2x20x10x1 to compare the tracking performance using a fixed network size. This may not be the optimum size for a particular controller, but is found to give uniformly good performance in all the cases. Hence to consider the computational aspects for real time implementation, each network has to be fine tuned individually. For example, in the case of yaw angle controller, the same tracking performance level is found to be achieved by two neural networks of different sizes. The first network is of size 2x20x10x1 and the second one is of 2x5x1. Even though the number of interconnections are reduced from 281 to 21 (a reduction of 1300%), the network performance is not degraded noticeably. The smaller network is expected to have poorer reliability and fault tolerance capabilities. Considering that each interconnection in the network corresponds to two multiplications when BP algorithm is used (one each for forward and backward pass), the network of size 2x20x10x1 needs approximately 562 multiplications and 250 additions. Since the network of size 2x5x1 needs only 42 multiplications and 15 additions, there is considerable savings in computation.

is also shown that the gain layer scheme provides an effective way to approach MIMO control, by splitting it into SISO cases. The robustness of the control scheme to adapt to slow and sudden variations in the dynamics was also studied. Following are the observations made during the studies on the proposed control scheme:

1. The learning rate is a critical factor for the overall stability of the system. It is the single most important parameter of the proposed neural network controller. The rate of learning could be adjusted to give the desired dynamic responses (such as over shoot and rise time)
2. The learning rate is to be determined on a trial and error basis, depending up on the dynamics. In the case of SISO simulations, each controller is found to have different optimum learning rates.
3. The initial weight distribution of the network influences the convergence. It is neces–sary to keep the initial weights to small[2] values (random values between –0.1 and +0.1 in our studies) because of the stability considerations.
4. No improvement in tracking performance of the system is observed when the neural network is presented with the error $(y_d - y)$ [see figure 5.2] instead of both y_d and y.[3] This indicates that the error information is computed inside the network itself.
5. The design complexity of a neural network controller for a linear model of the AUV dynamics is found to be same as that for the nonlinear one.
6. In all the simulations described above, the network size is deliberately chosen as

---

[2] A small learning rate generates only small fluctuations in the weight space and hence large initial weights cause slower convergence. The low learning rate becomes necessary from the stability considerations when using static backpropagation networks as learning controllers[51]..

[3] For a real–time implementation, this may be a better network architecture as the number of interconnect–ions are reduced. But the other performance characteristics of the controller will have to be studied before choosing this architecture.

The controller scheme of figure 5.22 is also employed for the simultaneous control of vertical velocity and pitch of the AUV dynamics. The results are shown in figure 5.24. Figure 5.24a shows the tracking of the pitch command signal, and figure 5.24b shows the tracking of the vertical velocity. One important point to be noticed here is that arbitrary command signals could not be given to the system for tracking as the relationships are highly coupled. The individual commands should be valid in the operating regime of the vehicle.

Also it may be noted that the individual learning rates for the controllers are not the same. A learning rate of 0.05 is used for the pitch command control while the learning rate used for the vertical velocity is 0.1.

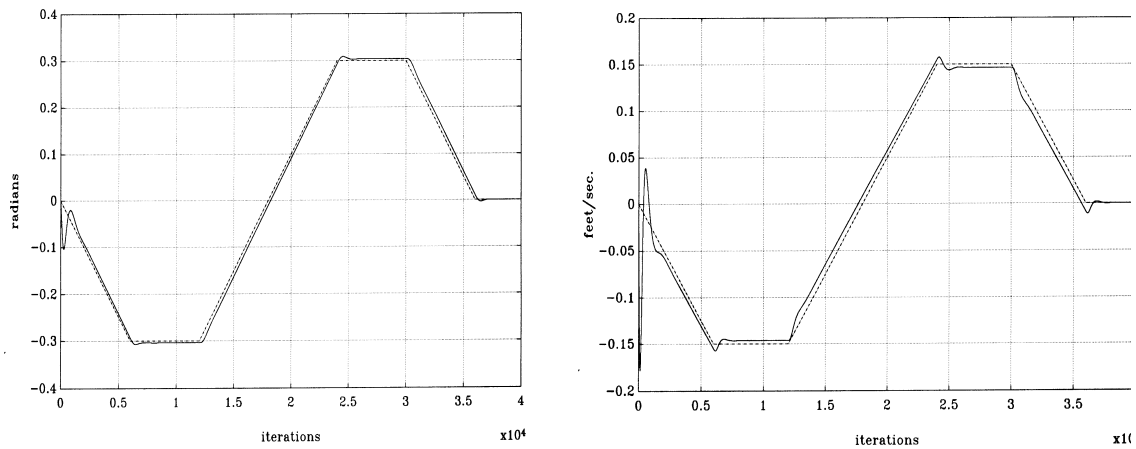


Figure 4.24 Simultaneous control of pitch (θ) and vertical velocity (w), using the scheme in Figure 5.2. (a) Tracking of the pitch command (b) Tracking of the vertical velocity command

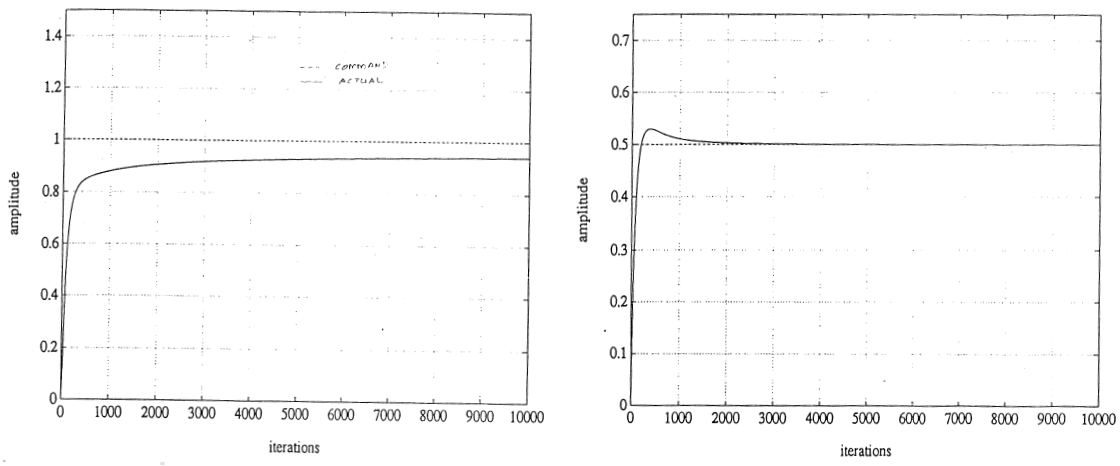
2.5. SUMMARY

In this chapter more detailed studies on the gain layer scheme are reported. Studies were conducted on both linearized dynamics as well as nonlinear dynamics of an autonomous underwater vehicle. It is verified that the proposed scheme gives good tracking performance. It

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \end{bmatrix} u \qquad (5.5)$$

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 2 & 1 \end{bmatrix} u$$

The results for step command inputs for the above system are shown in figure 5.23. With networks of size 5x20x10x1, the weights are adapted on–line to minimize the corresponding output errors $e_1(k)$ and $e_2(k)$, where $e_1(k) = y_{d_1}(k) - y_1(k)$ and $e_2(k) = y_{d_2}(k) - y_2(k)$. A learning rate of 0.002 is used for network 1 and 0.005 for the network 2. It can be seen that the system is able to track the command signal well. Note that the learning rate used for each of the networks is different.



