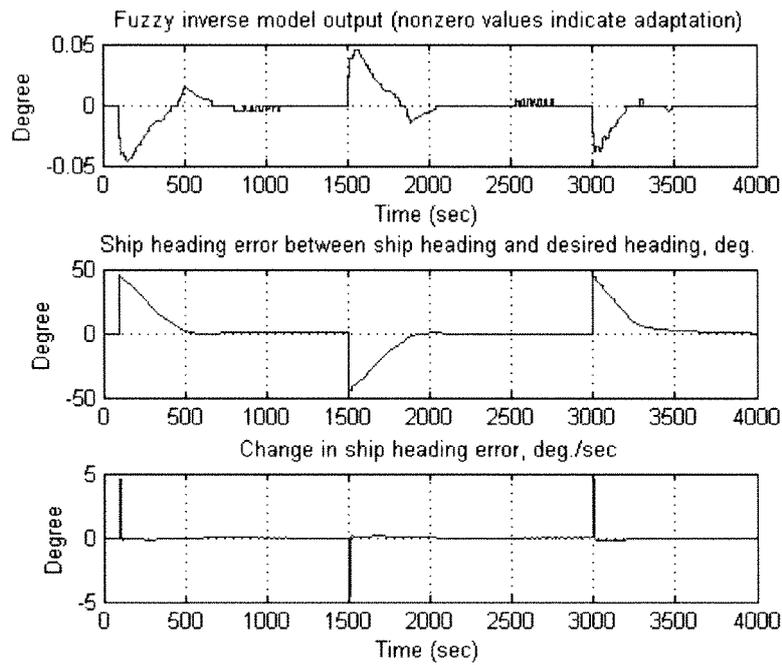


Figure 8.20: FMRLC Fuzzy Inverse model response & Heading Errors without noise

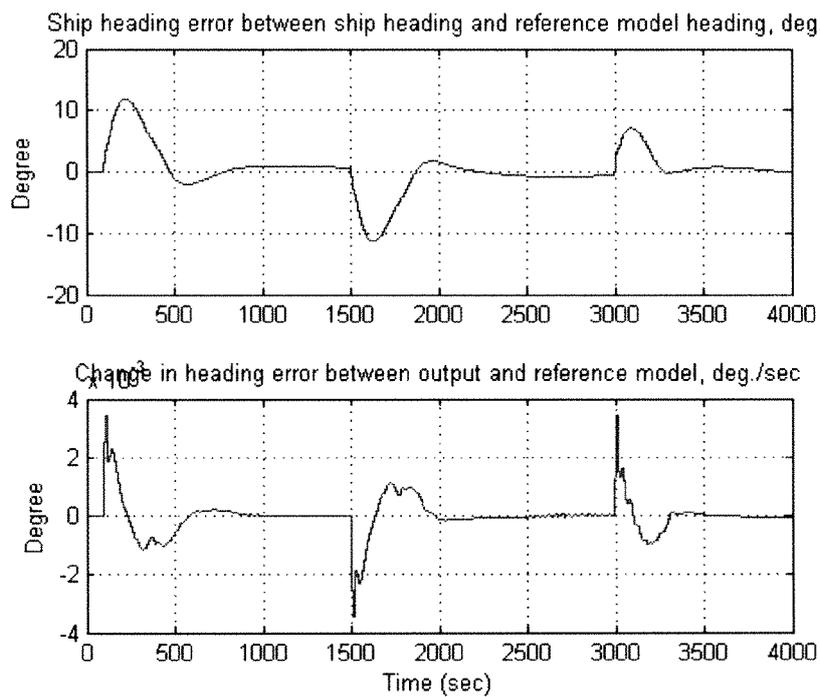


Figure 8.21: FMRLC : Heading and change in heading error, without noise

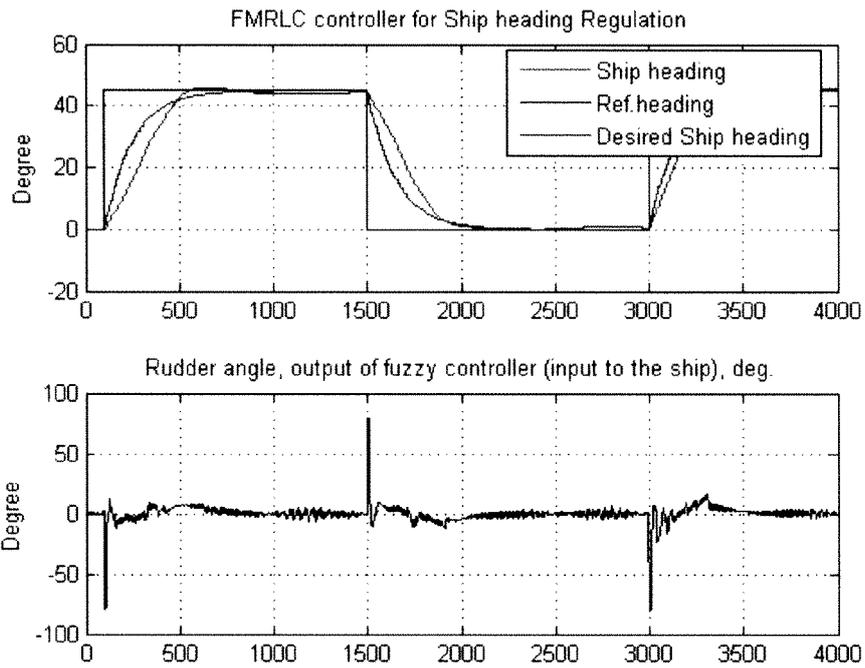


Figure 8.22: FMRLC with scalar gains $g_c=2/\pi, g_c=250, g_u=8*\pi/18$, with uncertainty

