NEURAL ADAPTIVE CONTROL FOR LAUNCH VEHICLE AUTOPILOT

Mohanlal P. P., Harisankar M., Dasgupta S.

ABSTRACT

Launch vehicle autopilot control performance can be appreciably improved using nonlinear adaptive control methodology since it can cater to the nonlinear, timevarying and uncertain nature of the plant involved. Moreover, the fault tolerant nature of the neural computation makes it ideal for launch vehicle adaptive control if stability based training methods are developed. In this paper, the feasibility of using two adaptive neural controllers for a launch vehicle autopilot to achieve optimum performance under large plant perturbations are studied and a suitable neural controller configuration is evolved for unstable, nonminimum phase plants. Optimality of performance achieved through simulation studies.

I. INTRODUCTION

In the past three decades, major advances have been made in adaptive identification and control for identifying and controlling linear time invariant plants with unknown parameters. Stable adaptive laws for the adjustment of paramaters in these cases which assure global stability of the relevant overall systems are based on properties of linear systems. But very few results exist in nonlinear systems theory which can be directly applied for the choice of the nonlinear identifier and controller structures as well as the generation of adaptive laws for the adjustment of the parameters. Hence considarable care has to be exercised in the adaptive control of nonlinear systems.

The area of artificial neural networks, in recent years has received considerable attention. Feed forward and Recurrent classes of neural networks, from a systems theoretic point of view represent static nonlinear maps and nonlinear dynamic feedback systems respectively.

Identification as well as controller structures using neural networks for the adaptive control of unknown nonlinear dynamical systems have been introduced by Narendra and Parthasarathy [1]. Their unified approach to the feedforward and recurrent classes of networks through the introduction of the idea of dynamic backpropagation is remarkable. However, their work was limited to the adaptive control of inherently stable nonlinear systems as well as the global stability of the overall systems were not addressed.

The neural adaptive control of nolinear systems is of great importance since it does not require setting up and solution of complex nonlinear equations. In addition, the neural network's ability to incremental learning motivates research efforts on neural adaptive control of nonlinear systems.

At present, the launch vehicle control is designed by approximating the plant model as linear models for different time slices. In reality, the plant is nonlinear in nature due to actuator slew rate limits, actuator saturation limits etc and time varying due to mass, inertia, environmental changes and C.G migration with time. In addition the plant is uncertain due to dispersions in thurst-time curves and environmental factors. Hence the nonlinear adaptive control methodology will improve the control performance substantially since it can cater to the nonlinear, time varying and uncertain nature of the plant. Moreover, the fault tolerant nature of the neural computation makes it ideal for launch vehicle adaptive control if stability based training methods are developed.

In this paper, the feasibility of using two adaptive neural controllers for a launch vehicle autopilot to achieve optimum performance under large plant perturbations are studied and a suitable neural controller configuration is evolved and optimality of performance achieved through simulation studies.

2. ADALINE BASED ADAPTIVE NEURAL CONTROLLER

ADALINE, the ADAptive LINear Element was developed by Widrow and others [2,3] during 1960s and demonstrated its applications to adaptive signal processing such as adaptive noise cancellation, adaptive channel equalisation and adaptive linear prediction etc. Basically, it is a single layer linear (linear activation function) neural network which is an extension of the single layer perceptron network of Resenblatt [4]. The basic structure of ADALINE is shown below in Fig. 1.

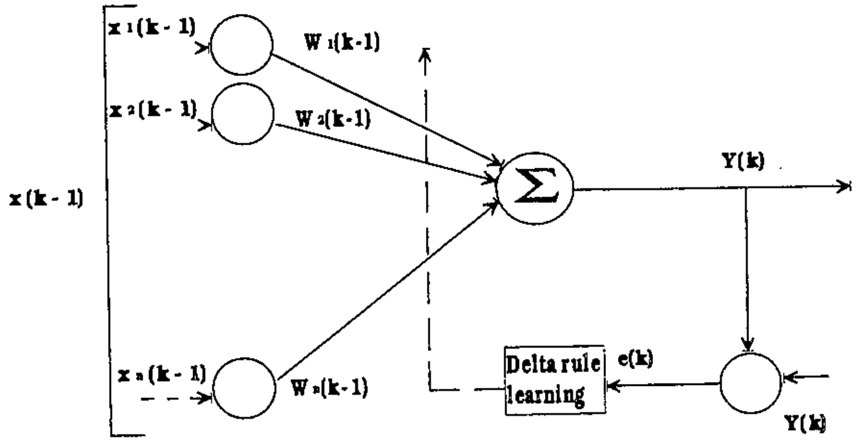


Fig. 1.

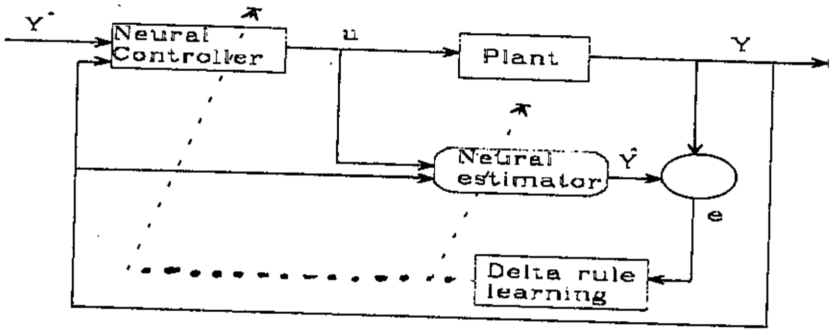
where, $x(k-1) = [x_1(k-1)\ x_2(k-1) \ldots\ldots\ldots\ldots x_n(k-1)]^T$ is the input vector at time instant (k-1) $w(k-1) = [w_1(k-1)\ w_2(k-1) \ldots\ldots\ldots w_n(k-1)]^T$ is the weight vector at time (k-1) Y(k) : the predicted output at time k. y(k) : desired/ actual output at time k. The delta rule learning is developed by Widrow and Hoff [3] and is also known as the LMS algorithm.

The delta rule is a steepest descent gradient algorithm which seeks the minimum of the error performance surface. The error performance surface of the network in Fig. 1 is a hyper paraboloid in (n+1) dimensional space. It is important to note that it has only one minimum which is the global minimum and the orientation of the paraboloid remains fixed in (n+1) dimensional space if the input vectors and the desired vectors are jointly stationary which is the case for time invariant systems. For time varying systems the orientation of this error surface will be changing with time [3].

