3.1 Implementing the controller

A good navigational controller is a main requirement for traveling. So the Takagi-Sugeno type fuzzy logic base controller was implemented successfully for navigation [5]. The boat is traveling as shown in Figure 3.1.

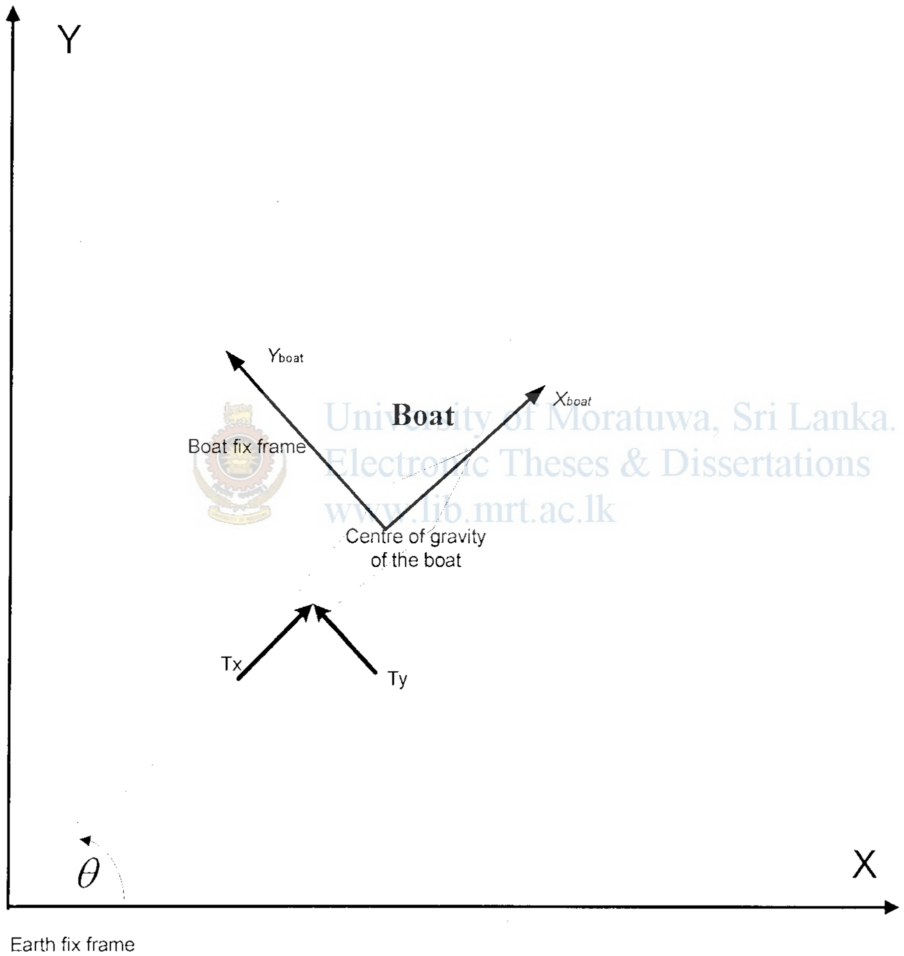

Figure 3.1 - Boat with fuzzy based navigational controller

Tx and *Ty* are the thrust forces exerted by the propellers of the boat. The heading of the boat is given by θ. The thrust force is controlled using a fuzzy controller. The overall control system (navigation) is shown in Figure 3.2. If the required position of the boat is (x_r, y_r) and the actual position is (x,y) then the error e is given by $e=x_r-x$.