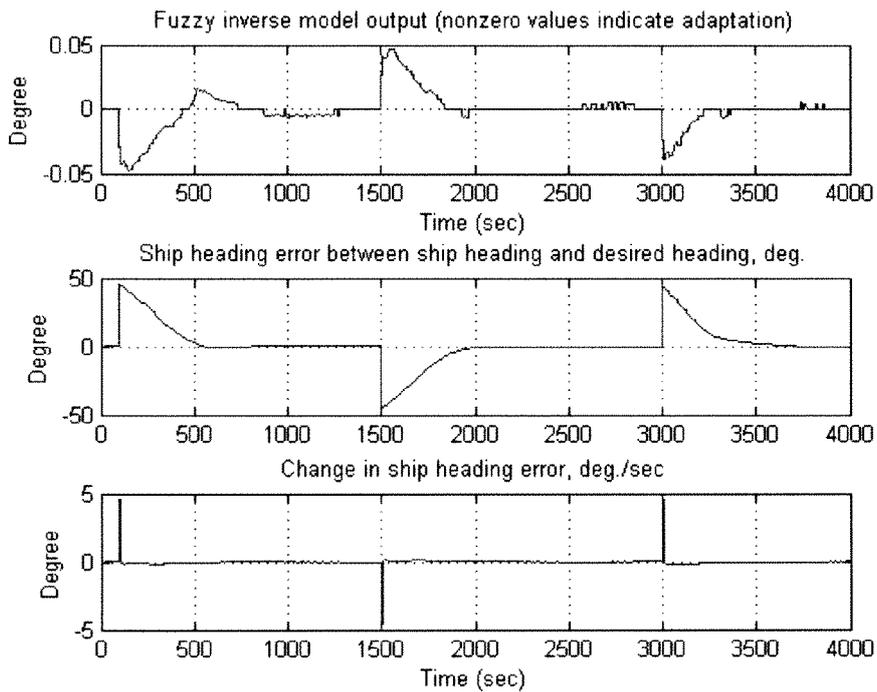


Figure 8.23: FMRLC Fuzzy Inverse model response & Heading Errors with uncertainty