Nonlinear relationships between inputs and targets cannot be represented exactly by such linear networks. But they make a linear approximation with minimum sum-squared error [5]. Eventhough, the linear approximation can degrade performance for nonlinear plants, this approximation is very important and useful if the nonlinearities are not too severe and also the adaptive version of this linear approximation is useful if the plant is slowly time varying. The obvious advantages are due to the solid theoritical foundation of the LMS algorithm with respect to its stability and convergence properties. Because of the above said reasons, the ADALINE based adaptive neural control scheme is investigated to see whether it can be adopted for launch vehicle autopilot.

2.1. Neural Adaptive Control Scheme [6]

As discussed above, the plant is assumed to be the linear approximation of the actual plant at any instant and it is allowed to vary with time. The basic scheme is shown below in Fig. 2.



A daptive neural scheme

Fig. 2

2.1.1. Plant Model

We shall deal with a linear DARMA (Discrete Auto Regressive Moving Average) plant described by

$$A(q^{-1})y(k) = B(q^{-1})u(k) \quad \text{..........................} \qquad (1)$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials defined by,

$$A(q^{-1}) = 1+a_1q^{-1} + \text{.................}+a_nq^{-n} \text{ ....} \qquad (2)$$

$$B(q^{-1}) = b_1q^{-1} + \text{.......................}+b_mq^{-m} \qquad (3)$$

With the q^{-1}, the backward shift operator [i.e., $q^{-1}y(k) = y(k-1)$]. u(k) and y(k) are the plant input and output respectively.

2.1.2. Assumptions

Three basic assumptions are made on the plant. (a) Upper bounds of the orders of the plant polynomials n and m are known and are n* and m*. (b) $B(q^{-1})$ is a stable polynomial. (i.e. plant is a minimum phase plant). (c) The coefficient $b_1 \neq 0$.

Assumption (a) allows the true order system to be overestimated. Although the neural model of the plant and the controllers depend on n, m; the effect of using these upper bounds of n*, m* instead of the true values in the neural networks design is simply that some of the connection weights may tend to zero.

Assumption (b) is necessary to obtain perfect tracking and closed loop stability for the proposed neural controller. Assumption (c) is required for the implementation of the controller.

2.2. Neural Model of the plant

The ADALIN of the form in Fig. 1. is chosen as the model of the plant. The identification structure is the series - parallel structure used by Narendra [1]. The input vector consists of (n + m) components. These components are the input-output signals measured at previous instants in the plant. It has only one output. This model estimates the plant's regular dynamics and is depicted in Fig. 3.

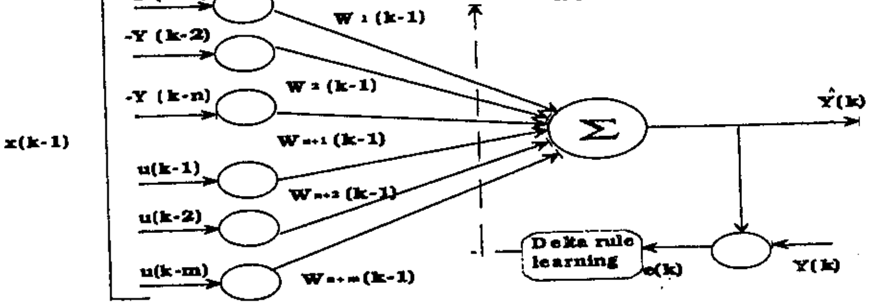


Fig. 3.

The weight vector is .

$$w(k) = [w_1(k), \ldots\ldots\ldots\ldots\ldots\ldots\ w_n(k),\ w_{n+1}(k), \ldots\ldots\ldots\ldots\ldots\ldots\ w_{n+m}(k)]^T \quad (4)$$

and is updated using Widrow-Hoff delta rule as $w(k+1) = w(k) + (\alpha\, e(k+1)\, x(k) / \varepsilon + x^T(k)\, x(k))$ where $\alpha \in (0,2)$ is the learning rate and ε is chosen to be close to zero and is included to avoid division by zero in eqn. (5). y(k) is the plant response.

2.3. Neural controller

The learning of the plant dynamics by the neural estimator presented above is used to adjust the connection weights of a neural controller which generates the control signal u(k). This control signal, when applied to the plant input, is intended to bring the plant output y(k) to a desired reference signal $y^*(k)$. The controller is shown in Fig. 4.

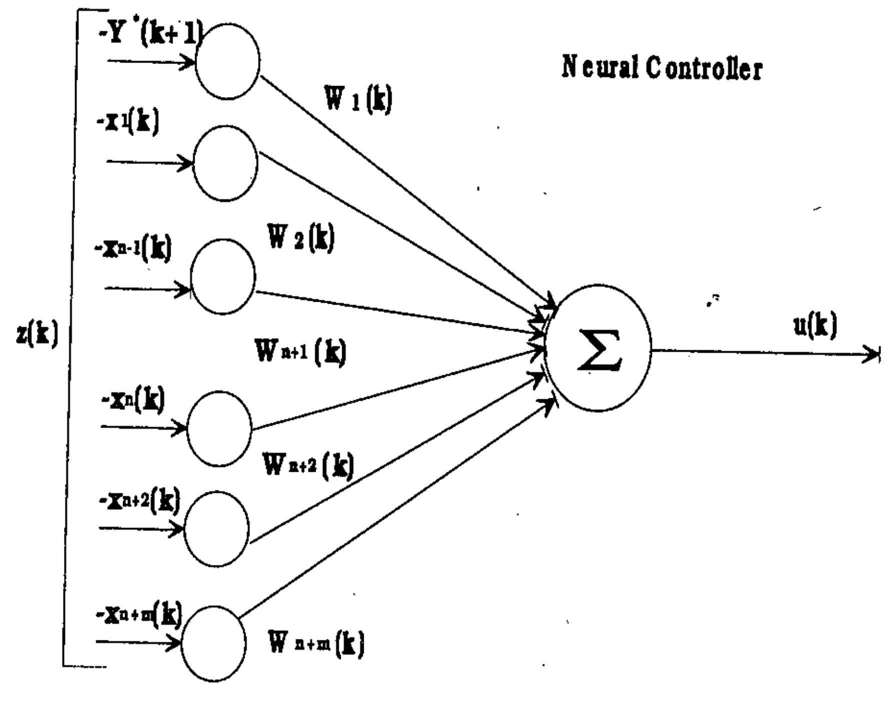


Fig. 4.

