

# An Inverse Optimal Control Approach for the Nonlinear System Design Using ANN

M. P. Nanda Kumar, K. Dheeraj

**Abstract**—The design of a feedback controller, so as to minimize a given performance criterion, for a general non-linear dynamical system is difficult; if not impossible. But for a large class of non-linear dynamical systems, the open loop control that minimizes a performance criterion can be obtained using calculus of variations and Pontryagin's minimum principle. In this paper, the open loop optimal trajectories, that minimizes a given performance measure, is used to train the neural network whose inputs are state variables of non-linear dynamical systems and the open loop optimal control as the desired output. This trained neural network is used as the feedback controller. In other words, attempts are made here to solve the "inverse optimal control problem" by using the state and control trajectories that are optimal in an open loop sense.

**Keywords**—Inverse Optimal Control, Radial basis function neural network, Controller Design.

## I. INTRODUCTION

NEURAL networks can be applied in two ways in the design of control systems. It can be used to obtain a mathematical model of the real system to be controlled. It can also be used to design a controller, once a model of the real system is available. If the model is an accurate representation of the real system and the controller has been designed correctly, then the controller will work on the real system. Both these tasks can be performed by neural network if properly trained [2]-[5]. Controllers are designed to meet the specification around an operating point (where linear model is valid) and then via a scheduler, a controller emerges, which can accomplish the control objectives over the whole operating range. The existing conventional theory cannot meet control demands required today. New control laws to perform novel control functions should be designed while the system is in operation. Consequently, the mathematical complexity increases. Thus an intelligent controller is required to replace the conventional controllers. This is where neural network comes into scenario of the control system design.

The ability of neural networks to deal with non-linear systems is perhaps the most important from the control theory viewpoint. The great diversity of non-linear systems is the primary reason why no generally applicable theory for non-linear control design has yet evolved. However it is the ability of neural networks to represent non-linear mappings and hence to model non-linear systems, which is the feature that can be readily exploited in the synthesis of non-linear

controllers.

The methodology used to solve the inverse optimal control problem consists of the following steps:

*Step 1.* The open loop optimal control trajectory is obtained using calculus of variations and Pontryagin's minimum principle [1]

*Step 2.* Train an RBF neural network using the input-output training set obtained above.

*Step 3.* Use the trained neural network as a feedback controller.

## II. INVERSE OPTIMAL CONTROL PROBLEM

Instead of asking for the control law corresponding to a given performance index, it is better to seek the performance criteria for which a given control law is optimal. This problem has come to be called the inverse optimal control problem; it is analogous to the older problem of the inverse problem of the calculus of variations.

We employ the inverse optimal control approach which circumvents the task of solving Hamilton-Jacobi equation and results in a controller optimal with respect to a meaningful cost functional [8]. In this approach, a stabilizing feedback control is designed first, and then it is shown that it optimizes a cost functional. The main characteristic of the inverse approach is that a cost function is a posteriori determined for the stabilizing feedback control law [9]-[11]. This approach, originated by Kalman to establish certain gain and phase margins of linear quadratic regulators [12], was introduced into nonlinear control in [13], and it was recently revived in [14] to develop a methodology for design of robust nonlinear controllers.

## III. SOLUTION TO THE OPTIMAL CONTROL PROBLEM

We consider the dynamic system

$$\dot{x} = a[x(t), u(t), t] \quad (1)$$

where  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$  denote the state and the control vectors, respectively.

The problem is to find the admissible control  $u^*$  that minimizes the performance index

$$J = h[x(t_f), t_f] + \int_{t_0}^{t_f} g[x(t), u(t), t] dt \quad (2)$$

where  $t_0$  and  $t_f$  are the initial and final time;  $h$  and  $g$  are

M.P. Nanda Kumar is with National Institute of Technology, Calicut, Kerala, India 673601(phone:+919495862820;e-mail: nandu@nitc.ac.in).