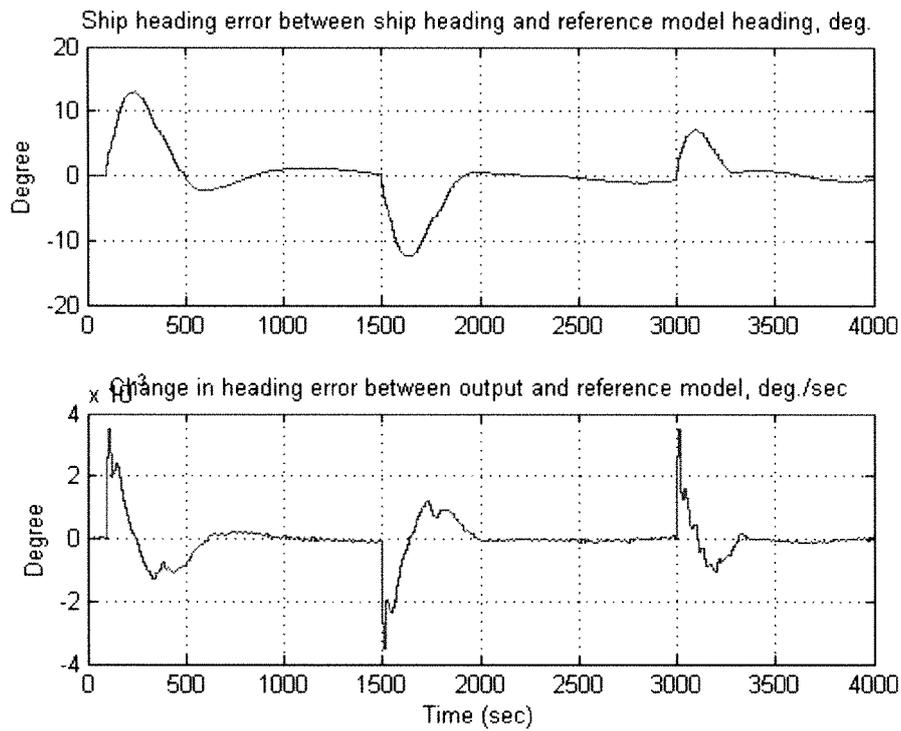


Figure 8.24 : FMRLC : Heading and change in heading error, with uncertainty

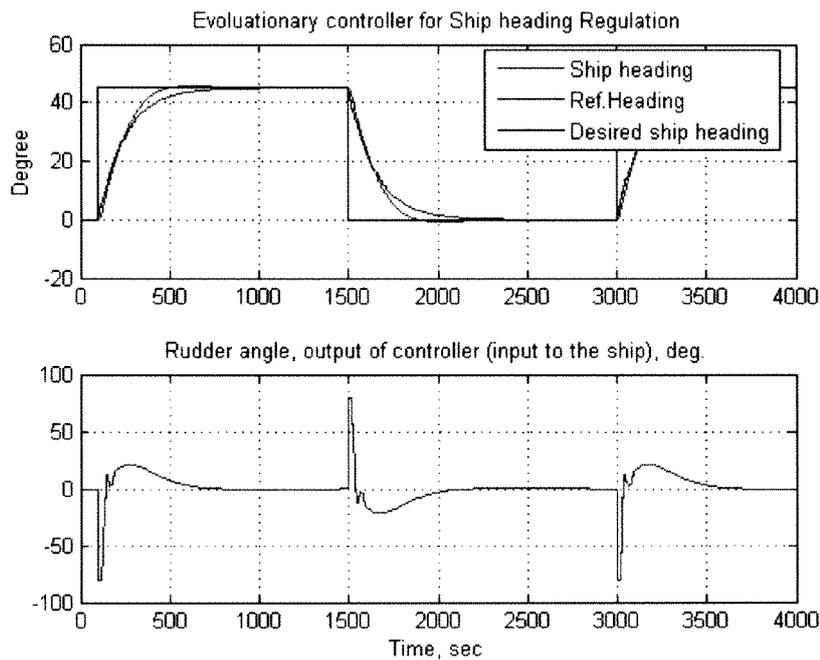


Figure 8.25: Ship Steering with Embedding Evolutionary Algorithm in FMRLC

Figure 8.25 & Figure 8.26 shows the response of the ship steering after embedding GA into FMRLC and Figure 8.27 shows improvement in evaluation function over the generation.

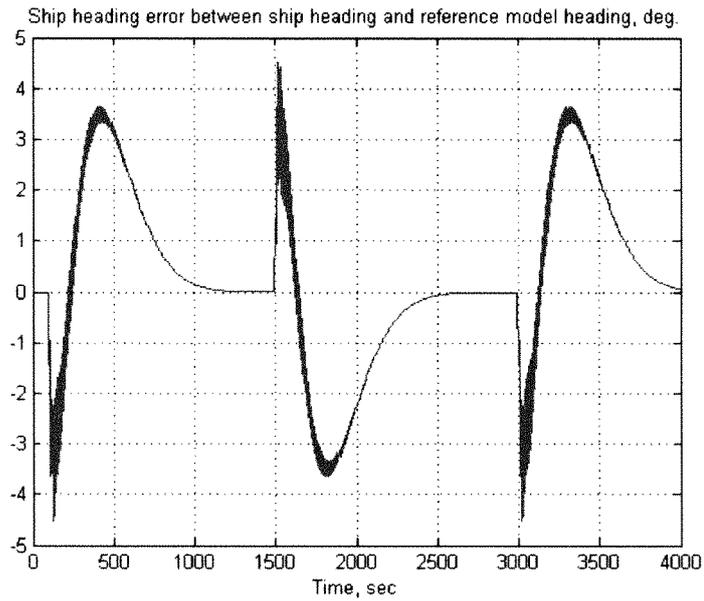


Figure 8.26: Heading and change in heading error for Evolutionary Algorithm

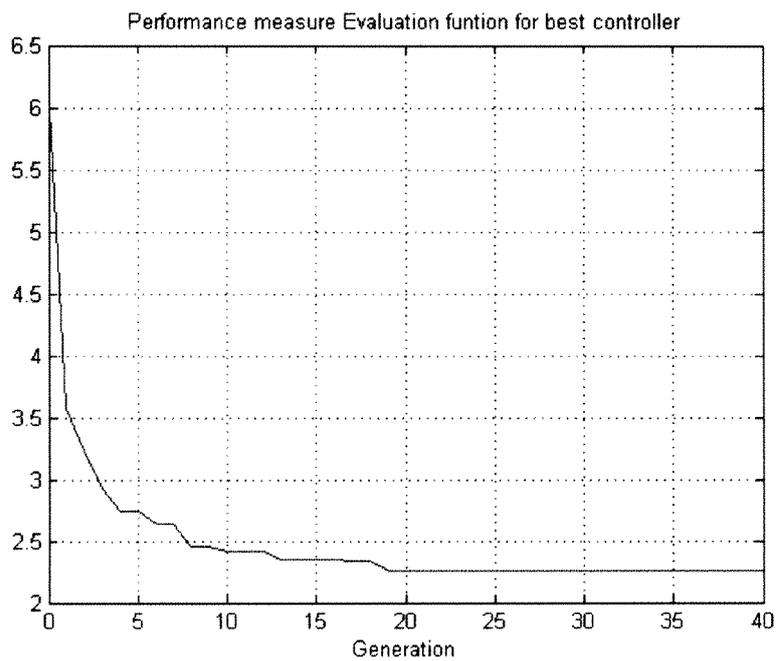


Figure 8.27: Evolution Function.