The neural controller consists of a second ADALIN with (n+m) input components and one output. The connection weights w'(k) is defined as a function of the neural model weights as follows.

$$W(k) = [1, W_1(k), W_2(k), \ldots\ldots W_n(k), W_{n+2}(k), \ldots W_{n+m}(k)]^T / W_{n+1}(k) \quad (6)$$

2.4. Sequence of actions in the control loop at instant k:

(a) Measure $y^*(k+1)$ and y(k). (b) Use the neural model of the plant to

Compute the predicted output y(k) using the old weights w_i (k-1). (c) Compute the error signal e(k) = y(k) - y(k) and use the delta rule to calculate the new weights w_i (k). (d) Update the neural controller weights $w'_{.i}(k)$ using eqn. (6).
(e) use the neural controller to generate the control signal u(k).

2.5. Closed loop properties

The neural control scheme is stable and proof of stability in available in [6]. Also the control objective is asymptotically achieved.

2.6. Simulation Results

Example:1 (Unstable plant)

Let us consider the following 5th order unstable plant.

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{q^{-1} - 0.0449\, q^{-2} - 0.0769\, q^{-3} - 0.0349\, q^{-4} + 0.1518\, q^{-5}}{1 + 0.2037\, q^{-1} - 0.7790\, q^{-2} - 0.3840\, q^{-3} - 0.0421\, q^{-4} + 0.0014\, q^{-5}}$$

Here, n = m = 5, α is set to 1, $\varepsilon = 0.001$. Initial values of weight vector is choosen as random values. The reference signal is a square wave of unit ± 1V amplitide and period 100 samples. Figs 5& 6 show the reference input and controlled system output y(k) after about 80,000 iterations. Closed loop response is stabilized eventhough the plant itself is unstable. Fig 7&8 give open loop response and control input respectively. The neural network is able to learn the plant dynamics quite well. The large number of initial iterations are due to the random selection of initial weights.

Example 2

Adaptation While the plant transfer function changes from a unstable TF to a Stable TF: initially the plant is chosen as a unstable plant as in example:1. After the controlled output has converged sufficiently well, the plant transfer function is changed to a stable TF as below

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{0.7q^{-1} + 0.3q^{-2} + 0.25\, q^{-3} + 0.1\, q^{-4} + 0.05\, q^{-5}}{1 - 0.7946\, q^{-1} - 0.0358\, q^{-2} + 0.4312\, q^{-3} + 0.0574\, q^{-4} - 0.0095q^{-5}}$$

at the iteration number k = 201. The controlled response and control input are shown in Fig. 9 & 10. The adaptation transient can be seen at k = 201 and the adaptation duration is about 40 samples. For a sampling rate of 50 samples / sec, this amounts to an adaptation duration of 0.8 Sec which is a quite fast.

Example 3

Adaptation while the plant transfer function changes from a stable TF to a unstable TF: Initially the plant is chosen as a stable plant in example 2.