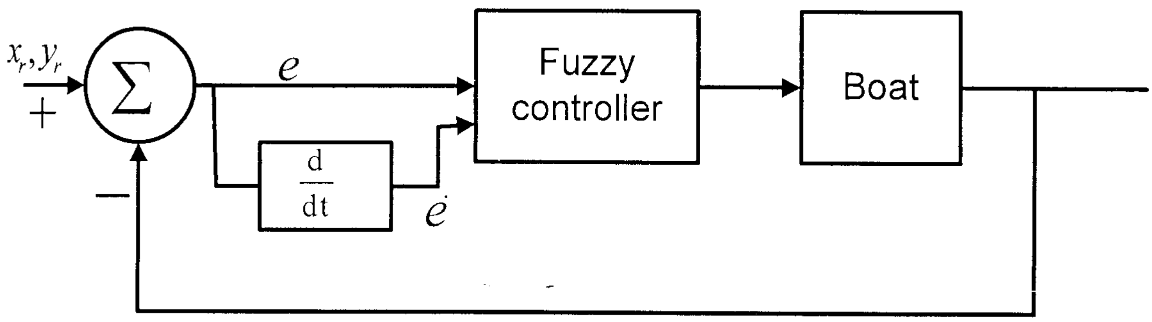

Figure 3.2 - Fuzzy based navigational controller

Five input membership functions were defined to represent the error input named negative large (NL), negative small (NS), zero (Z), positive small (PS), positive large (PL) and same names were used for the change rate of error as well. Input membership functions to fuzzy controller and output membership functions from the fuzzy controller are given in Figure 3.3, 3.4 and 3.5.

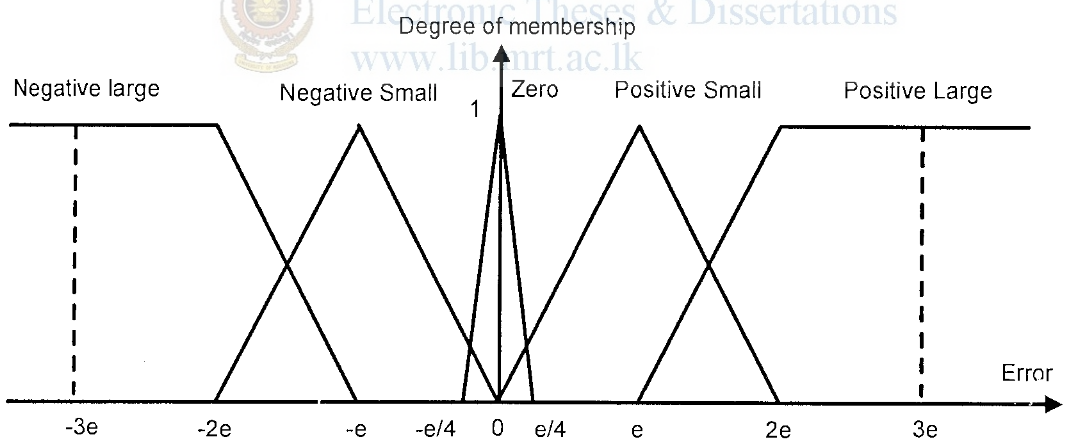

Figure 3.3 - Error input membership function of the fuzzy based navigational controller

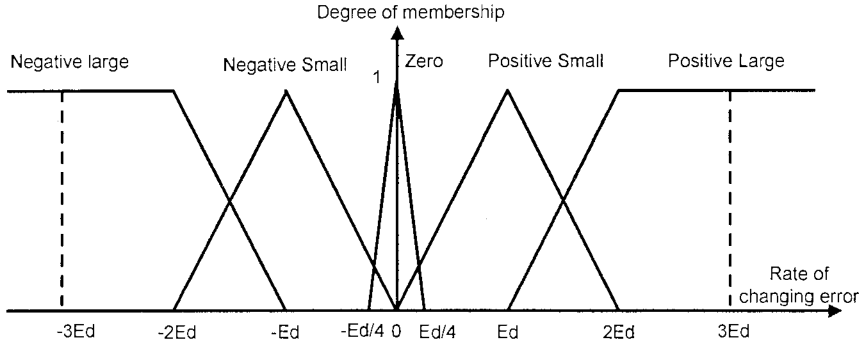

Figure 3.4 - Rate of change error input membership function of the fuzzy based navigational controller