8.3 Fault Tolerant Aircraft Application

Nominal Control Law

The nominal control laws for the aircraft are for the lateral channel as well as for the longitudinal channel. The lateral channel is shown in figure 8.28. The inputs to the controller are pilot commands and the system feedback signal. Pilot commands for the longitudinal channel is desired pitch A_{zd} and for lateral channel are desired roll rate p_d & desired side slip β_d . The controller gains for both the channels are function of different dynamic pressures \bar{q} . At 499.24 psf, for lateral channel the value of the gain matrix elements $K(\bar{q})$ is given below, assuming constant speed and altitude of aircraft.

$$K(499.24) = \begin{bmatrix} 0.47 & 0.14 & 0.14 & -0.56 & -0.38 \\ -0.08 & -0.056 & 0.78 & -1.33 & -4.46 \end{bmatrix} \quad \dots(8.5)$$

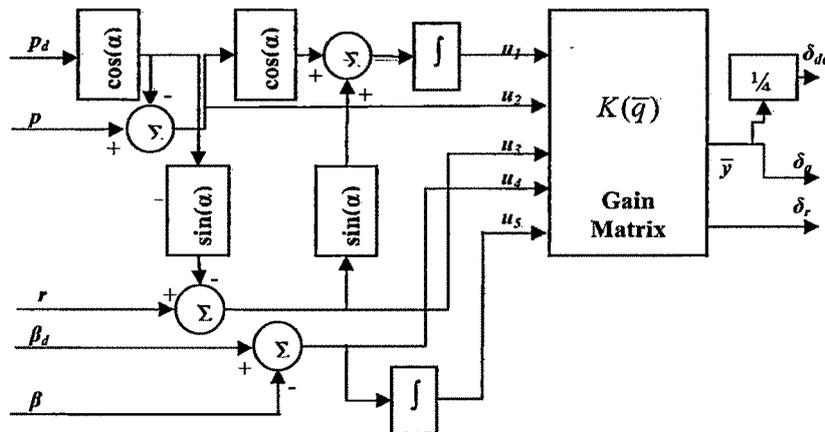


Figure 8.28 : Nominal control law for lateral channel.

The transfer function $\frac{20}{s+20}$ is used to represent the actuator dynamics for each of the aircraft control surfaces, and the actuators have physical saturation limits so that $-21^\circ \leq \delta_e \leq 21^\circ$, $-21^\circ \leq \delta_{de} \leq 21^\circ$, $-23^\circ \leq \delta_a \leq 20^\circ$, and $-30^\circ \leq \delta_r \leq 30^\circ$. The actuator rate saturation is $\pm 60^\circ/\text{sec}$ for all the actuators.

To simulate the closed-loop system, we interpolate between the five perturbation models based on the value of α , which produces a nonlinear simulation of the F-16. For all the

simulations, a special “loaded roll command sequence” is used. This command sequence is as follows:

At time $t = 0.0$, a $60^\circ/\text{sec}$ roll rate command (p_d) is held for 1 second.

At time $t = 1.0$, a 3g pitch command (A_{zd}) is held for 9 seconds.

At time $t = 4.5$, a $-60^\circ/\text{sec}$ roll rate command (p_d) is held for 1.8 seconds.

Finally, at time $t = 11.5$, a $60^\circ/\text{sec}$ roll rate command (p_d) is held for 1 second.

The sideslip command β_d is held at zero throughout the sequence.

Failure Scenarios

Many different failures can occur on a high-performance aircraft such as the F-16.

For instance, there are two major types of actuator failures:

1. *Actuator malfunction*: Two main types are possible:

- a. Actuator performance degradation (e.g., a bandwidth decrease).
- b. Actuator stuck at a certain angle (e.g., an arbitrary angle during a motion, or at the maximum deflection).

2. *Actuator damage*: Again, two main types are possible:

- a. Actuator damaged so that the control surface oscillates in an uncontrollable fashion.
- b. Control surface loss due to severe structural damage.

Here, focus is on the actuator malfunctions for the F-16.

FMRLC for the F-16

A MIMO FMRLC for the fault-tolerant aircraft control application is required to be developed. The basic structure for the FMRLC is shown in Figure 7.1 is used with a slightly different notation for the variables. In particular, we use bar for vector quantities so that $\bar{y}_r(kT)$ is the vector of reference inputs, $\bar{y}_f(kT)$ is the vector of outputs from the MIMO fuzzy inverse model, $\bar{e}(kT)$ is the vector of error inputs to the fuzzy controller, $\bar{y}_e(kT)$ is the vector of error inputs to the inverse model, and $\bar{c}(kT)$ and $\bar{y}_c(kT)$ are the change-in-error vectors to the fuzzy controller and inverse model, respectively. The scaling gains are denoted as, for example, $\bar{g}_e = [g_{e1}, \dots, g_{es}]$, if there are s inputs to the fuzzy controller. Similarly for the other scaling

gains (the gains on the inverse model output $\bar{y}_f(kT)$ will be denoted with \bar{g}_f) so that $g_{ei}ei(kT)$ is an input to the fuzzy controller. The gains \bar{g}_e are chosen so that the range of values of $g_{ei}ei(kT)$ lies on $[-1, 1]$, and \bar{g}_u is chosen by using the allowed range of inputs to the plant in a similar way. The gains \bar{g}_c are determined by experimenting with various inputs to the system to determine the normal range of values that $\bar{c}(kT)$ will take on; then \bar{g}_c is chosen so that this range of values is scaled to $[-1, 1]$. We utilize r MISO fuzzy controllers, one for each process input u_n , equivalent to using one MIMO controller.

To begin the design of the FMRLC, it is important to try to use some intuition that we have about how to achieve fault-tolerant control. For instance, generally it is not necessary to utilize all the control effectors to compensate for the effects of the failure of a single actuator on the F-16. If the ailerons in the lateral channel fail, the differential elevators can often be used for compensation, or vice versa. However, the elevators may not aid in reconfiguration for an aileron failure unless they are specially designed to induce moments in the lateral channel. Hence, it is sufficient to redesign only part of the nominal controller to facilitate control reconfiguration. Here, we will replace the $K(\bar{q})$ portion of the lateral nominal control laws, refer figure 8.28, with a fuzzy controller and let the learning mechanism of the FMRLC tune the fuzzy controller to perform control reconfiguration for an aileron failure, refer figure 8.29.