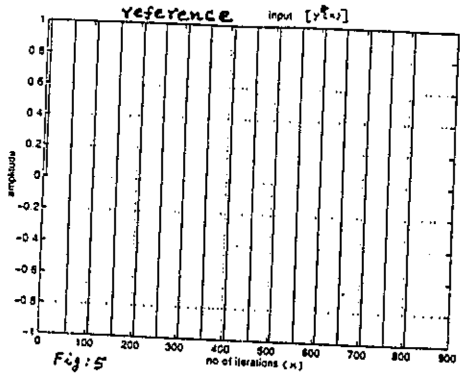


Fig: 5

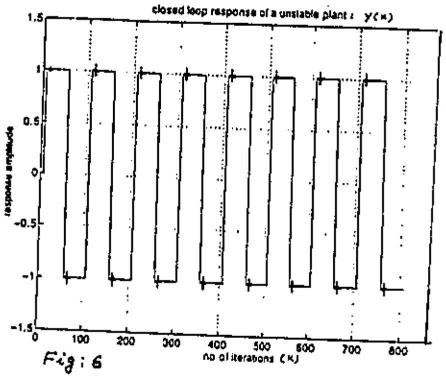


Fig: 6

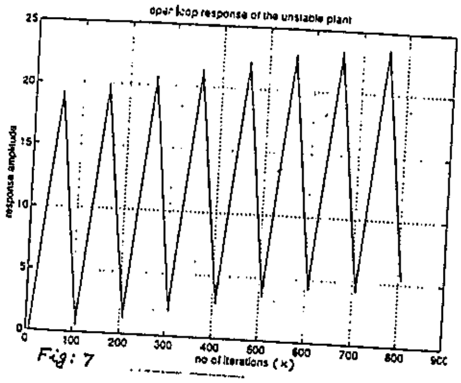


Fig: 7

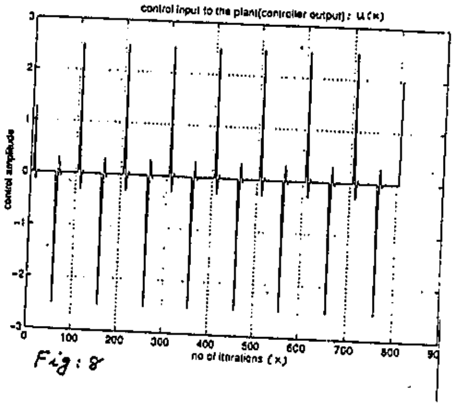


Fig: 8

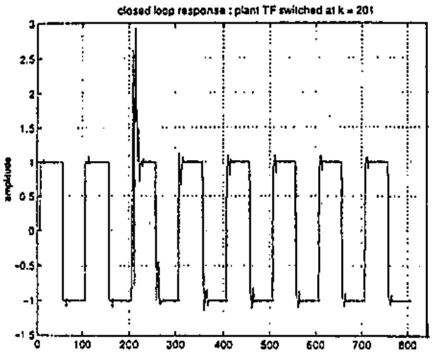


Fig: 9

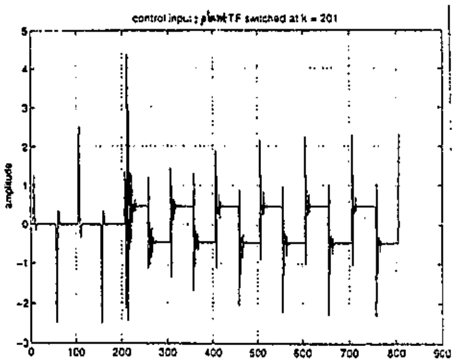


Fig: 10

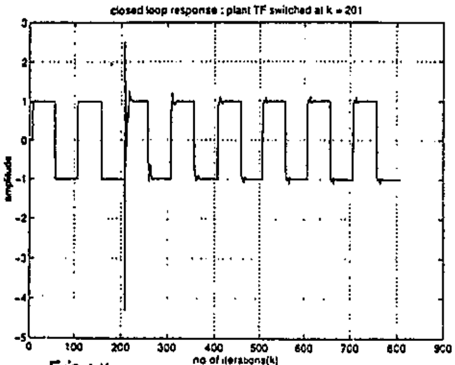


Fig: 11

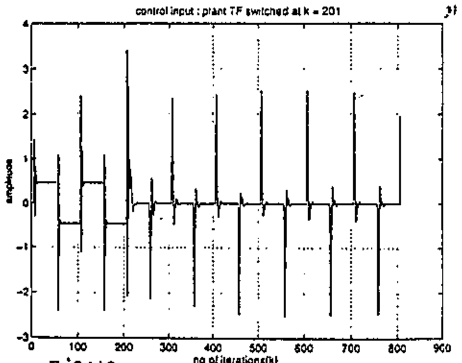


Fig: 12

After the controlled output has converged sufficiently well, the plant TF is changed to the unstable TF as in example 1 at the iteration number k = 201. The controlled response and control input are shown in Fig 11 & 12. The adaptation duration is again about 40 samples.

2.7. Suitability for Digital autopilot design for Launch vehicle:

The ADALIN based scheme caters to plants for which linear approximation is valid and the plant TF is of minimum phase.. This scheme is typically suitable for upper stages of launch vehicle autopilot where the plant essentially is of minimum phase in nature.

Since it is desired to evolve an adaptive control scheme for non-minimum phase plants also, another adaptive control scheme is investigated in the next section.

3.DYNAMIC NEURAL UNIT BASED ADAPTIVE CONTROL SCHEME:

3.1. Dynamic Neural Unit:

The dynamic neural unit (DNU), proposed by Gupta and Rao [7,8], is a dynamic model of the biological neuron. The neural structure was motivated by the fact that biological neuronal systems always function with continous feedback. One example of such a system is the reverberating circuit in the neuronal pool of the CNS. The basic structure of one DNU is shown in Fig. 13.

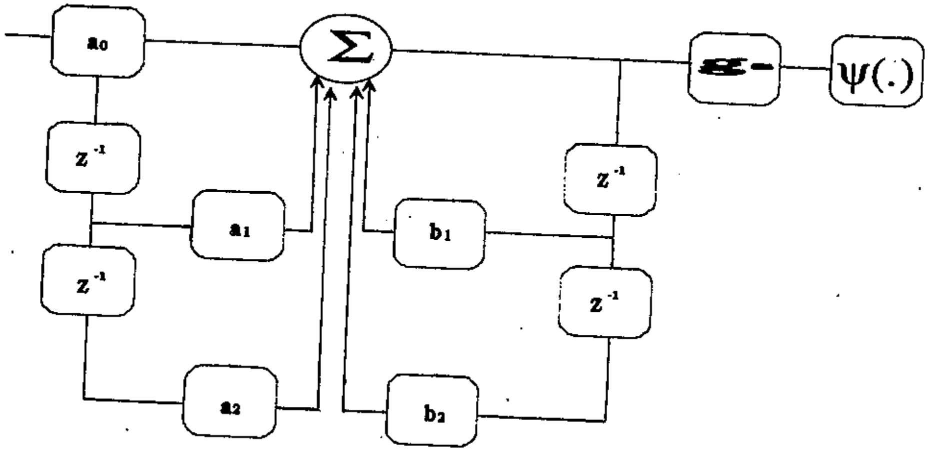


Fig. 13.

The DNU consists of delay elements, feed forward and feedback paths weighted by the synaptic weights a_{ff} and b_{fb} respectively followed by a non-linear activation function. The linear dynamics of the DNU can be expressed in the form of a transfer relation:

$$W(k,a_{ff},b_{fb}) \;=\; V_1(k) / \; S(k) \;=\; [a_0+a_1z^{-1}+a_2z^{-2}] \,/\, [1+b_1z^{-1}+b_2z^{-2}]$$

Where, $S(k) = [\Sigma \; W_iS_i - \theta]$, $i = 1$ to n is the neural input to DNU, $S_i \in R^n$ are the inputs from other neurons, $W_i \in R^n$ are the corresponding input weights, θ is the bias term, $V_i\,(k) \in R^1$ is the output of the dynamic structure, $U(K) \in R^1$ is the neural output, $a_{ff} = [a_0\, a_1\, a_2]^T$ and $b_{fb} = [b_1\, b_2]^T$ are the vectors of adaptable feed forward and feedback weights respectively. Let us define the vectors of signals and adaptable weights of DNU as

$$X(k,V_1,S) \;= [V_1(k\text{-}1)\;\; V_1(k\text{-}2)\; S(k)\; S(k\text{-}1)\; S(k\text{-}2)]^T \qquad (8)$$

$$W(a_{ff},\, b_{ff}) \;=\; [\text{-}b_1\text{-}b_2\; a_0\; a_1\; a_2]^T \qquad (9)$$

$$V_1(k) \;= W^T X \qquad (10)$$

The nonlinear mapping operation on V_1 (k) yields a neural output U(k) given by

$$U\,(K) = \psi\,[g_s\; V_1\,(k)] \qquad (11)$$

Where ψ [.] is the nonlinear activation function. Adaptation of parameters of W results in synaptic adaptation and adaptation of g_S results in somantic adaptation.