Dheeraj K. is with National Institute of Technology, Calicut, Kerala, India 673601 (phone: +917736541308; e-mail: dheerajshares@gmail.com).

scalar functions  $u^*$  is the optimal control and  $x^*$  is the optimal state trajectory. Assuming that the state and control trajectories are not constrained by any boundaries, that the final time  $t_f$  is free, we can summarize the two-point boundary value-problem that results from the variational approach by the equations [3]

$$\dot{x}^*(t) = \frac{\partial H}{\partial p} = a[x^*(t), u^*(t), t] \quad (3)$$

$$\dot{p}^*(t) = -\frac{\partial H}{\partial x} \quad (4)$$

$$= -\left\{ \frac{\partial a}{\partial x} [x^*(t), u^*(t), t] \right\}^T p^*(t) - \frac{\partial g}{\partial x} [x^*(t), u^*(t), t] \quad (5)$$

$$0 = -\frac{\partial H}{\partial u} = \left\{ \frac{\partial a}{\partial u} [\dot{x}(t), u(t), t] \right\}^T p^*(t) + \frac{\partial g}{\partial u} [\dot{x}(t), u(t), t] \quad (6)$$

$$x^*(t) = x_0 \quad (7)$$

$$p^*(t_f) = -\frac{\partial H}{\partial x} [x^*(t_f)] \quad (8)$$

Equations (3)-(6) are in general nonlinear. We assume that the above equations along with the split initial conditions in (7) and (8) can be solved by using a suitable iterative numerical procedure like steepest descent or variation of extremals. The state and control trajectories resulted from this solution are optimal in an open loop sense and is used to design the neural feedback controller. The methodology adopted is as follows:

#### IV. METHODOLOGY TO OBTAIN FEEDBACK CONTROLLER USING ANN

For a general non-linear control problem, the calculus of variation approach to solve the optimal control problem leads to a two-point boundary value problem, which can be solved using methods like steepest descent, variation of extremals etc. [6], [7]. This results in an open loop control and the corresponding trajectories. Using these control and state trajectories training data are generated to train the radial basis neural network, whose inputs are the state trajectories and the target is the open loop optimal control trajectory. This neural network is then used as the feedback controller to control the non-linear dynamic system. This methodology is explained in Fig. 1. The structure of radial basis function neural network (RBFNN) is shown in Fig. 2. The RBFNN consists of three layers, an input layer, a hidden layer and an output layer. Hidden units are known as radial centres. Each radial centre is represented by a vector  $C_i$ ,  $i = 1, \dots, L$ , where  $L$  is the number of radial centres in the hidden layer. The transformation from the input space to the hidden unit space is nonlinear whereas

the transformation from the hidden unit space to output space is linear. The dimension of each centre for 2 input network is  $2 \times 1$ . The training of an RBFNN requires optimal selection of the centre  $C_i$  and the weights  $W_i$ ,  $i = 1, \dots, L$ . There are many approaches to update centres and weights namely, pseudo-inverse approach and gradient-descent approach etc. In this work, gradient-descent algorithm is used to update the centres and weights. The number of input neurons is 2, hidden neurons are 100 and output neurons are 1.