To apply the FMRLC in the F-16 reconfigurable control application, it is of fundamental importance that for an unimpaired aircraft, the FMRLC must behave at least as good as (indeed, the same as) the nominal control laws. In normal operation, the learning mechanism is inactive or used only to maintain the aircraft performance at the level of specified reference models. In the presence of failures, where the performance becomes different from the specified reference model, the learning mechanism can then tune the fuzzy controller to achieve controller reconfiguration.

The Fuzzy/Nominal Controller

Gain matrix block $K(\bar{q})$ of figure 8.28 is replaced by a fuzzy controller in Figure 8.29, which will be adjusted by the FMRLC to reconfigure part of the control laws in case there is a failure. Therefore, to copy the nominal control laws, all that is necessary is for the fuzzy controller to simulate the effects of the portion of the gain matrix $K(\bar{q})$ that affects the aileron and differential elevator outputs. In this way, the FMRLC is provided with the very good initial

guess of the control strategies as nominal control laws are results of years of experience of the designer. The approximate scalar gains at the input to the fuzzy controller are derived using concepts of weighted sum of inputs.

$$[g_0, g_1, g_2, g_3, g_4, g_5] = \left[14, \frac{1}{29.79}, \frac{1}{100}, \frac{1}{100}, \frac{1}{25}, \frac{1}{36.84} \right] \dots (8.6)$$

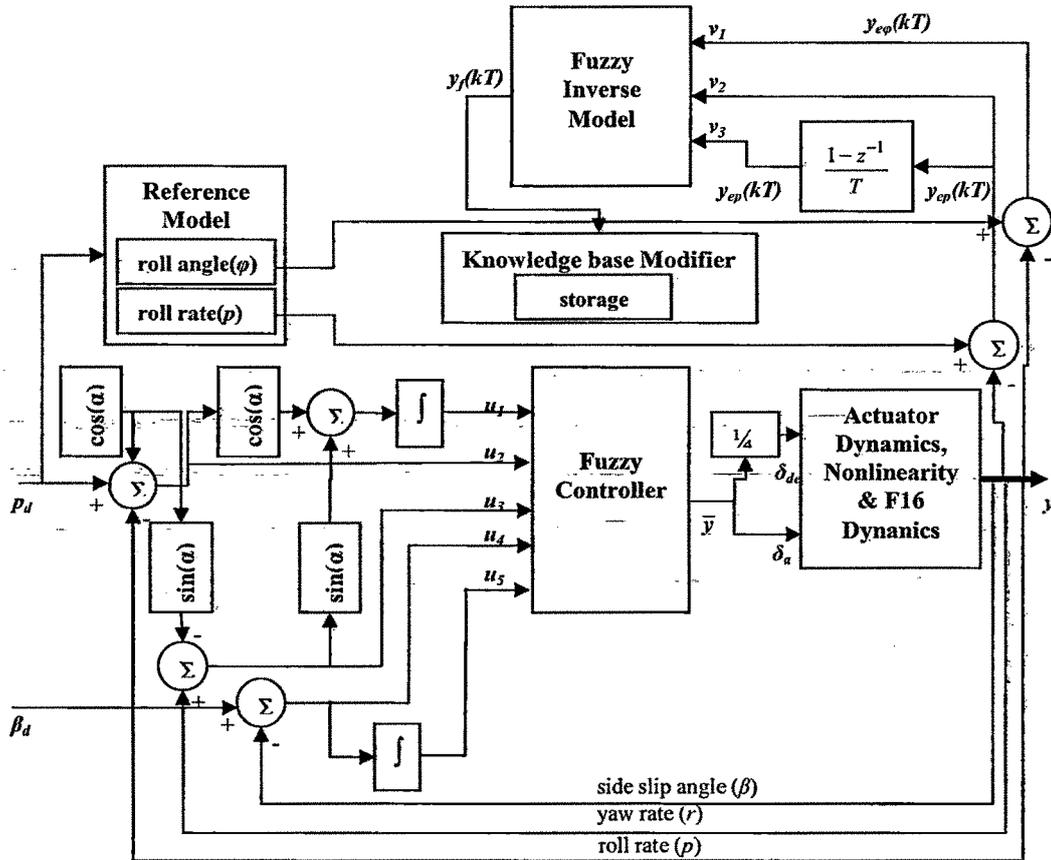


Figure 8.29: FMRLC for F-16 Aircraft.

F-16 Reference Model Design

The reference model is used to characterize the closed-loop specifications such as rise-time, overshoot, and settling time. The performance of the overall system is computed with respect to the reference model by generating error signals between the reference model output and the plant outputs—that is, $y_{e\phi}(kT)$, $y_{ep}(kT)$, and $y_{er}(kT)$ in Figure 8.29.

To achieve the desired performance, the learning mechanism must force $y_{e\phi}(kT) \approx 0$, $y_{ep}(kT) \approx 0$, and $y_{ep}(kT) \approx 0$ for all $k \geq 0$. For the aircraft, the reference model must be chosen so that the closed-loop system will behave similarly to the unimpaired aircraft when the nominal control laws are used, and so that unreasonable performance requirements are not requested. With these two constraints in mind, we choose a second-order transfer function

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \dots(8.7)$$

Where, $\omega_n = \sqrt{200}$ and $\zeta = 0.85$ for the reference models for the roll rate and $H(s)/s$ for the reference model of the roll angle. An alternative choice for the reference model would be to use the actual nominal closed-loop system with a plant model since the objective of this control problem is to design an adaptive controller that will try to make a failed aircraft behave like the nominal non-failed aircraft.