(a) (b)

Figure 4.23 Control of the MIMO system. Tracking of step command inputs

(a) Output 1
(b) Output 2

2.4.2. Direct Control Scheme for MIMO Systems

In Section 5.2, a direct control scheme for MIMO systems was discussed. There the MIMO system was split into SISO systems and were trained individually to minimize the corresponding system error. The scheme can be modified by providing delayed states of the control inputs as additional inputs to the neural network. Figure 5.22. shows such a scheme for a 2–input 2–output MIMO system. Note that with feedforward controller the number of delayed states fedback to the network depends upon the order of the system. For schemes with recurrent networks as controllers, only one previous state of the signals is needed, at least theoretically.

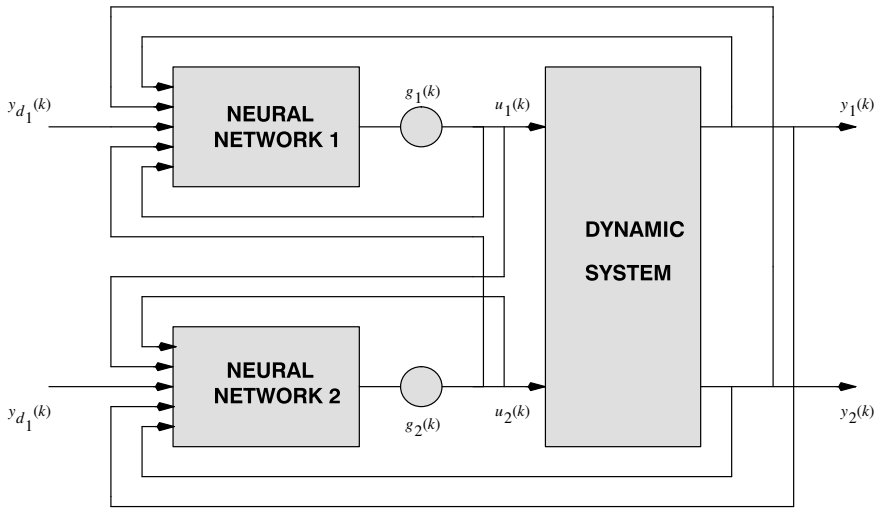


Figure 4.22 Direct Control Scheme for a 2−Input 2−Output MIMO System using gain layer

Simulation studies using gain layer scheme were conducted to investigate the approach shown in figure 5.22. on a controllable and observable linear MIMO system described by

2.4. GAIN LAYER SCHEME FOR THE LEARNING CONTROL OF MIMO SYSTEMS

2.4.1. Indirect Control Scheme for MIMO Systems

In Chapter 2, the use of series parallel schemes for identifying SISO nonlinear dynamical systems was considered. Here the same approach is extended to MIMO systems. As an example, consider a 2–input 2–output system described by;

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} f_1[y_1(k), y_2(k)] \\ f_2[y_1(k), y_2(k)] \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_1(k) \end{bmatrix} \tag{5.1}$$

The nonlinear functions $f_1(.)$ and $f_2(.)$ can be mapped on to two separate neural networks N_1 and N_2. If we know the desired outputs of the system $y_{d_1}(k+1)$ and $y_{d_2}(k+1)$, the control signals can be generated according to the indirect control scheme, as given below.

$$\begin{bmatrix} u_1(k) \\ u_1(k) \end{bmatrix} = \begin{bmatrix} y_{d_1}(k+1) \\ y_{d_2}(k+1) \end{bmatrix} - \begin{bmatrix} f_1[y_1(k), y_2(k)] \\ f_2[y_1(k), y_2(k)] \end{bmatrix} \tag{5.2}$$

It was pointed out before, in Chapter 2, that this approach may not be easy and practical in situations where the nonlinear functions f_1 and f_2 involve terms in $u_1(k)$ and $u_2(k)$. Equations (5.3) and (5.4) provide examples for such systems.

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} f_1[y_1(k), y_2(k), u_1(k), u_2(k)] \\ f_2[y_1(k), y_2(k), u_1(k), u_2(k)] \end{bmatrix} \tag{5.3}$$

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} f_1[y_1(k), y_2(k)] \\ f_2[y_1(k), y_2(k)] \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_1(k) \end{bmatrix} \tag{5.4}$$

and back to a depth of 0 feet in 1000 seconds. and then back to zero. The vehicle is able to follow the command signal effectively. The corresponding control input generated by the neural network is shown in figure 5.21b. A steady input of about 2 degrees is generated by the network during the initial period of 1000 seconds to reach the vehicle to a depth of 200 feet. To keep the vehicle steady at that height, the rudder deflection is switched to –3 degrees. and then is maintained at 0.5 degrees. until the command signal is changed to reach a depth of 0 feet. The network size and learning rate are same as those for the previous case. A modification to the above scheme may be to feedback the forward speed signal to the depth controller and use another neural network for the control of the forward speed.

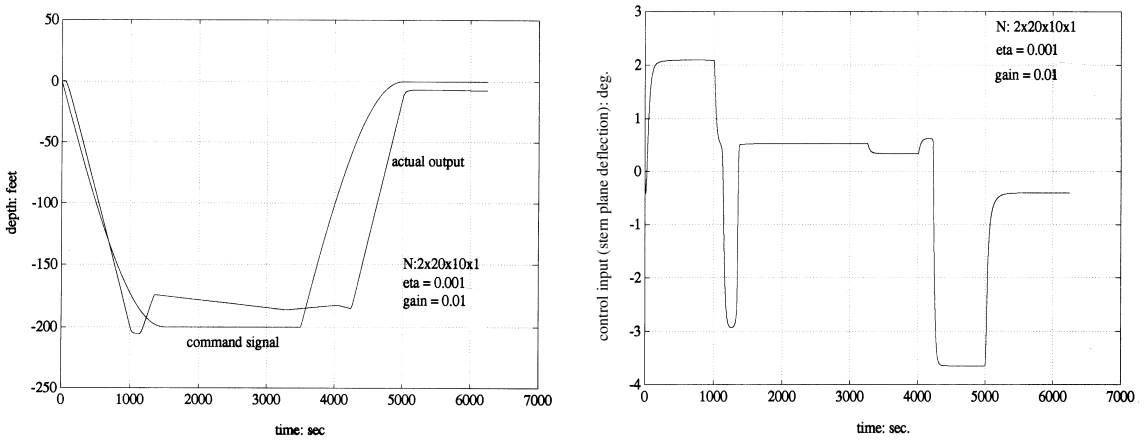


Figure 4.21 Depth Control for a time varying command signal

(a) Command Signal and the corresponding Vehicle Response

(b) Control input generated by the neural network controller

2.3.4.3. Depth Control:

The response of the vehicle to a depth command signal which is meant to direct the vehicle to a depth of –200 feet in 1000 sec. and remain steady at that depth, is shown in figure 5.20. Again a sudden transition in command signal is found to create large oscillations in the depth because of the vehicle inertia and hence the command signal is varied in a linear fashion initially. As pointed out earlier, such sudden transitions in command signals are not practical. In our case, the forward thruster rpm is initially set to 550 and when the depth is brought to –200 feet, the rpm is decreased to 225. This reduction in rpm is a necessary operational requirement whenever there is a change in the depth command signal, because of the constraints due to the vehicle inertia. A learning rate of 0.001 is used for the neural network controller.