3.2. Learning and Adaptive algorithm of a DNU:

If the weights of the DNU are considered as the elements of a parameter vector W (a_{ff},b_{fb}), then the goal of the learning and adaptive algorithm is to determine the vector W* (a_{ff},b_{fb}) which optimises a performance index J based on the output error. This learning process will cause the neural output U (K) to iteratively approach the desired state Yd (k). Let us define the error as e(k) = Yd (k) - U(k) and the performance index

$$J \;\; = \tfrac{1}{2}\, E\, \{e^2\,(k;\, W(a_{ff},\, b_{fb})\} \qquad (12)$$

has to be minimised with respect to the weighting vector W (a_{ff},b_{fb}). E is the expectation operator. Thus the back propagation based adaptation alogorithms [7] can be written as

$$a_{ffi}\,(k+1) = a_{ffi}(k) + \mu ai \;\; E\,\{e(k).\, g_s\, \psi'\,(v)\}\;\; g_s \;\; X(k\text{-}i) \qquad (13)$$
$$i = 0,1,2$$

$$b_{fbj}\,(k+1) = b_{fbj}(k) + \mu bj \;\; E\{e(k)\;\; g_s\;\; \psi'\,(v)\}\;\; (\text{-}g_s)\;\; V_1(k\text{-}j) \qquad (14)$$
$$j = 1,2$$

$$g_s(k+1) = g_s(k)\; [1+\mu_g \;\;\; E\{e(k)\; \psi'\,(v)\; v_1(k)\}] \qquad (15)$$

where μai, μbj and μ_g are learning rate constants. The learning and representation abilities of a single DNU are much higher compared to a conventional neuron by virture of its proposed structure.

3.3. DNU for adaptive control

Gupta and Rao [8] have demonstrated the application of DNUs for the adaptive control of linear and nonlinear systems. Their aim was to show the faster adapting rate of the DNUs for adaptive control. For this, they have chosen the simple inverse dynamics adaptive control scheme and did not use any identification model.

The inherent assumption in this kind of scheme is that the Jacobian of the plant do not change sign for various ranges of input-output. Also the plant is assumed to be a stable plant. These restrictions on the plant clearly indicate that the inverse adaptive scheme which they had used cannot be applied for general unstable, nonminimum phase plants.

However, the fast adaptation abilities of DNUs demonstrated by them may be quite useful for a suitable alternate adaptive control scheme. This is the motivation to search for a suitable adaptive control configuration which can accommodate general unstable, nonminimum phase plants and make use of the fast adaptation characteristics of the DNU architecture. The DNU architecture can be viewed as a general feed back nonlinear dynamical system with the ability of very fast adaptation.

4. NEURAL ADAPTIVE CONTROLLER FOR LAUNCH VEHICLE DIGITAL AUTOPILOT [9]

The general block diagram of Launch vehicle control scheme in one of the three planes (pitch, yaw & roll) is shown in Fig. 14. In this, the inner loop control is mainly for stabilizing the vehicle and deals with the short period dynamics and the outer loop control is the guidance loop which deals with the long period dynamics. The inner loop control is termed as the Autopilot loop.

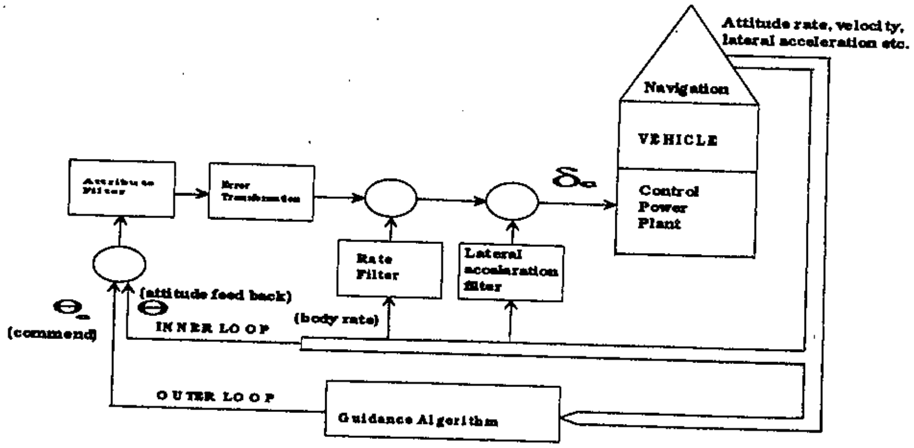


Fig. 14.

Here, the digital autopilot of a satellite launch vehicle in the yaw plane is considered. Specifically, the plant transfer function at high dynamic pressure region of flight is considered.

4.1. Plant model

The full plant model is described by a 23*23 system matrix in continuous time state space representation in yaw plane which considers flexibility of the vehicle also. This involves 23 state variables. They are

1) Attitude (ψ)
2) Rate ($\psi^{\cdot}$)
3) Drift (V/U_0) V: lateral velocity U_o : axial velocity
4) Slosh 1 - displacement : ($\mu_1^{\cdot}$)
5) Slosh 1 - rate : ($\mu_1{\cdot}$)
6) Slosh 2 - displacement : ($\mu_2^{\cdot}$)
7) Slosh 2 - rate : ($\mu_2{\cdot}$)
8, 9) Slosh 3- displacement & rate (μ_3, $\mu_3^{\cdot}$)
10,11) Slosh 4 - displacement & rate ($\mu 4$, $\mu 4^{\cdot}$)
12,13) Bending mode 1 - displacement & rate (q_1, $q_1^{\cdot}$)
14,15) Bending mode 2 - disp & rate (q_2, $q_2^{\cdot\cdot}$)
16,17) Bending mode 3 - disp & rate (q_3, $q_3^{\cdot}$)
18,19) Bending mode 4 - disp & rate (q_4, $q_4^{\cdot}$)
20, 21) Bending mode 5 - disp & rate (q_5, $q_5^{\cdot}$)
22) Actuator command - (δc)
23) Actuator command rate - ($\delta^{\cdot}c$)

The state variables the correspond to the rigid body dynamics are Sl.Nos :1,2,3,22,23 from above. The four slosh modes are induced by the liquid propellant stages. These modes are negligible for the considered flight which is assumed to be a solid propellant stage. Also out of the 5 bending modes, the two predominant modes BM1 & BM2 are only considered for the flexible body case of the above flight. Thus for the rigid body case, slosh modes and bending modes are not considered which result in 5 state variables and hence a 5*5 system matrix. Similarly, the total number of states considered for the flexible body case is 9. The corresponding system matrix is 9*9. The state space representation of the vehicle model is as follows

$$X^{\cdot} = AX + BU$$
$$Y = CX + DU$$

Where A, B, C, D are system matrices. X : state vector, U : is the input : δc
Y : output vector : $[\psi\ \psi^{\cdot}]^T$

4.1.1. Discretization of the plant model

Continuous time state space A, B, C, D matrices of the reduced order cases are converted to continuous time transfer functions first. Then continuous time transfer functions are converted to discrete time transfer functions using 20 ms sampling time and zero order hold method. The two discrete time transfer functions are

a) attitutde output / actuator command I/P = ψ / δc

b) attitude rate output / actuator command I/P $= \psi / \delta c$

They are of fifth order for rigid body case and of nineth order for the flexible body case which are considered here. The discrete transfer functions thus obtained were found to be unstable as well as nonminimum phase for both cases.