Learning Mechanism Design Procedure

The learning mechanism consists of two parts: 1. a fuzzy inverse model, which performs the function of mapping the necessary changes in the process output error $y_{e\phi}(kT)$, $y_{ep}(kT)$, and $y_{ep}(kT)$ to the relative changes in the process inputs $y_f(kT)$, so that the process outputs will match the reference model outputs, and 2. a knowledge base modifier that updates the fuzzy controller's knowledge-base. The fuzzy inverse model can be considered as another fuzzy controller in the adaptation loop that is used to monitor the error signals $y_{e\phi}(kT)$, $y_{ep}(kT)$, and $y_{ep}(kT)$, and then choose the controller parameters in the main loop in such a way that these errors go to zero. This is the main in the design procedure for the FMRLC for aircraft, which is very much useful for the fault-tolerant control application. Rests of the procedures are same as we discussed in the previous section.

8.4 Simulations

The F-16 aircraft with the FMRLC is simulated using the sampling time T of 0.02 seconds, and tested with an aileron failure at 1 second. Figure 8.30 compares the performance of the FMRLC to the nominal control laws for the case where there is no failure. Figure 8.31 shows the response of impaired aircraft with aileron struck at 1 second.

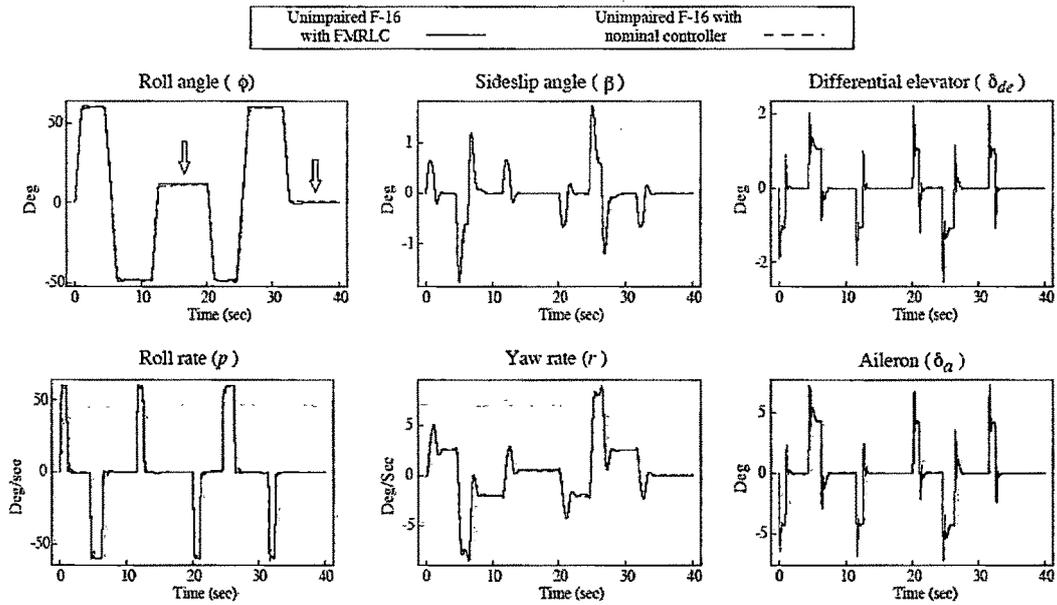


Figure 8.30: Unimpaired F-16 with FMRLC and Nominal Controller

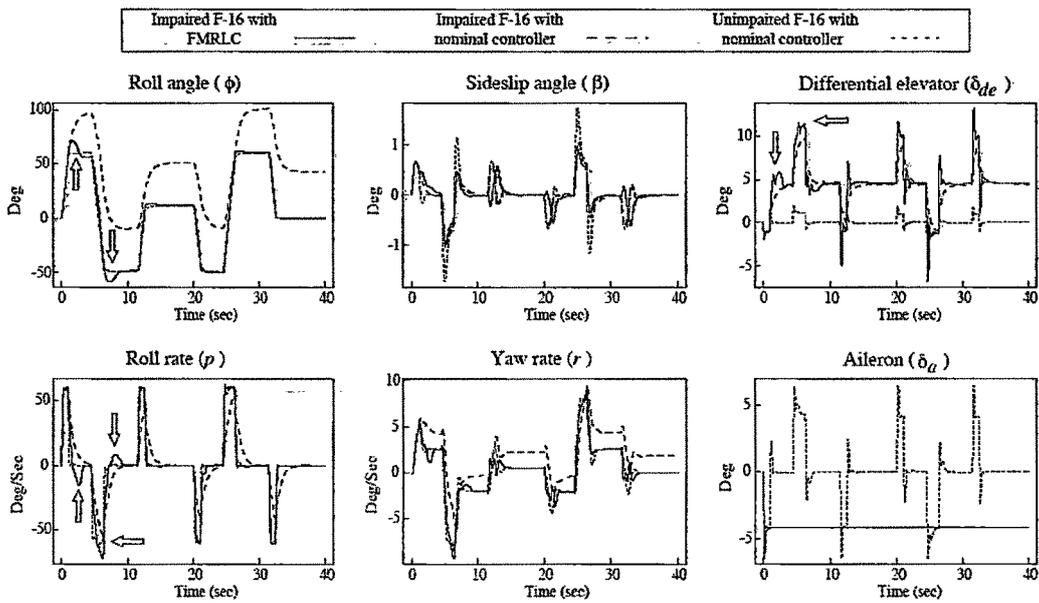


Figure 8.31: Impaired FMRLC with aileron struck at 1 second.