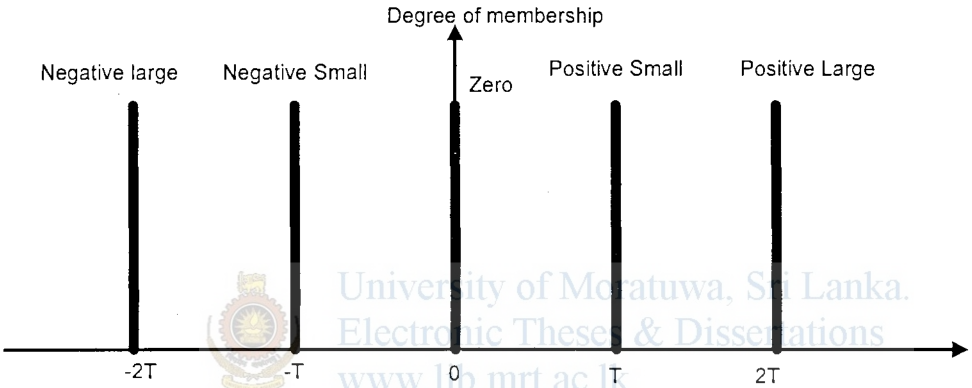

Figure 3.5 - Rate of change error input membership function of the fuzzy based navigational controller

Rule base is one of the important parts in the fuzzy controller. When Table 3.1 shows that the "thrust is positive large" means that the fuzzy controller will give controlling signals to the boat's thrust force controller to increase its thrust force by *2dT* (maximum safe thrust increase) by changing propeller angle and torque. "Thrust is negative large" is represented the maximum thrust force reduction of the boat. The Figure 3.6 presents the output surface of the fuzzy-logic navigational controller.

Table 3.1 - Rule table for fuzzy PD controller

e \ de/dt	NL	NS	Z	PS	PL
NL	NL	NL	NS	NS	NS
NS	NL	NS	NS	NS	NS
Z	NS	NS	Z	PS	PS
PS	PS	PS	PS	PS	PL
PL	PS	PS	PS	PL	PL

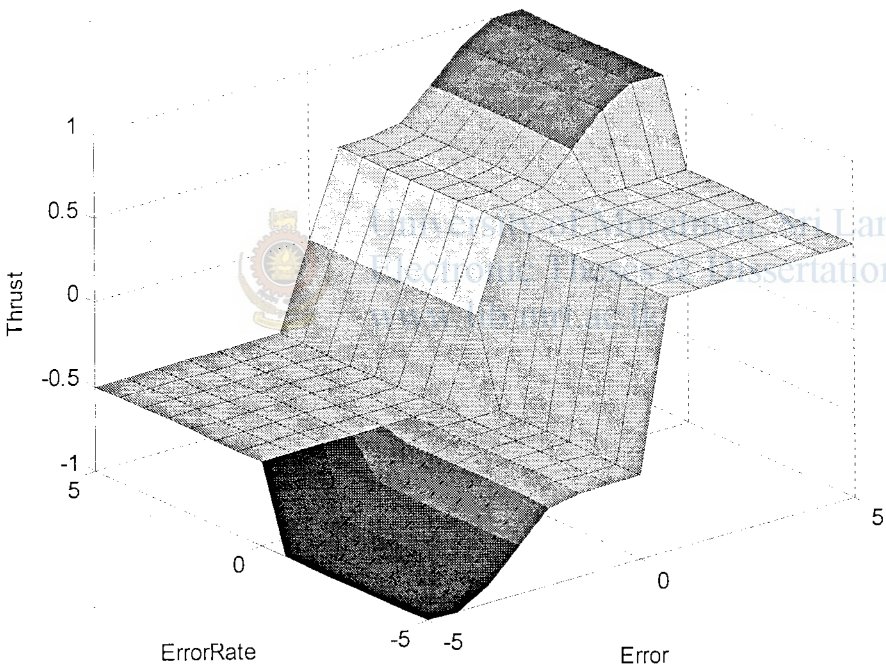

Figure 3.6 - Output surface of the fuzzy based navigational controller

3.2 Mathematical model for USV

The mathematical model of the boat called Delfim (Figure 3.7), developed by Dynamical Systems and Ocean Robotics Laboratory (Portugal) [37] is utilized for the simulation purposes. The following dynamic equations are modeled in MatLab environment [19]. This mathematical model is controlled by using the fuzzy logic navigational controller in the simulations which are performed