4.2. Selection of Neural control configuration for Digital Autopilot design of a Satellite Launch Vehicle

The ADALINE based neural scheme cannot be used for this application since the plant is of nonminimum phase type. The Adaptive control schemes used by Gupta and Rao [8] cannot be used here since the plant here is unstable and nonminimum phase. Also, the adaptive control schemes proposed by Narendra and Parthasarathy [1] cannot be used here since it is not practicable to have a neural identification model for a unstable plant. Also in the design of Digital Autopilot for a Launch vehicle, stability of the closed loop system is atmost essential. Hence it is desired to design a conventional linear feedback controller to ensure stability of the closed loop system utilising the available knowledge of the plant. Thus a conventional linear output feed back control technique is adopted for the stabilization of the plant. This controller configuration is shown in Fig. 15.

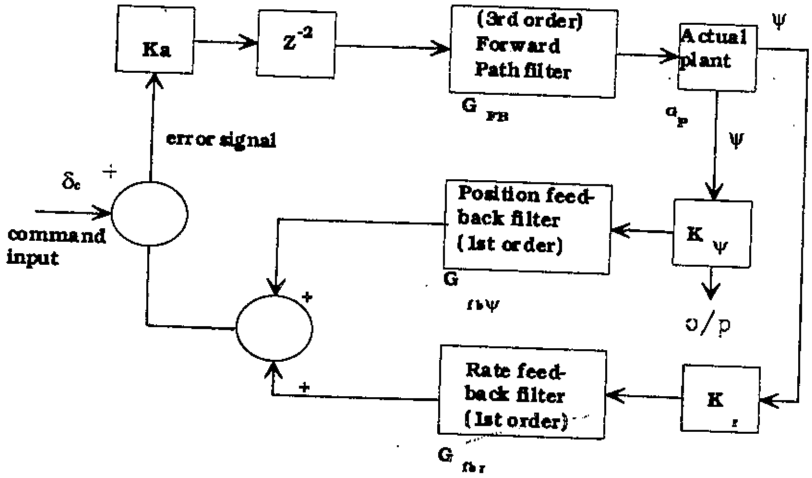

Ka
z^-2
(3rd order) Forward Path filter
G FB
Actual plant
ψ
G P
ψ
error signal
δc
+
command input
Position feed-back filter (1st order)
G fbψ
K ψ
o/p
+
+
Rate feed-back filter (1st order)
G fbr
K r

Fig. 15.

The typical values of the coefficients of the controller and plants are given below. ka = 0.477, Position gain kψ = 1.0, Rate gain kr = 0.645

$G_{tp} = 0.52 + (0.33z^{-1} - 0.41z^{-2}) / (1-0.57z^{-1} - 0.15z^{-2}) + (0.001 z^{-1} / (1-0.99z^{-1}))$
$G_{fb\psi} = 0.59 - 0.51z^{-1} / (0.59 - 0.51z^{-1})$
$G_{fbr} = 0.22 - 0.16z^{-1} / (1-0.94 z^{-1})$

and the rigid body nominal plant TFs are

$$G_{p\psi} = \frac{0 + 0.0002z^{-1} + 0.001\,z^{-2} - 0.0002\,z^{-3} - 0.001\,z^{-4} - 0.0001z^{-5}}{1 - 4.24z^{-1} + 7.17z^{-2} - 6.07z^{-3} + 2.6z^{-4} - 0.45z^{-5}}$$

$$G_{p\psi} = \frac{0 + 0.03z^{-1} + 0.04z^{-2} - 0.14z^{-3} + 0.06z^{-4} + 0.02z^{-5}}{1 - 4.24z^{-1} + 7.17z^{-2} - 6.07z^{-3} + 2.6z^{-4} - 0.45z^{-5}}$$

The linear output feedback controller is designed for the nominal TF of the plant. Once stability of the closed loop system is achieved by the proven linear feedback technique, it is logical to think of an additional neural controller which can be added to the scheme in fig. 15 specifically for the purpose of on-line performance optimisation for the expected extremes of the plant TF perturbations. It is with this logic, the following way of adding the neural controller to the linear feedback stabilized system is suggested.

4.2.1. Neural controller in shunt with the forward path filter

The overall configuration with the neural controller is shown in Fig. 16.

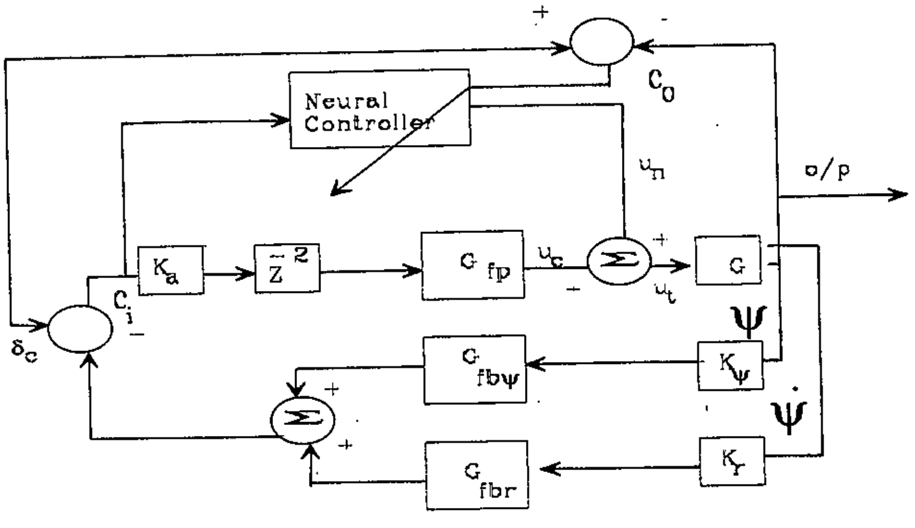


Fig. 16.