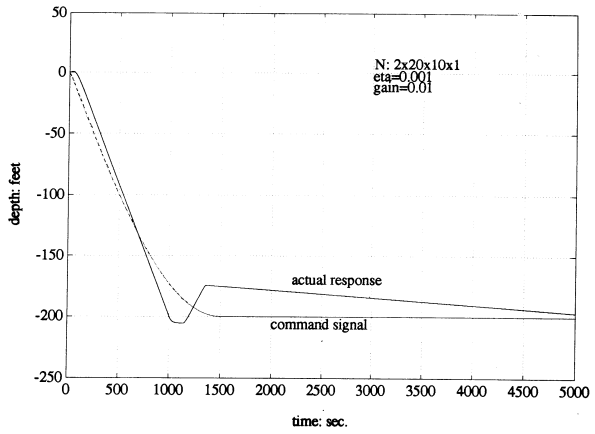
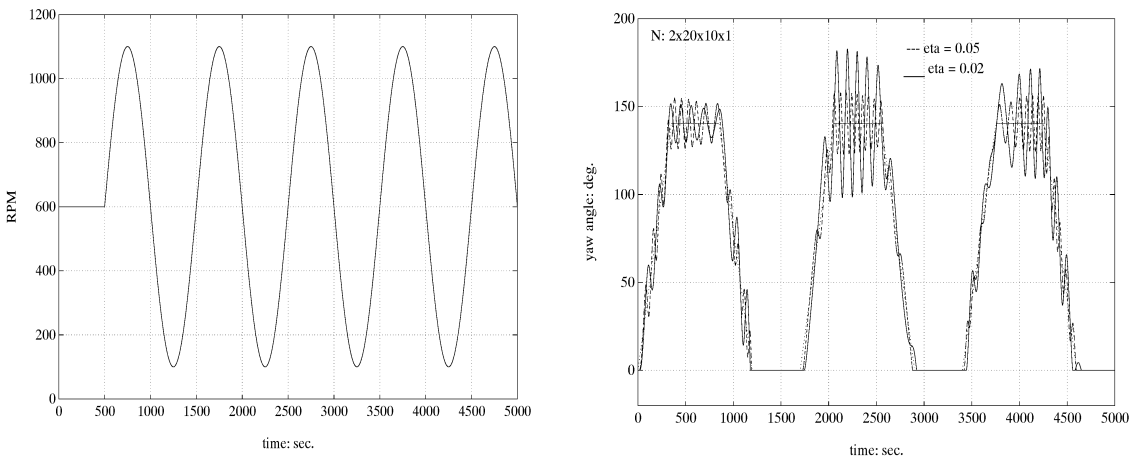


Figure 4.20 Tracking of a constant depth command. Learning rate for the neural network is 0.001

Figure 5.21a shows the tracking of the depth signal when the vehicle is commanded to a depth of –200 feet in 1000 seconds and remain at a steady depth of –200 feet for 2500 seconds

namics is evident from the decrease in the amplitude of oscillations as time proceeds. The corresponding control signal provided by the neural network is shown in figure 5.18b. The learning rate for the controller is 0.05 with the same forward speed as in the previous case.

The robustness of the neural network controller is investigated by disturbing the vehicle parameters. The forward thruster rpm is changed between 100 and 1100 (corresponding to a speed variation between 0.84 ft/sec. & 9.3 ft/sec.) in a sinusoidal fashion (figure 5.19a) and the tracking of the command signal is observed. Figure 5.19b shows the results for two learning rates, 0.05 and 0.02. The solid line represents a learning rate of 0.02 and the dashed one is for a learning rate 0.05. As observed in the case of pitch control (figure 5.15), a higher learning rate helps in adapting to changes in dynamics faster (even though the initial overshoot is higher) than a lower learning rate. The controller is found to adapt well when dealing with sudden disturbances also.



(a) (b)

Figure 4.19 Tracking of the command signal when a slow varying disturbance is injected into the dynamics.
(a) Variation in the forward thruster rpm
(b) Response of the vehicle (heading) for two different learning rates

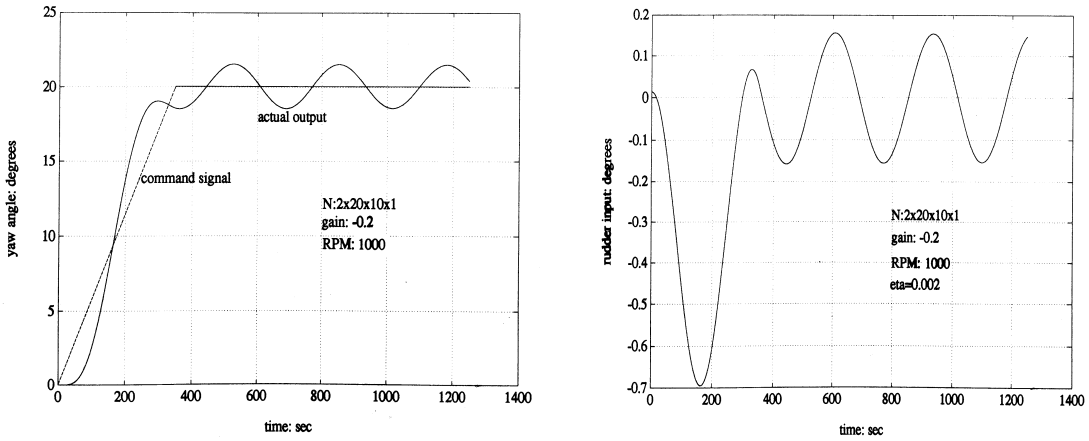


Figure 4.17 Controlling the heading of the AUV using the gain layer scheme
(a) Response of the vehicle (heading) for a constant command of 20 degrees

(b) The control input to the rudder generated by the neural network for the response in (a)

The vehicle heading in the case of a periodic trapezoidal command signal is shown in figure 5.18a. The neural network is able to control the heading effectively even when the heading command was varied between 0 and 140 degrees. The ability of the network to learn the dy–