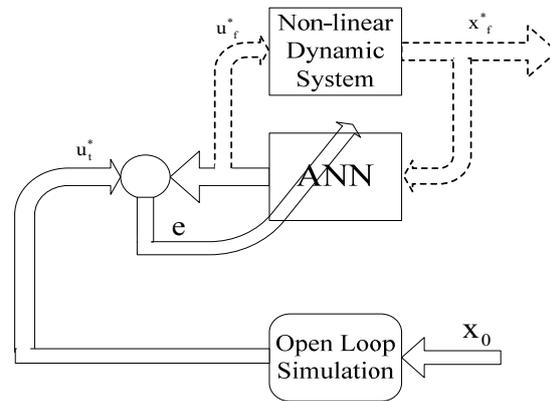


Fig. 1 ANN as the feedback controller

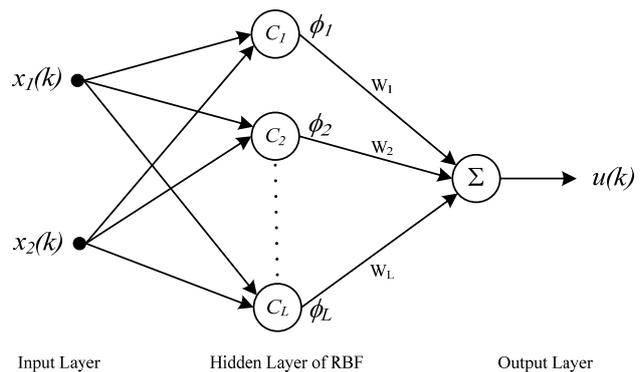


Fig. 2 A 2 input and 1 output RBF network

#### V. NUMERICAL EXAMPLE

The continuous stirred tank chemical reactor with the following state equations is considered.

$$\dot{x}_1(t) = -2[x_1(t) + 0.25] + [x_2(t) + 0.5] e^{\frac{25x_1(t)}{x_2(t)+2}} - [x_1(t) + 0.25]u(t) \quad (9)$$

$$\dot{x}_2(t) = 0.5 - x_2(t) - [x_2(t) + 0.5] e^{\frac{25x_1(t)}{x_2(t)+2}} \quad (10)$$

where  $x_1(t)$  is the normalized value of deviation in temperature from the steady state value,  $x_2(t)$  is the normalized value of the deviation in concentration from the steady state value and  $u(t)$

is the normalized control which refers to the rate of removal of heat by coolant from the reactor. The performance measure is

$$J(u) = \int_0^{0.78} [x_1^2 + x_2^2 + Ru^2(t)] dt \quad (11)$$

indicating that the desired objective is to maintain the temperature and concentration close to their steady-state values without expending large amounts of control effort. R is the weighting factor, which we shall select (arbitrarily) as 0.1. The co-state equations are determined from the Hamiltonian,

$$\begin{aligned} H[x(t), u(t), p(t)] = & x_1^2(t) + x_2^2(t) + Ru^2(t) + p_1(t)[-2x_1(t) + 0.25] \\ & + (x_2(t) + 0.5)e^{\frac{25x_1(t)}{x_1(t)+2}} - [x_1(t) + 0.25]u(t) \\ & + p_2(t)[0.5 - x_2(t) - [x_2(t) + 0.5]e^{\frac{25x_1(t)}{x_1(t)+2}}] \end{aligned} \quad (12)$$

$$\begin{aligned} \dot{p}_1 = -\frac{\partial H}{\partial x_1} = & -2x_1 + 2p_1(t) - p_1(t)[x_2(t) + 0.5] \left[ \frac{50}{[x_1(t) + 2]^2} \right] e^{\left( \frac{25x_1(t)}{x_1(t)+2} \right)} \\ & - p_1(t)u(t) + p_2(t)[x_2(t) + 0.5] \left[ \frac{50}{[x_1(t) + 2]^2} \right] e^{\left( \frac{25x_1(t)}{x_1(t)+2} \right)} \end{aligned} \quad (13)$$

$$\begin{aligned} \dot{p}_2 = -\frac{\partial H}{\partial x_2} = & -2x_2(t) - p_1(t)e^{\left( \frac{25x_1(t)}{x_1(t)+2} \right)} + p_2(t) \left[ 1 + e^{\left( \frac{25x_1(t)}{x_1(t)+2} \right)} \right] \end{aligned} \quad (14)$$

The algebraic equation that must be satisfied is

$$\frac{\partial H}{\partial u} = 2Ru(t) - p_1(t)[x_1(t) + 0.25] = 0 \quad (15)$$

The guessed initial conditions for solving the co-state equations are  $p_1(0)=2$  and  $p_2(0)=2$ . The desired final values of co-state are  $p_1(0.78)=0$  and  $p_2(0.78)=0$ . The influence function matrix differential equations are

$$\dot{p}_x(t) = \begin{bmatrix} a & \alpha \\ b & -1-\alpha \end{bmatrix} p_x(t) + \begin{bmatrix} c & 0 \\ 0 & 0 \end{bmatrix} p_p(t) \quad (16)$$