8.5 Helicopter Application

With the data as given in [211], using MATLAB, the **A** & **B** matrices were constructed. In order to make a quick check for the output response using MATLAB, the linear simulation is carried out to test the effect of applying 1 degree of collective pitching for 5 seconds. The simulation shows a linear and continuous vertical acceleration to about 10m/s after 5 seconds. The simulation shows clearly that the system is unstable and it produces diverging response. Also the Eigen values are found to be positive real, meaning that the system is unstable. The system needs a state feedback matrix **K** to ensure that the system is stable. The matrix **K** will be obtained using the method of linear-quadratic (LQ) state-feedback regulator for continuous plant. The value of **K** obtained is

$$K = \begin{bmatrix} 0.2937 & -0.9585 & -0.1035 & 0.0220 & -0.0391 & -0.0145 & 0.5150 & 0.0555 \\ -0.9620 & -0.2837 & 0.3646 & 0.7292 & 0.0489 & 0.0392 & -1.6046 & -0.4184 \\ 0.0973 & -0.0227 & -0.0809 & 0.2695 & 0.8805 & 0.3861 & 0.1500 & -1.7452 \\ -0.0630 & -0.0106 & -0.0628 & 0.1518 & 0.4441 & -1.0105 & 0.0407 & -16.8467 \end{bmatrix}$$

We have used the weighting factor p to be 85 after few iterations to produce the maximum vertical speed at a reasonable collective pitch angle. Using this value of **K** the simulink model shown in figure 8.32

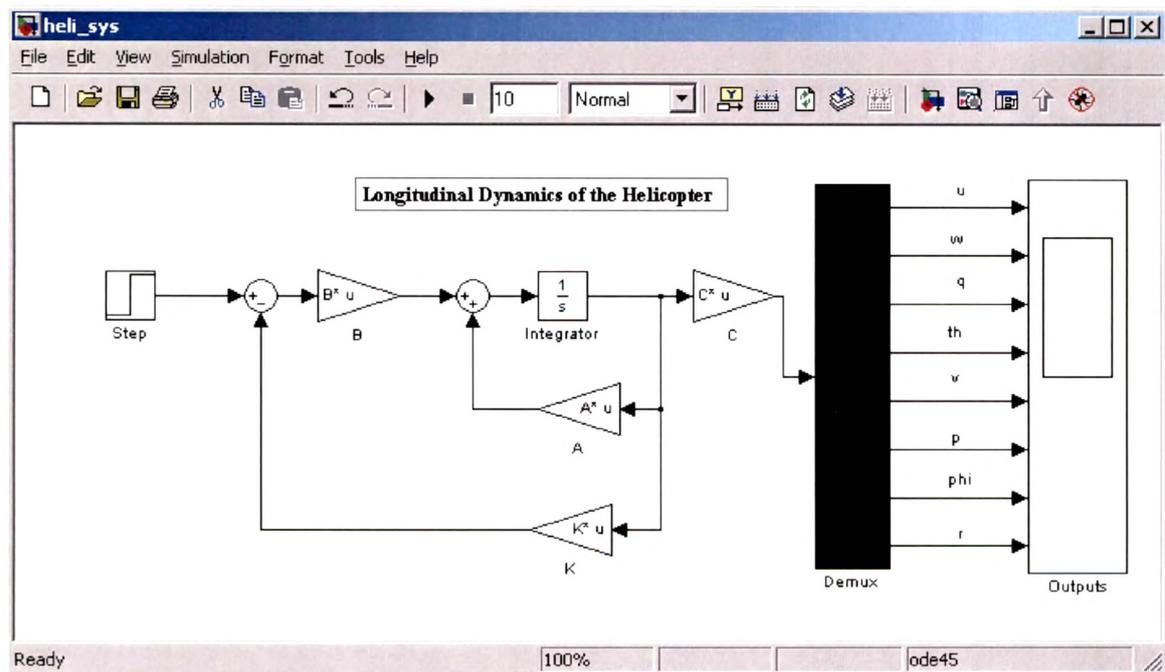


Figure 8.32 Simulink Model of Longitudinal Dynamics of the Helicopter

In order to achieve a maximum climb rate of 1500 ft/min (7.62 m/s), the control input was modified to be: $\theta_0 = 5^\circ$ and $\theta_{ls} = 1.5^\circ$

Notice that in addition to the applied collective pitch angle of 5° , we also have to apply a longitudinal cyclic pitch angle of 1.5° , in order to minimize the forward speed coupling. On the other hand, in order to achieve a forward speed of 100 knots (52 m/s), the control input was modified to be: $\theta_0 = 3^\circ, \theta_{ls} = -10^\circ, \theta_{lc} = 0.7^\circ$. The results obtained for the simulink model shown in 8.32 with above values are shown in figure 8.33.