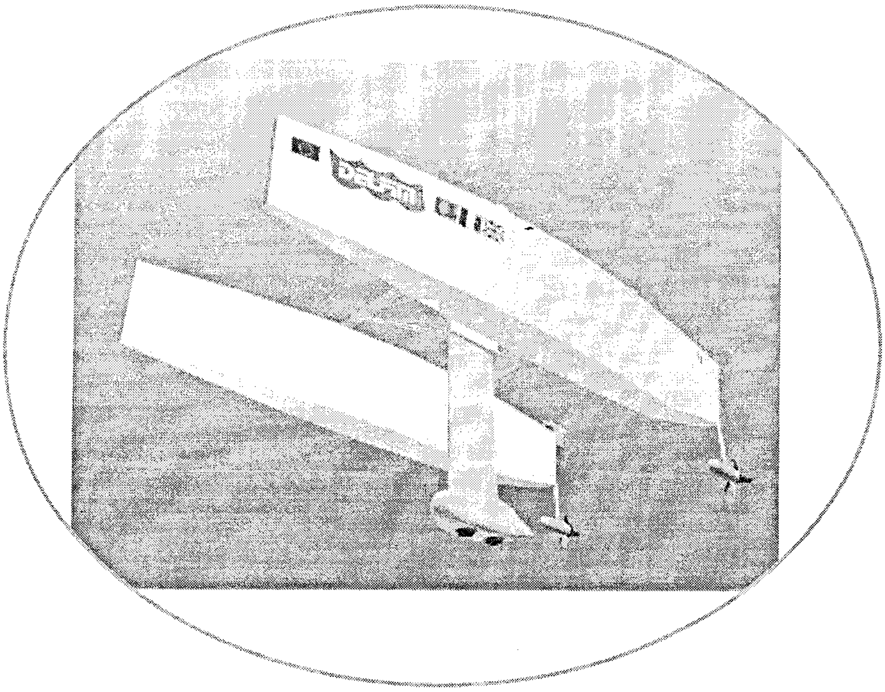

Figure 3.7 - Picture of the actual boat model

The velocities of surge (XB-direction), sway (YB-direction), and yaw (rotation about ZB-direction) are defined as $u = u(t)$, $v = v(t)$, and $r = r(t)$. Then the dynamic equations for the model are given as below,

$$m\dot{u} = -D_X(u,v,r) + mvr + T_P\cos\alpha_p + T_S\cos\alpha_s \quad \text{---------------(3.1)}$$

$$m\dot{v} = -D_Y(u,v,r) - mur + T_P\sin\alpha_p + T_S\sin\alpha_S \quad \text{---------------(3.2)}$$

$$m\dot{r} = -D_\varepsilon(u,v,r) + T_p\cos\alpha_p \times d_{py} + T_S\cos\alpha_s \times d_{sy} + T_P\sin\alpha_p \times d_x + T_S\sin\alpha_S \times d_x. \quad \text{----(3.3)}$$

Where T is thrust delivered by port side and starboard side propellers, respectively the α_P and α_S are inclinations to the $X_{B\text{-}axis}$, d is diameter of the propeller.

Basic steps of the MatLab program are given below,

a) Define all the variables and set the dimensions of the boat
b) Calculate mass and inertia matrix of the boat
c) Calculate frictional, form and additional resistance forces

e) Calculate the damping and total moments

f) Set all the variables to its initial values, including time *t*, *t* = *0*

g) Define boundary conditions for all variables of input and output

h) Initialize the reference values, such as reference trajectory of the ship in 2-D plane

i) While $t < t_{stop}$

Calculate the *position error*

Feed the inputs to the controller (within equal intervals)

Process the rules according to the inputs, by means of MatLab FIS.