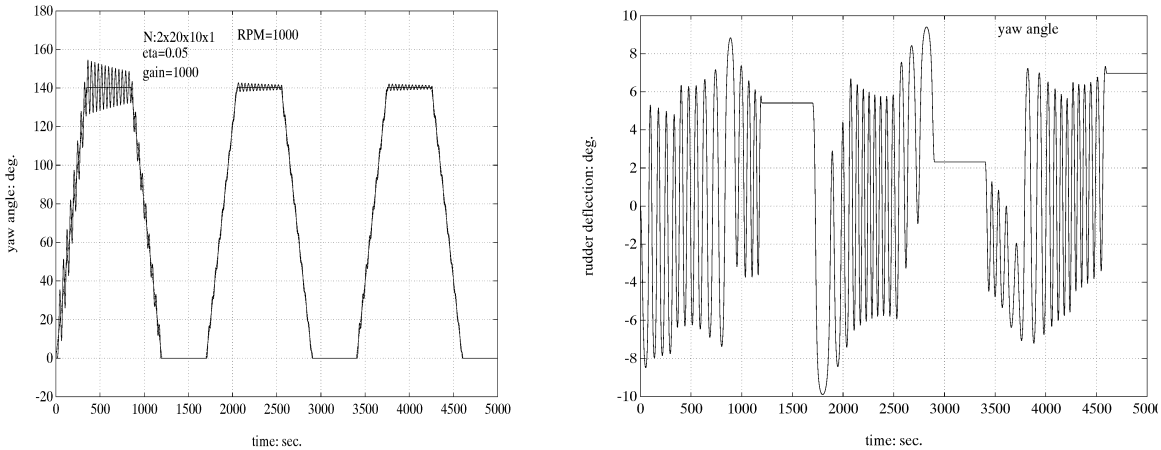


Figure 4.18 Heading Control for oscillatory command signal
(a) Vehicle Response (b) Corresponding control input

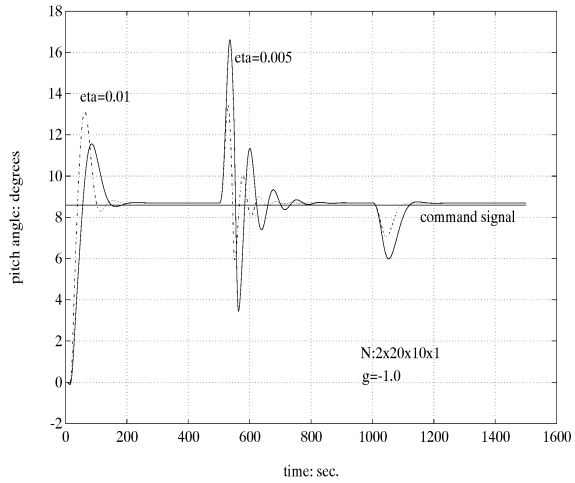


Figure 4.16 Tracking of the pitch command under sudden disturbance. The forward thruster rpm is changed from 400 to 1200 at t=500 sec. and back to 400 at t=1000 sec.

2.3.4.2. Heading Control

Using the same control strategy, a neural network controller is designed to command the vehicle for tracking a specified heading (yaw). Figure 5.17a shows the response of the vehicle for a constant heading command. Because of the inertia of the vehicle, an ideal step input is found to produce a large overshoot, hence a trapezoidal command signal is used. The error in tracking in this case is within an acceptable $\overline{+}2$ degrees. The corresponding control input to the rudder given by the neural network controller is shown in figure 5.17b The network learning rate is 0.002 and the forward thruster rpm is 1000 (8.44 ft/sec.). A higher learning rate produces larger oscillations but the error diminishes faster. The total error is found to be within +/– 2 degrees even when the commanded heading is 140 degrees (instead of 20 degrees as in figure 5.17a, with the same learning rate).

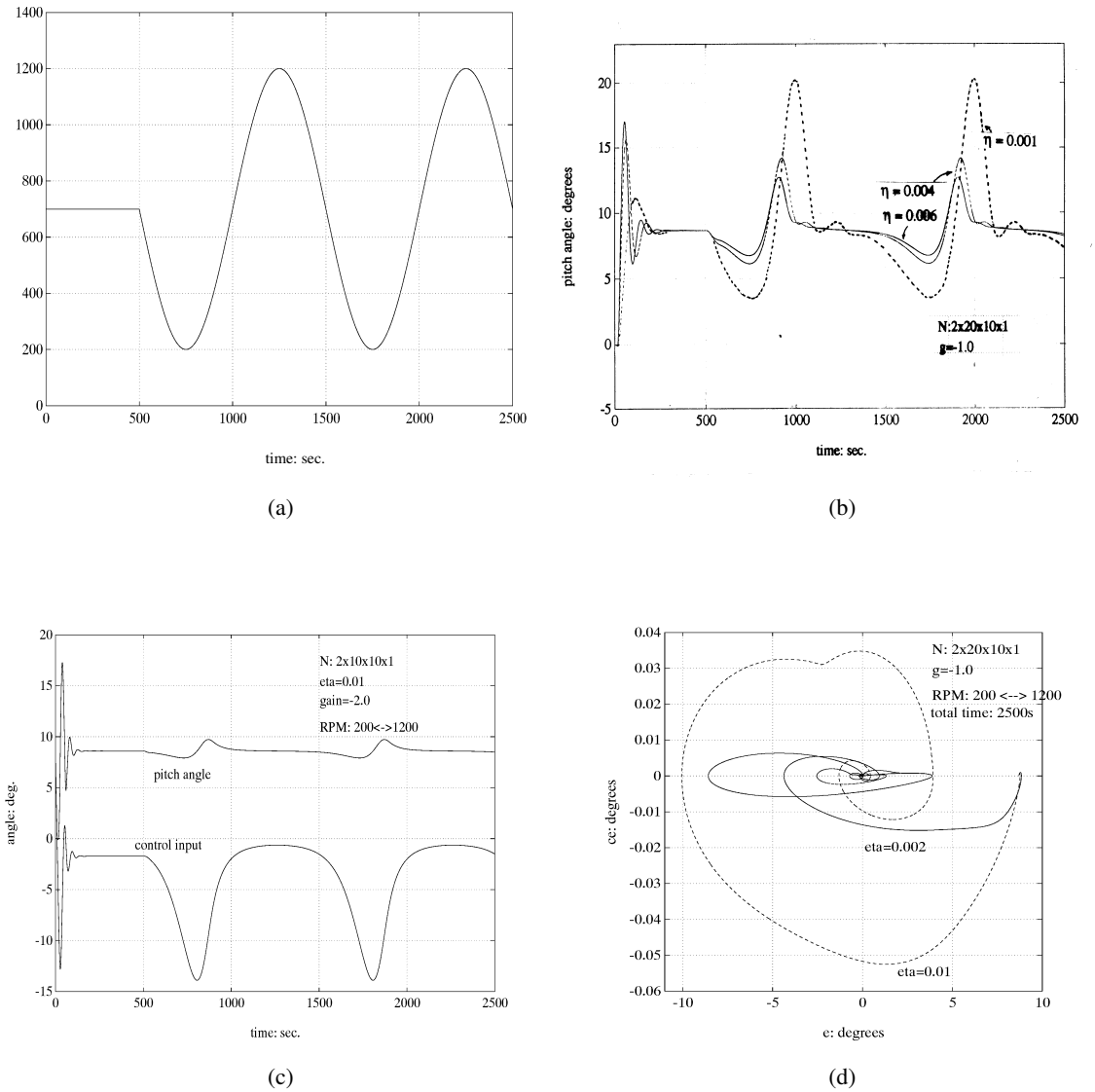


Figure 4.15 Effect of learning rate on the adaptation of the controller under disturbance

(a) Slow varying disturbance on the dynamics: change in forward thruster rpm.