$$\dot{p}_p(t) = (-2 + p_2(t) - p_1(t)) \begin{bmatrix} d & \alpha \\ e & -2 \end{bmatrix} p_x(t) + \begin{bmatrix} f & g \\ -\alpha & 1+\alpha \end{bmatrix} p_p(t) \quad (17)$$

where  $\alpha = e^{\left( \frac{25x_1(t)}{x_1(t)+2} \right)}$

$$\begin{aligned} a = & -2 + \frac{50[x_2(t) + 0.5]}{[x_1^2(t) + 2]^2} - \frac{[x_1(t) + 0.5]p_1(t)}{R} \\ b = & \frac{50[x_2(t) + 0.5]}{[x_1^2(t) + 2]^2} \alpha \\ c = & \frac{-[x_1(t) + 0.25]^2}{2R} \\ d = & \frac{100[23 - x_1(t)][x_2(t) + 0.5]}{[x_1(t) + 2]^4} \alpha + \frac{p_1^2(t)}{2R} \\ e = & \frac{50[p_2(t) - p_1(t)]}{[x_1^2(t) + 2]^2} \alpha \\ f = & -2 - \frac{50[x_2(t) + 0.5]}{[x_1^2(t) + 2]^2} \alpha + \frac{[x_1(t) + 0.5]p_1(t)}{R} \\ g = & \frac{50[x_2(t) + 0.5]}{[x_1^2(t) + 2]^2} \alpha \end{aligned}$$

Equations (9), (10), (13)-(17) are solved using the boundary conditions  $P_x(0)=0$  and  $P_p(0)=I$ . The state and costate values appearing in (16) and (17) are obtained by integrating the reduced state and costate equations (9), (10), (13) and (14) with the initial conditions  $x(0)=[0.05 \ 0.00]^T$  and  $p(0)=P^i(0)$ . After integrating (9), (10), (13), (14), (16) and (17) from  $t=0$  to  $t=0.78$ , the matrix  $P_p(0.78)$  is used to determine  $P^{i+1}(0)$  as

$$P^{(i+1)}(0) = P^{(i)}(0) - [P_p(0.78)]_i^{-1} P^i(0.78) \quad (18)$$

The initial guess used to start the iterative procedure is

$$P^{(0)}(0) = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} \quad \text{and} \quad (19)$$

$$|p_1(0.78)| + |p_2(0.78)| \leq 10^{-5} \quad (20)$$

is the stopping criterion used. The method converged after 35 iterations to yield the initial co-state values

$$P_1(0)=1.0782 \quad \text{and} \quad P_2(0)=0.1918$$

The minimum value of the performance measure obtained is  $J^* = 0.0266$ . The optimal state trajectories are plotted in Fig. 3. Fig. 4 shows the performance index minimized.

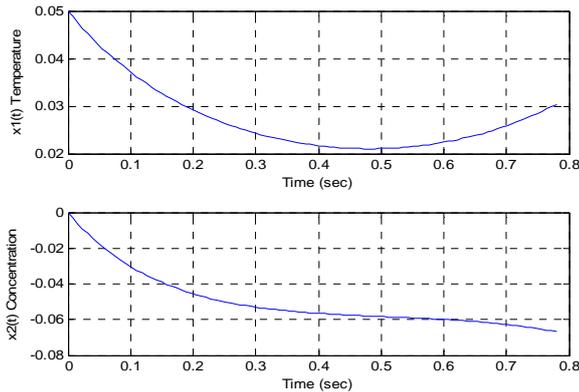


Fig. 3 The open loop (optimal) state trajectories

An RBFNN has been trained using the trajectories resulted from the above open loop optimal control problem. The values of the optimal state trajectories  $x_1(t)$  and  $x_2(t)$  are used as the input training samples  $x_1(k)$  and  $x_2(k)$ ,  $k = 0, 1, \dots, 100$ . The desired values of the output is derived from the optimal control trajectory  $u(t)$  as  $u(k)$ ,  $k = 0, 1, 2, \dots, 100$ . The network has been trained using the MATLAB software [4] and obtained the convergence after adding 100 neurons in the hidden layer.