This location of the neural controller requires only a simple 3 DNU cascade neural controller compared to other possibilities. DNU based neural controller is chosen in order to exploit the fast adaptation capability of the DNU structure explained as before.

In this configuration, the input to the neural controller is the error signal from the linear controller. The error index to be minimised is $||eo|| = ½ [\delta c - \psi]^2$. The sign of the jacobian of the plant $\partial\psi$ is ∂u_n computed during every iteration and used with standard back propagation algorithm to adapt the DNU controller to minimise the error index above.

The neural controller which is in shunt with the forward path filter actually auguments the forward path filter dynamics to achieve the required closed loop performance optimisation.

4.2.2. Simulation procedure

First, small random weights were assigned for the neural controller. The plant is chosen as the nominal rigid body TF. Then the neural network is trained for sufficient number of iterations until the output response becomes satisfactory. The set of final weights so calculated is taken as the starting weights for all perturbation studies. Hence the adaptation of the neural controller weights for plant perturbations is from a initial nominal optimised weights learned sufficiently well earlier. This is a logically acceptable procedure since the nominal plant TF is always known and is the one used for conventional linear feedback controller design.

The performance analysis is carried out by simulation studies with the reference input chosen as the unit step input for the convenience of performance comparison.

4.2.3. Simulation Cases *Rigid body cases and performances*

a) Nominal rigid body Figs 17&18 (Kr = 0.645, $\mu\alpha = -1.2$)
b) With high angley of attack Figs 19&20 (Kr = 0.645, $\mu\alpha = -2.4$)
c) With lower angley of attack Figs 21&22 (Kr = 0.645, $\mu\alpha = -0.6$)
d) With lower rate gain Fig 23 (Kr = 0.5, $\mu\alpha = -1.2$)
e) With higher rate gain Fig 24 (Kr = 0.8, $\mu\alpha = -1.2$)
$\mu\alpha$: angley of attack coefficient

Flexible body cases and performances

a) Nominal flexible body plant (actuator BW = 5Hz & damping = 0.7)
Figs 25 & 26.
b) With reduced actuator bandwidth (actuator BW = 2.5 Hz & damping = 1.0)
Figs 27& 28.
c) With the two bending mode frequencies reduced by 15% from the nominal
($W_1 = 0.85\ W_1^*$, $W_2 = 0.85\ W_2^*$)
Figs 29 &30.
d) With the two bending mode frequencies increased by 15% from the nominal
($W_1 = 1.15\ W_1^*$, $W_2 = 1.15\ W_2^*$)
Figs 31 & 32.

Where W_1^* and W_2^* are the nominal first and second bending mode frequencies respectively.