(b) Vehicle response for three different learning rates

(c) Vehicle response and corresponding control signal generated by the neural network

(d) State space representation of the error convergence

of about 25 sec., while an η of 0.002 produces no overshoot but the rise–time is about 200 sec. A learning rate of 0.005 may be a good choice in this case. A state space plot for the step command (error 'e' versus error rate 'ce') shown in figure 5.14 for three different learning rates, also verifies that the dynamic response of the system can be controlled accurately by adjusting the learning rate of the neural network.

The effectiveness of the neural network controller to adapt to slowly varying changes in the vehicle parameters is investigated by perturbing the dynamics. The forward speed is one of the important parameters affecting the vehicle characteristics. The robustness of the controller is studied by changing the vehicle speed between 1.7 ft/sec. and 10.13 ft/sec., and by varying the thruster rpm between 200 and 1200 in a sinusoidal fashion, as shown in Figure 5.15a. Figure 5.15b shows the corresponding tracking of the step pitch angle command signal. It can be seen that a higher learning rate produces larger initial overshoot, but it makes the controller easily adaptable to changes in the dynamics. When the learning rate is decreased to 0.005 and further to 0.001, the initial overshoot does decrease but the controller's ability to adapt to disturbances reduces considerably.

Figure 5.15c shows the control input and the corresponding pitch angle for a case with η=0.01. A compromise between adaptivity, overshoot and rise–time is needed to select a proper learning rate for the controller. The state space plot shown in figure 5.15d illustrates the effect of higher learning rate in faster adaptation of the network. Figure 5.16 shows the change in the vehicle pitch when the forward thruster speed is suddenly increased by 300%, from 400 rpm to 1200 rpm at t=500 sec. and back to 400 rpm at t=1500 sec. The controller is able to adapt well even to this fast varying disturbance signal.

is shown in figure 5.12. The learning rate for the network controller is 0.005 and the forward speed of the vehicle is set constant by keeping the thruster rpm to 800, which corresponds to a speed of 6.8 ft./sec. The tracking control of a rectangular pitch command signal is shown in figure 5.13. It is to be noted again that in practical situations the command signals will not be having sudden variations, hence the studies presented here deal with the worst case. It can be seen that even though the controller is started from a random initial state (with no prior training), it is able to produce the correct control action in a few learning iterations. The large initial overshoot is attributable to the high inertia of the vehicle and can be controlled by using an appropriate learning rate for the neural network. The shown results correspond to the initial period of operation. As time proceeds, the overshoot and the rise–time are observed to be improving.

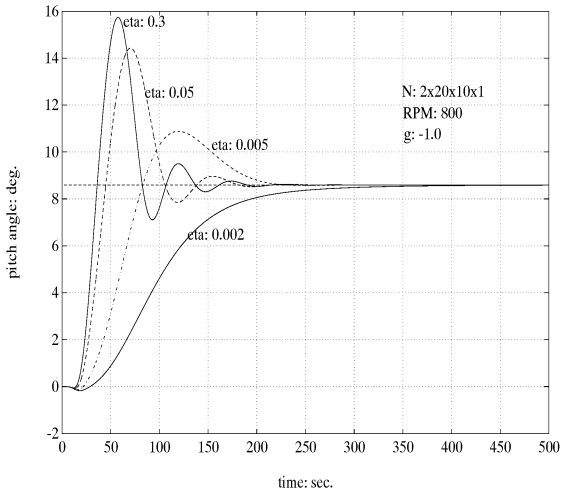


Figure 4.13 Effect of the controller learning rate on the system response (pitch).

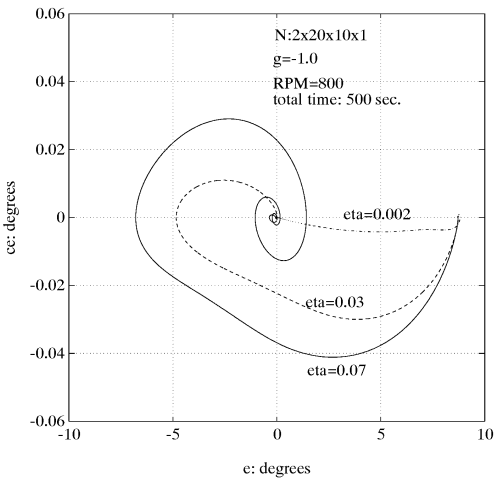


Figure 4.14 State space representation of the error convergence for the step pitch command

Figure 5.13 shows the control of the pitch angle for a step command input, for different learning rates(η). It can be seen that the system rise time and overshoot depends upon the learning rate used for the controller. An η of 0.3 results in an overshoot of about 90% with a rise time

gence times to the get to the solution and slower learning. The effect of the initial weight distribution was studied by considering the heading command signal shown in figure 5.10. A network with smaller initial weight distribution (between –0.1 and +0.1) is able to track the command faster and closer. The learning rate and initial weights both influence the overall system time constant and determine the dynamic responses.

2.3.4. Nonlinear Dynamics

The studies using the gain layer schemes are repeated on the nonlinear model of the vehicle dynamics. As in the previous experiments, controllers are designed for vehicle parameters pitch, depth and yaw[53][65][66].

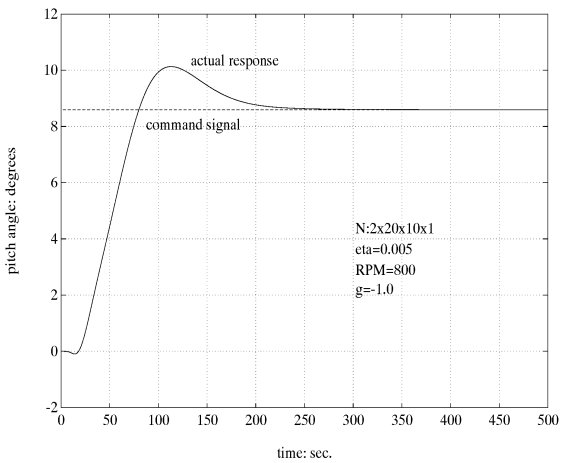


Figure 4.11 Response of the vehicle to step pitch command

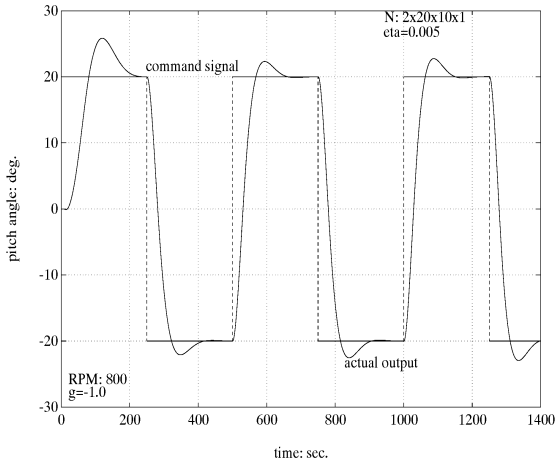


Figure 4.12 Response of the vehicle (pitch) for a rectangular command signal. Learning rate for the controller is 0.008.