Get the defuzzified output

Solve dynamic equations to find new position

Plot the results

3.3 Results from the navigational controller

The performance of the fuzzy based navigational controller is compared with the performance of a PD controller to prove the brilliance of the fuzzy logic based controllers. The results are presented in Figure 3.8 to 3.12 and MatLab progam is given in Appendix B. The desired path is presented in red while blue is utilized for the actual path.

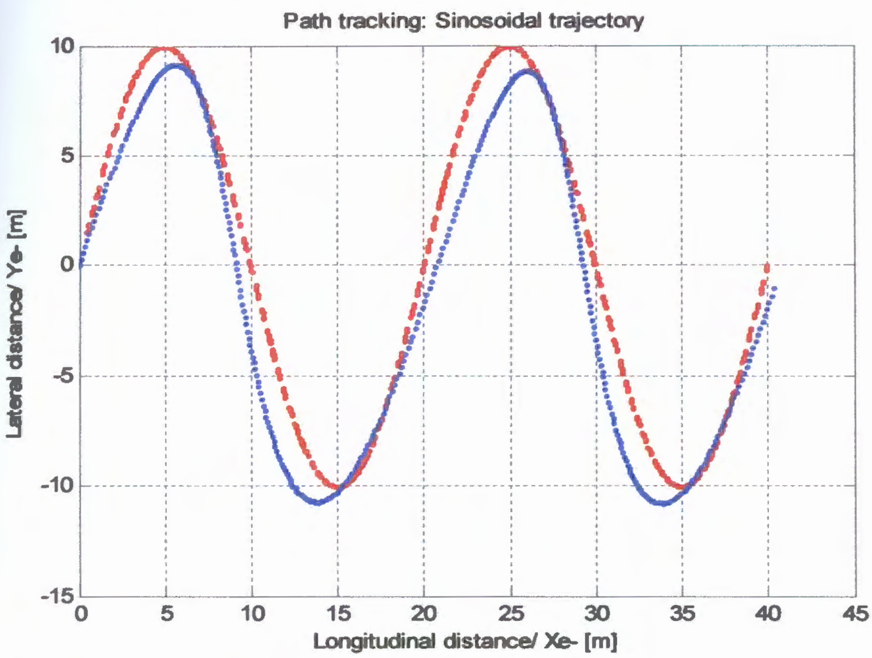

Figure 3.8 - Path tracking of a Sinusoidal trajectory with PD controller

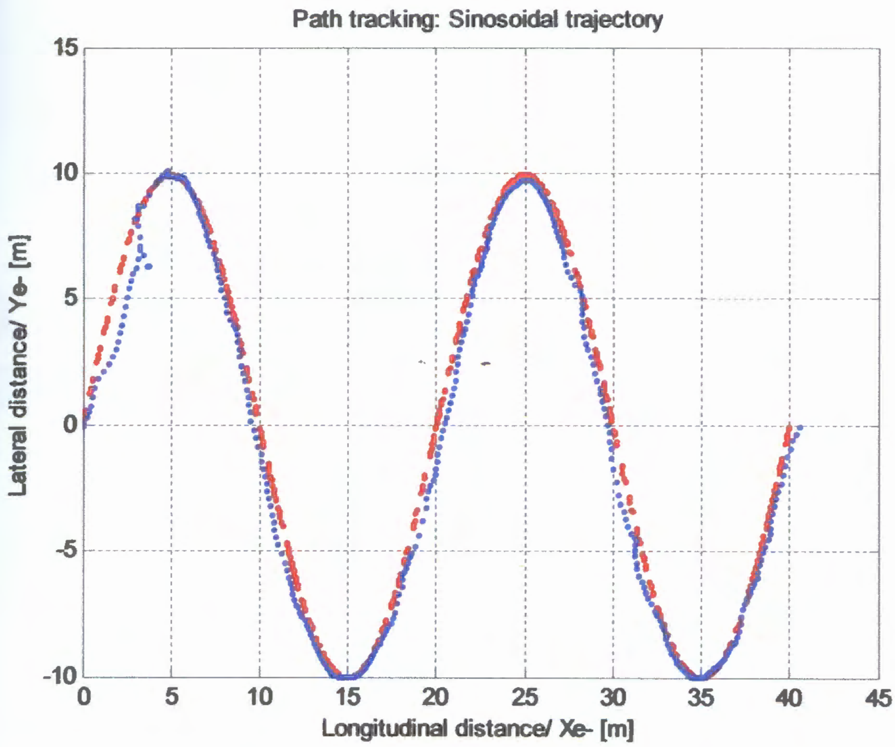

Figure 3.9 - Path tracking of a Sinusoidal trajectory with Fuzzy controller

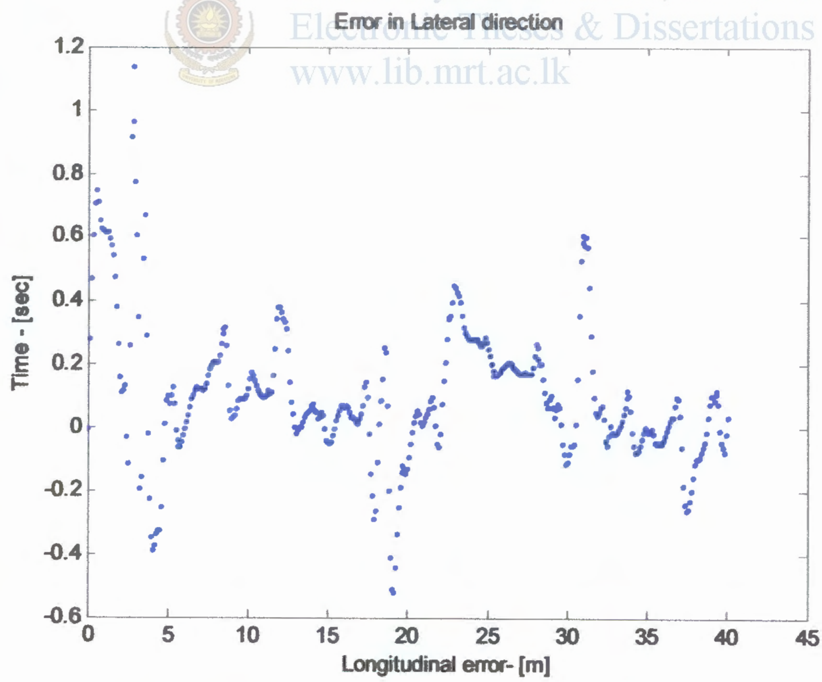

Figure 3.10 - Error in Lateral direction of a Sinusoidal trajectory with Fuzzy Controller