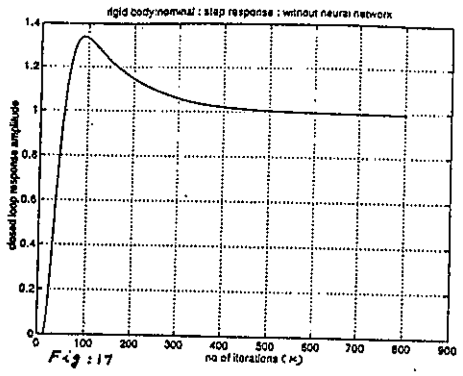


Fig: 17

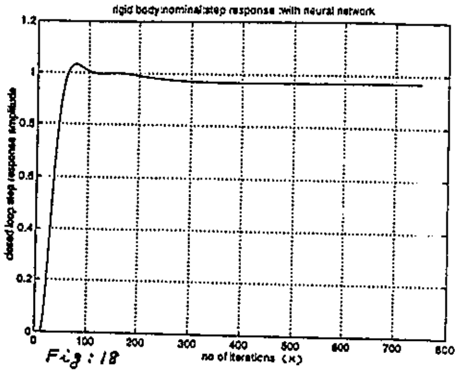


Fig: 18

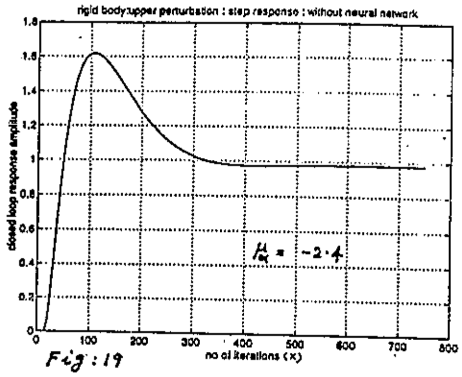


Fig: 19

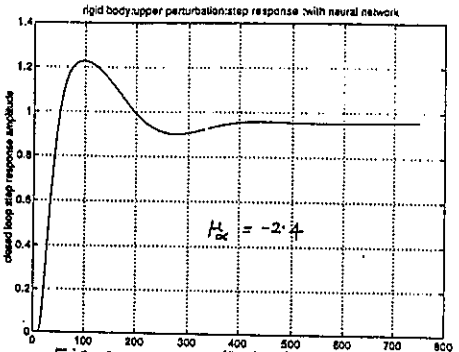


Fig: 20

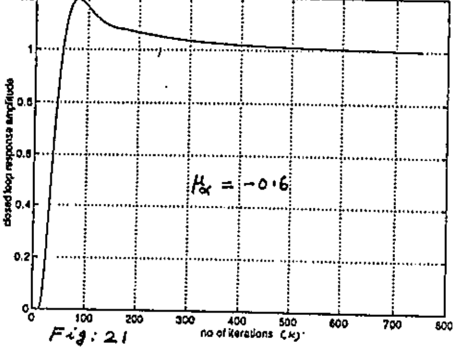


Fig: 21

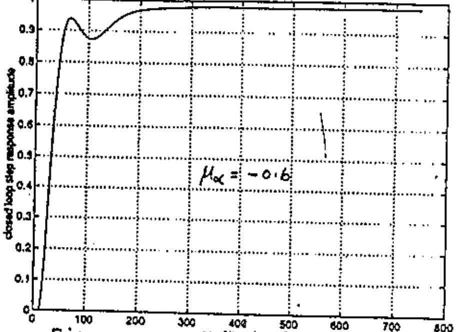


Fig: 22

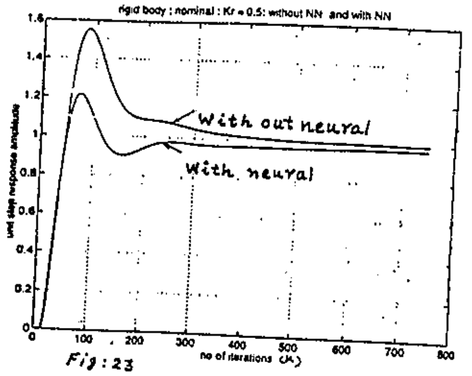


Fig: 23

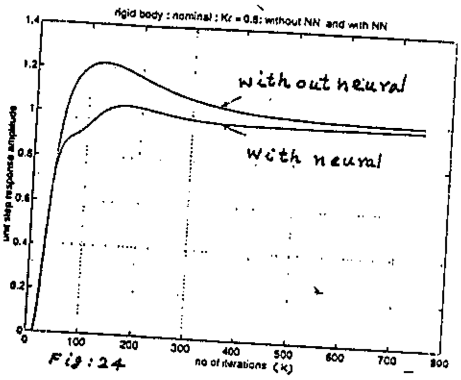


Fig: 24

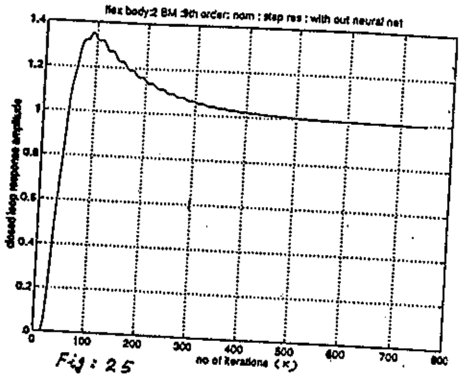


Fig: 25

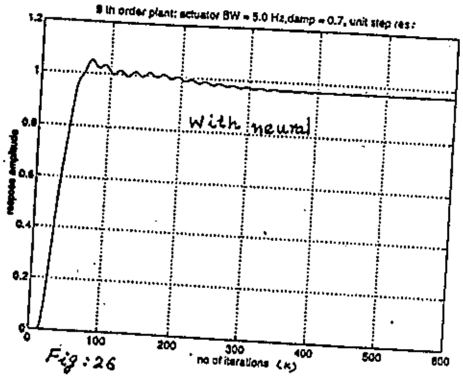


Fig: 26

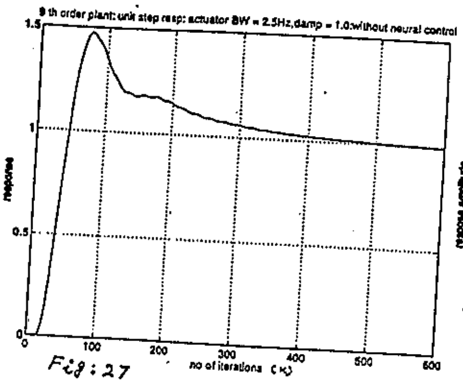


Fig: 27

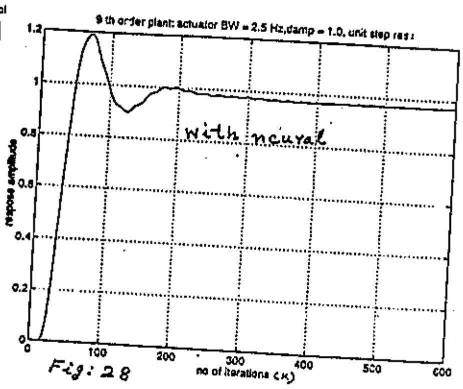


Fig: 28

5. CONCLUSION

The 3 DNU cascade neural controller and the new method of augumenting the linear output feedback control configuration is proved to be very effective by simulation studies for the adaptive control of satellite launch vehicle autopilot. This is evident from the performance summaries given in section IV. The scheme developed is particularly important since it caters to unstable and non minimum phase plants.

For flexible body cases, the bending mode responses could not be suppresses using the present error norm used for the neural controller. However the neural controller has not worsesed the bending mode response. Suitable modification of the error norm may result in proper attenuation of the bending mode response.

The control scheme developed will cater to nonlinearities of the plants since the controller itself is a nonlinear controller. However nonlinearities have not been simulated. Stability analysis of the overall system during adaptation and guaranteeing global stability of the system are being investigated. Study can be extended for full flight duration.

REFERENCES

[1] Kumpathi S. Narendra and Kannan Parthasarathy, Identification and control of dynamic system using neural networks: *IEEE Trans on Neural Networks* Vol. 1 no. 1, pp 4-27, march 1990.

[2] Widrow B. and Lehr M.A. : 30 years of adaptive neural networks: preception, madaline and abck propagation; *proc. IEEE* 1990, 78, pp 1415-1442.

[3] Widrow B. and S.D. Sterns, *Adaptive signal processing*, NewYork Prentice Hall, 1985.

[4] E. Rosenblatt, *Principles of Neurodynamics*, Washington D.C. Spartan Press, 1961.

[5] Howard Demuth and Mark Beale, *Neural Network Tool box for use with Matlab*, The Mathworks INC, Mass 01760, jan 1994.

[6] V. Etxebarria, Adaptive control of discrete systems using neural networks: *IEEE Proc - Control Theory Appl*; Vol. 141, No. 4. july 1994.

[7] M.M. Gupta & D.H. Rao, Dynamic neural units with applications to the control of unknown nonlinear systems, *Journal of Intelligent and Fuzzy systems*, Vol: 1, No. 1. pp 73-92, 1993.

[8] M.M. Gupta & D.H. Rao, Dynamic Neural Adaptive Control schemes for linear and nonlinear systems, *Proceedings of the American Control Conference*, San Francisco, California, June 1993 pp 1450 - 1454.

[9] Mohanlal, *Adaptive control for sattelite launch vehicle autopilot using neural networks*, M.Tech Thesis, College of Engineering, Trivandrum, Sept 1995.