2.3.4.1. Pitch Control

In this case, the vehicle is commanded to move at a specified pitch. The stern plane deflection is the control input to the dynamics. The response of the vehicle for a step pitch command

2.3.2. Effect of the Learning Rate:

The effect of the learning rate on the vehicle response is studied from the side velocity relationship. Figure 5.9 shows the control of the *v* for a rectangular command input, for different learning rates(η) for the controller. It can be seen that the system rise time and overshoot depends upon the learning rate used for the controller. An η of 0.5 results in an overshoot of about 50% with a rise time of about 7 sec.(one iteration corresponds to 0.1 sec.), while an η of 0.05 produces no overshoot but the rise–time is about 300 sec. A learning rate of 0.2 may be a good compromise in this case. The learning rate corresponds to the time constant of the system and the dynamical response of the system can be controlled accurately by adjusting the learning rate of the neural network.

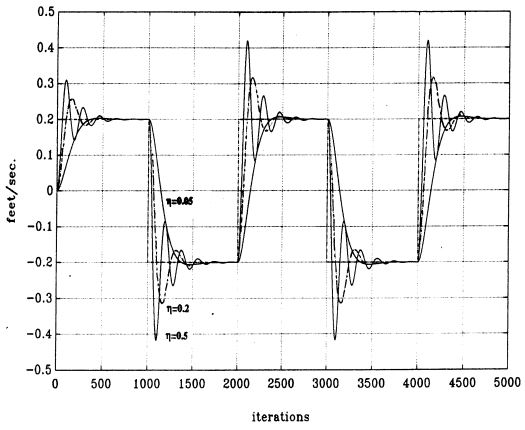


Figure 4.9 Effect of the learning rate on the tracking performance of the system

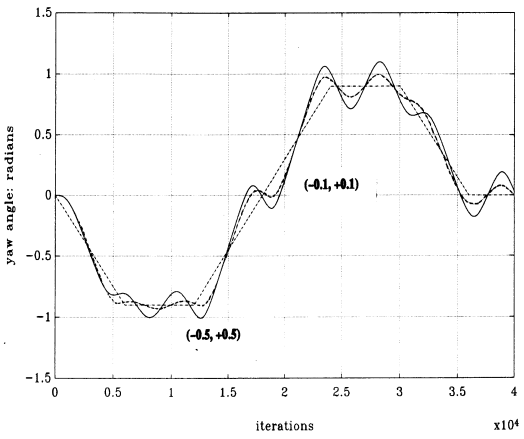


Figure 4.10 Effect of initial weight distribution on the tracking performance of the system.

2.3.3. Effect of Initial Weight Distribution:

A relationship which is sensitive to the transition in command signal will be sensitive to the distribution of weights also. This follows as a wider distribution will result in longer conver-

situations, the requirement may be just to steadily keep the roll angle at an angle zero rather than commanding the vehicle to be at different angles. The sway of the vehicle is determined by the side velocity and the response of the vehicle towards a rectangular command signal is shown in figure 5.8. The vehicle is commanded to move to the side at a rate of +0.2 feet/sec. for 100 sec. and at rate –0.2 feet/sec. for another 100 sec. It can be seen that with a learning rate of 0.2, the controller is able to track the command well.

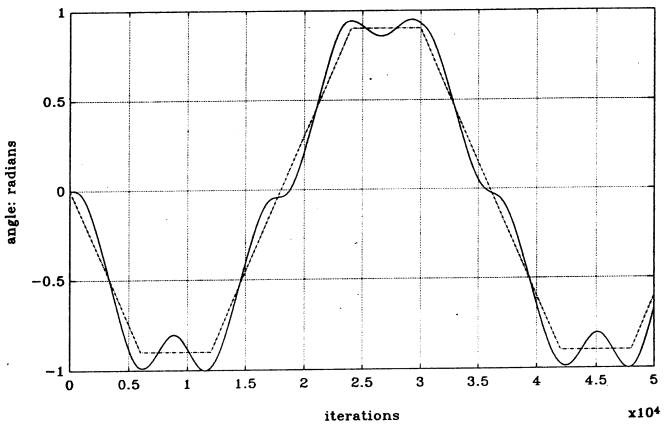


Figure 4.6 Heading control of the vehicle. The learning rate for the neural network is 0.02

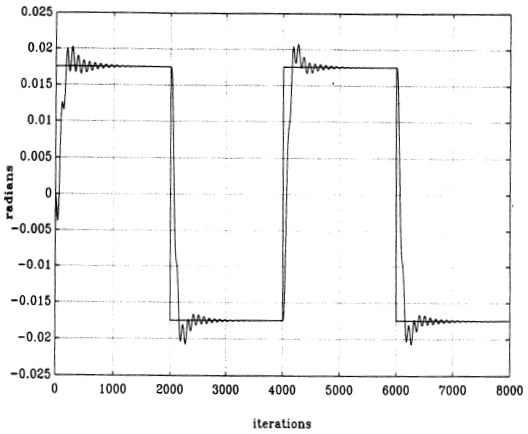


Figure 4.7 Tracking of the roll angle command signal. The learning rate used is 0.5

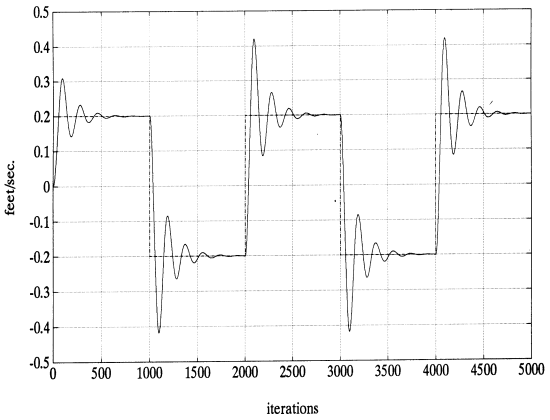


Figure 4.8 Tracking of the side velocity command input with a neural network learning rate of 0.2.

Once it attained +8.5 degrees, the vehicle was asked to maintain a steady pitch of 0 degree. The command signal and response are given in figure 5.4. The dotted line corresponds to the command signal and the solid one to the actual response. The network learning rate was 0.1. It is found that a sudden transition, like a step signal, produces excessive overshoot in the response, and thus a trapezoidal command is necessary for a perfect tracking. This is because of the vehicle inertia and is observed in the case of heading control also.

2.3.1.2. Depth Control

The vehicle response for a rectangular velocity command is shown in figure 5.5. It can be seen that the vertical velocity relationship is easily controllable by the neural network controller. The vehicle is able to track the signal perfectly, even for a sudden transition from +0.15 to –0.15 (feet/sec.). The learning rate used for updating the weights is 0.002, which is much smaller than the value used in the earlier case. A rectangular command signal, in general, necessitates a smaller learning rate.