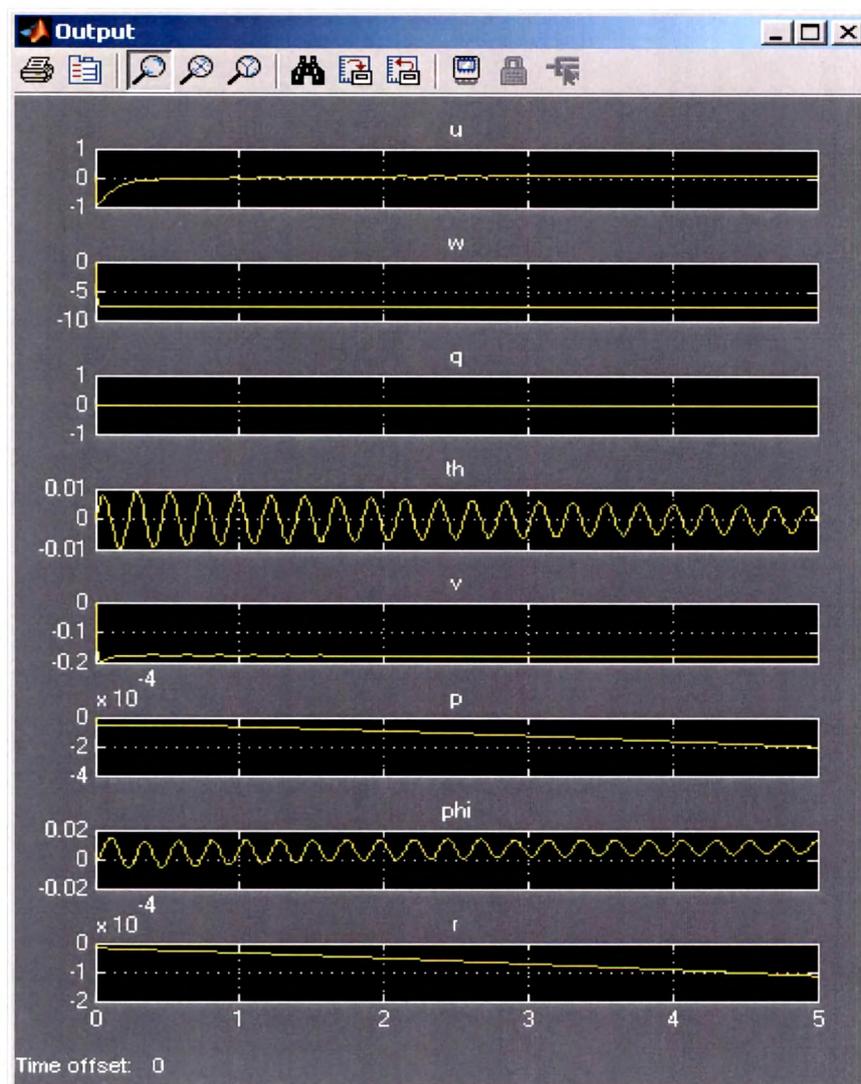


Figure 8.33: Response of helicopter

As more evaluations of the helicopter performance are simulated and better decisions about the best turn rates and speeds to output are made. It is visible that the oscillations in θ is still large, which is not preferable. A PID controller is then implemented for the same and results of the same are shown in figure 8.34.

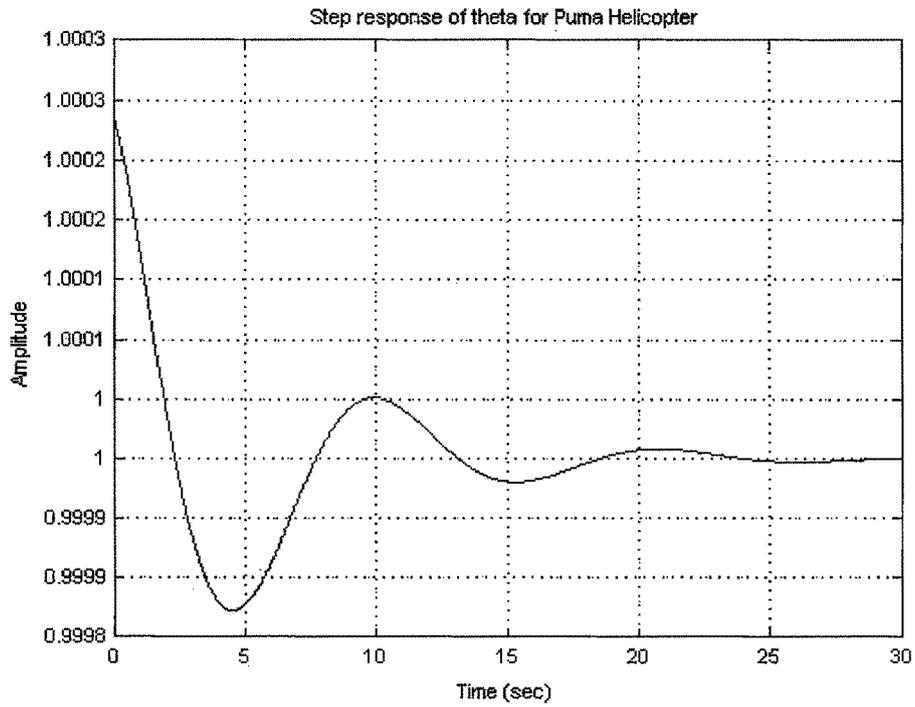


Figure 8.34: Response of Helicopter with GA tuned PID