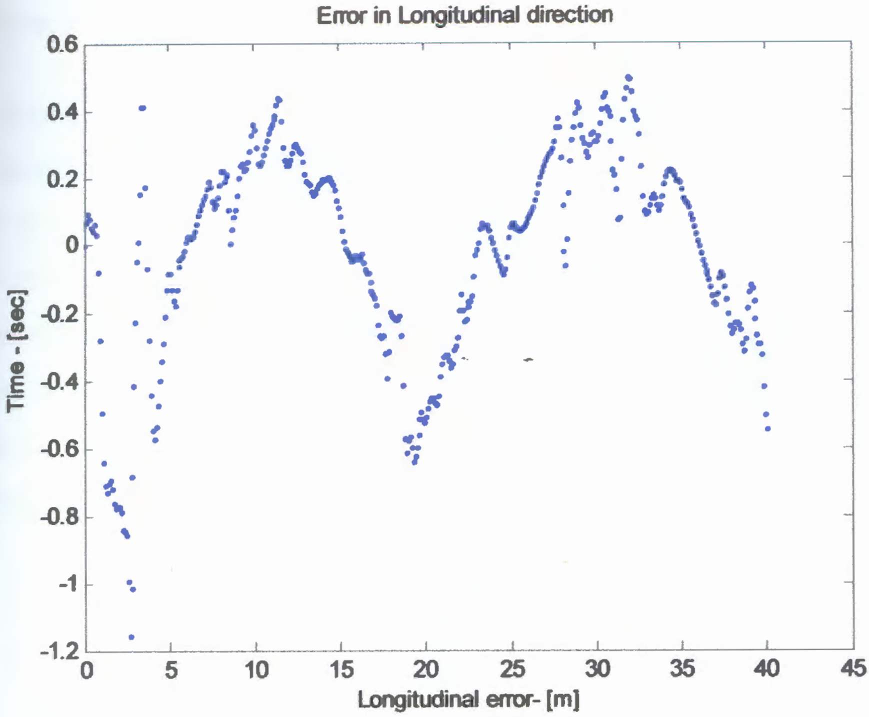

Figure 3.11 - Error in Longitudinal direction of a Sinusoidal trajectory with Fuzzy controller

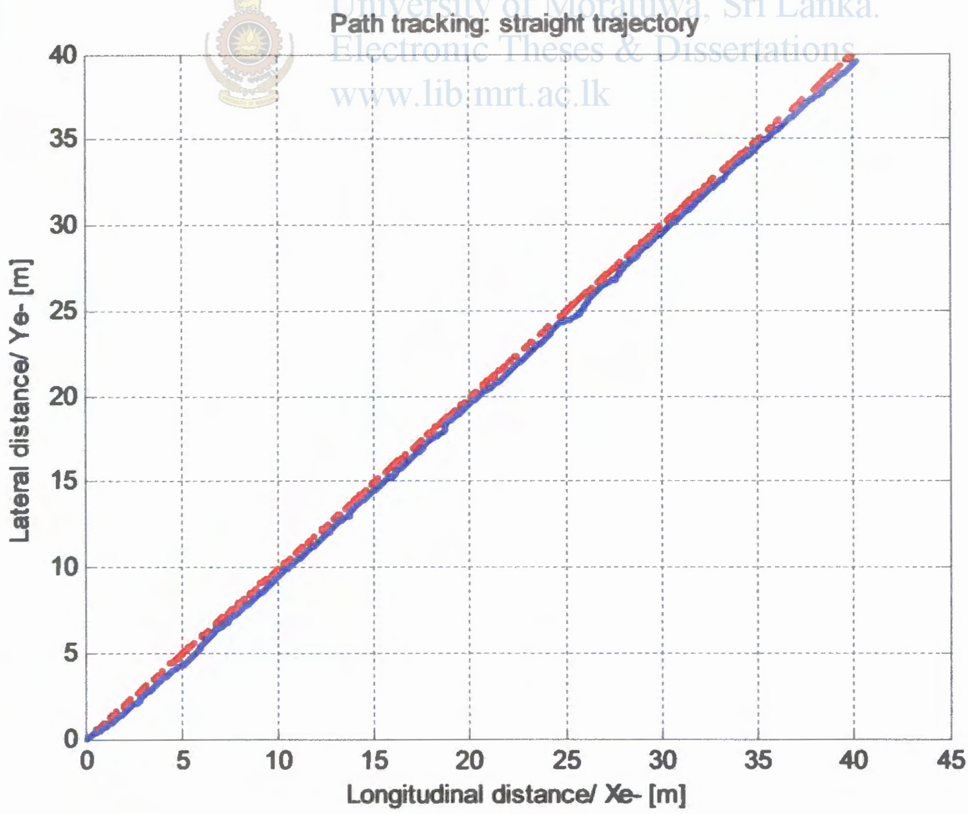

Figure 3.12 - Path tracking of a Straight trajectory with Fuzzy controller

3.4 Summary

This chapter mainly described about designing and simulating a fuzzy logic-based navigational controller for unmanned surface vehicles. An already developed dynamic model of a boat is utilized for the simulations though out the report. The controller considered in this study is a fuzzy based system. MatLab framework with Fuzzy-logic toolbox was used to design and implementing the whole system. MatLab programs were employed for the navigational controller simulations. The basic design procedure and the simulation procedure were included in detail. The results including the desired and actual paths are plotted at the end.