2.3.1.3. Heading Control

The vehicle was commanded go at a heading as shown in figure 5.6. The vehicle response is shown along with. As in the case of pitch, it is found that sudden transitions in the command are not possible. The heading has to be changed in a slow manner. The learning rate used for updating the controller is 0.02.

2.3.1.4. Control of Other Parameters:

The ability to control the roll angle by the rudder deflection is studied by presenting a rectangular command signal to the controller. With a learning rate of 0.5, the neural network controller is able to track the command perfectly(figure 5.7). It is to be noted that in practical

of 0.1 sec. is used in the case of linear systems and 0.05 sec. is used in the case of nonlinear systems; which is same as the time interval for one iteration for the neural network; (iii) The neural network used in all the cases are of size 2x20x10x1; (iv) The network weights are initialized to small values between +0.1 and –0.1 in all the cases; (v) On-line learning is used for the control with the network weights updated at every iteration; (vi) The learning rate is varied in each case for a better control and stability; (vii) No momentum term[14] is used in the BP learning algorithm; (viii) the gain factor is updated on-line using the expressions given in previous sections, starting from a rough estimate.

2.3.1. Linear Dynamics

2.3.1.1. Pitch Control

The objective, in this case, is to keep the vehicle at a specified pitch. In the first experiment, the vehicle was commanded to go at a pitch of –8.5 degrees and to +8.5 degrees afterwards.

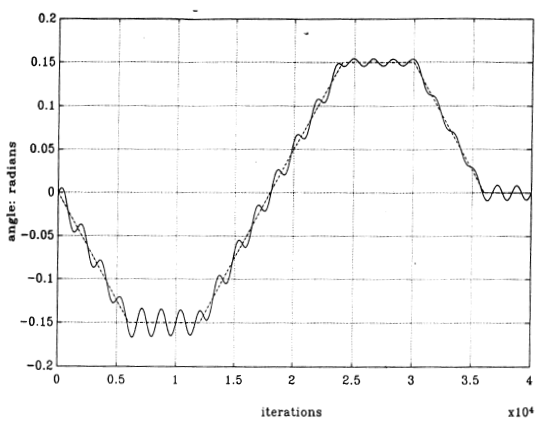


Figure 4.4 Tracking of the pitch command signal. Solid lines correspond to the vehicle response. The network weights are updated at each iteration

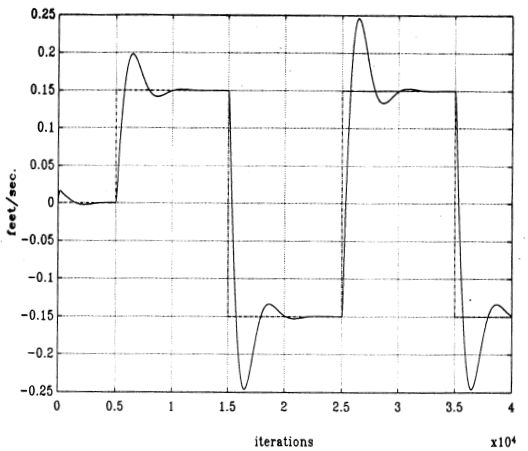


Figure 4.5 Tracking of the vertical velocity comm– and signal. Note that the initial overshoot could be controlled by changing the network learning rate. The response shown is for η = 0.002.

depth are discussed. Subsequently, the effectiveness of the gain layer scheme for the control of MIMO dynamics is presented by considering the scheme given in figure 5.2.

Each neural network controller has two input neurons, one for the command signal and the other for the vehicle output; the neural network output provides the control input to the vehicle. In the present studies four layer networks of size 2x20x10x10, are implemented.

2.2.1. Scaling of the command signal

For proper operation, the input to the neural network need to be in the range (–1.0, 1.0). Thus it is necessary to scale down the actual command signals which need not lie in the above range. For example, in the case of heading control, the commanded depth can be in range of hundreds of feet. There are two ways to tackle this problem; they are: (i) Normalize the command signal to the range (–1.0,1.0) before feeding as the input to the network; since any error outside the range (–1.0,1.0) is meaningless, the error also has to be scaled down by the same factor before using it for the updating of network interconnection strengths. (ii) Modify the training algorithm to incorporate a threshold function having larger bounds; this in turn, expands the bounds on the weight space and hence there is chance for larger fluctuations in the control input. Only the first method is used in the studies discussed here.

2.3. SIMULATION RESULTS

The feasibility of the controller architecture is studied through computer simulations. Three specific cases, involving the separate control of pitch, yaw and depth of the vehicle, are described here. In each case, the performance of the controller is studied by monitoring the speed of convergence, tracking error and stability of the system. The following considerations are made during the simulations: (i) The desired velocities and positions with respect to a global coordinate system is available from a path planning system; (ii) A sampling time

The path planner generates the desired velocities and positions at each time instant and those are given as inputs to the networks. θ_d and w_d are the desired pitch and vertical velocity and the control input δ_s is generated by combining the neural network outputs. The change in forward velocity due to the control of stern plane is neglected and only the depth and pitch are controlled in this scheme. Each network is trained using the corresponding errors. Figure 5.3 shows the rudder control scheme. ψ_d, ϕ_d and v_d denote the desired signals for yaw, roll and side velocity respectively. The corresponding outputs of the dynamics are ψ, ϕ and v. As in the case of stern plane control, the networks are trained using the corresponding errors.

The preliminary phase in the controller design process is to design the SISO controllers independently and optimize their performance. In the subsequent phases, the designed networks can be used as start–up controllers for further training for the combined schemes described

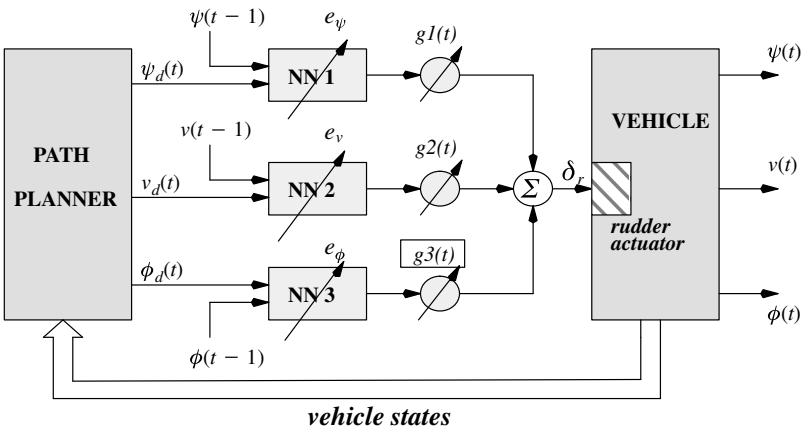


Figure 4.3 Controller architecture for the longitudinal motion

in figures 5.2 and 5.3 to implement the overall control scheme. In this section, the design of single–input single–output controllers for the output parameters of the vehicle yaw, pitch and

for the stern plane relationship. Each network controls one output parameter and the control input δ_s is generated by combining them as shown in the figure. *g1* and *g2* are the gain factors associated with each network. Splitting the controller into SISO networks has the following advantages[1]: (a) As in the case of conventional controllers, the behavior of SISO systems is well understood and hence controller design for them is easier than MIMO systems. (b) From the simulation studies, it is found that the learning rate of the networks is a critical factor, and it is easy to choose separate optimum values for each SISO network to get better convergence and tracking[53][65][66]. The coupling between the individual channels can be incorporated by explicit external connections to the input of the neural network from the other channels, even though it may complicate the convergence issues.

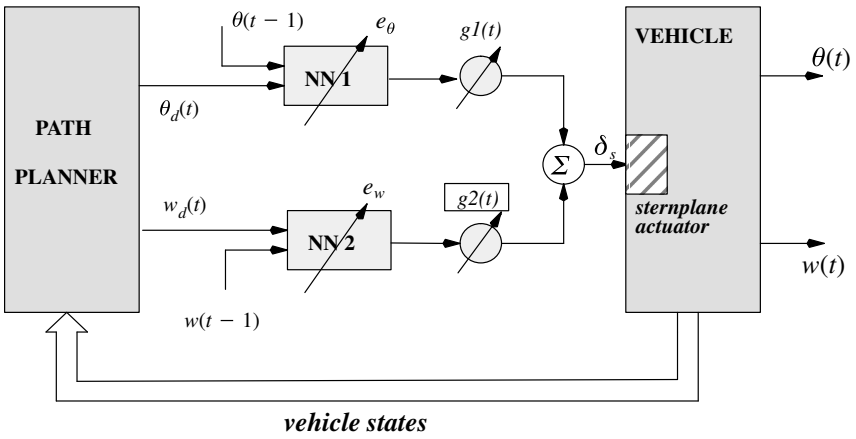


Figure 4.2 The controller architecture for the lateral motion. Each neural network and associated gain factor are trained with the corresponding error between the command signal and actual output. The vehicle states fed back to the path planner are the linear and angular velocities.

---

1. Multivariable control problem could be approached as separate SISO control of individual channels, when the coupling between channels are negligible. Even when the individual channels are coupled, controllers for individual channels can be designed separately considering the contributions from the other channels as disturbances[103].

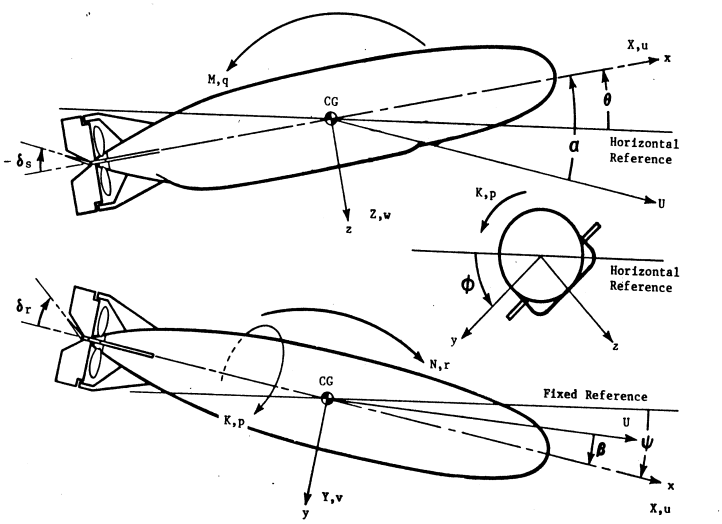


Figure 4.1 Details of the vehicle coordinate system (adopted from [96])

2.2. CONTROLLER IMPLEMENTATION

The objective of the controller is to make the vehicle track a desired trajectory with minimum error. The desired trajectory is related to the corresponding displacement along the three axes and the angular velocities. Hence the overall objective of the controller can be seen as tracking of the desired linear displacements and angular velocities.

It is found during the simulations of the dynamics of the 'Ocean Voyager' that the rudder and stern plane relationships of the vehicle are fairly separable and hence the control scheme could be simplified, into two separate implementations[66] which are; (i) the subsystem with the rudder deflection δ_r as control input and with the output parameters yaw(ψ), roll(ϕ) and side velocity(v) (ii) the subsystem with the stern plane deflection δ_s as the control input and with the output parameters vertical velocity(w) and pitch(θ) as the outputs; the effect of stern plane deflection on the forward speed is negligible. Figure 5.2 shows the control scheme

of the controller; and, (c) the ability of the network controller to adapt to sudden and slow parameter variations in the dynamics.

2.1. THE VEHICLE DYNAMICS

The dynamics used in the studies is that of the 'Ocean Voyager', an AUV being built at the Center for Advanced Marine Systems, Florida Atlantic University. The vehicle is about 22' long and 21'' in diameter with a displacement of 2700 lb. The forward propulsion is of thruster type with two control surfaces; the stern plane and the rudder. The designed maximum speed of the vehicle is 10.13 ft./sec. and an overall cruise time of 120 minutes. The dynamics is developed from the conventional Euler equations of motion for a rigid body, as given below[95][96]:

$$\Sigma X = m(U' + QW - RV)$$

$$\Sigma Y = m(V' + RU - PW)$$

$$\Sigma Z = m(W' + PV - QV)$$

$$\Sigma K = P' I_x - R' I_{xy} + QR(I_z - I_y) - PQ I_{xz}$$

$$\Sigma M = Q' I_y + PR (I_x - I_z) - R^2 I_{xz} + P^2 I_{xz}$$

$$\Sigma N = R' I_z - P' I_{xz} + PQ (I_y - I_x) + QR I_{xz}$$

where ΣX, ΣY and ΣZ are the summation of the forces along three co–ordinate axis. ΣK, ΣM and ΣN corresponds to total moment of inertia with respect to these axes. U, V and W are the linear velocities while P, Q and R are the angular velocities. U', V' and W' and P', Q' and R' correspond to the derivatives of the total velocities and I stands for the moment of inertia, with the subscripts indicating whether planar or axial. See figure 5.1 for details on the coordinate descriptions. The complete set of nonlinear equations of motion and the linearized model are given in Appendix C.

CHAPTER 2.

LEARNING CONTROL OF DYNAMICAL SYSTEMS USING FEEDFORWARD NEURAL NETWORKS: PART II

In the previous chapter, the existing direct control schemes for the on–line learning control of dynamical systems were discussed in detail. Two modifications were proposed. It was shown that the proposed methods (gain layer schemes) provide better dynamic response and tracking performance than the existing schemes. More detailed studies on the gain layer scheme are reported in this chapter. This chapter also illustrates how the gain layer scheme can be used for the control of MIMO dynamical systems by splitting the dynamics into SISO cases. The studies are conducted both on the linearized dynamics as well as the nonlinear dynamics of an autonomous underwater vehicle. Three specific issues are addressed in these studies; they are: (a) comparison of the complexity of the network in terms of the number of interconnection strengths with linear and nonlinear dynamics; (b) the effect of network parameters, such as learning rate and initial weight distribution, on the tracking performance