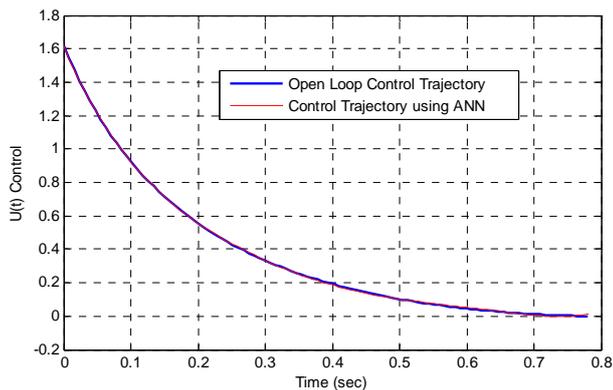


Fig. 4 The control trajectories

The trained neural network is used as the optimal feedback controller as shown in the Fig. 1. The resulting comparisons of the state and control trajectories are given in Figs. 3 and 4. The same is tested adding 10% zero mean random noise (ZMRN) of the initial value of the states yielded satisfactory results shown in Fig. 5.

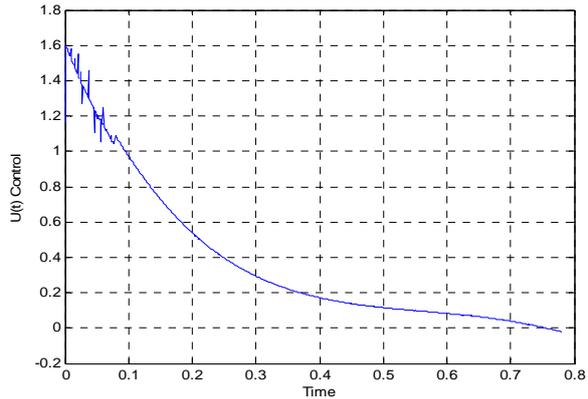


Fig. 5 The control trajectory with 10% ZMRN

## VI. CONCLUSION

The design of a controller for a general non-linear dynamical system is considered. For many control problems the open loop control that optimizes a performance index can be obtained rather easily using numerical integration. The main characteristics of the methodology is in the fact that the cost function for the control problem is obtained posteriori by solving the open loop optimal control problem using calculus of variations and Pontryagin's minimum principle. The optimal trajectories so obtained are used to train the neural network and is then used as the neural feedback controller. Furthermore, the methodology used is free from the hard task of obtaining the solution of the Hamilton Jacobi Bellman (HJB) Equation. From the stimulation studies it is clear that such feedback controller is feasible and it is also shown that such a controller is robust.

## REFERENCES

- [1] Donald E. Kirk, *Optimal Control Theory an Introduction*, 2nd ed, Dover Publication, Inc., Mineola, N. Y., USA, 2004.
- [2] K. S. Narendra and Kannan Parthasarathy, "Identification and Control of Dynamic Systems Using Neural Networks", *IEEE Transactions on Neural Networks*, vol. 1, March 1990
- [3] Madan M. Gupta and Dandina H. Rao, "Neuro Control Systems – Theory and Applications, IEEE press – A Select Reprint Volume, 1994.
- [4] Mark Hudson Beale, Martin T. Hagan and Howard B. Demuth, "Neural Network Toolbox, User's Guide, September 2012, The MathWorks, Inc.
- [5] Paul J Webrose, "An Overview of Neural Networks for Control" *IEEE Transactions on Control Systems*, pp. 40-42, January 1991.
- [6] Simon Haykin, *Neural Networks – A Comprehensive Foundation*, IEEE Press 1994.
- [7] Steepen W. Riche, "Steepest Descent Algorithm for Neural Network Controllers and Filters" *IEEE Transactions on Neural Networks*, vol. 5, no. 2, March 1994.
- [8] M. Krstic, I. Kanellakopoulos and P. Kokotovic, "Nonlinear and Adaptive Control Design", John Wiley and Sons, New York, USA, 1995.
- [9] K. D. Do, Z. P. Jiang, and J. Pan, "Simultaneous tracking and stabilization of mobile robots: An adaptive approach", *IEEE Transactions on Automatic Control*, vol. 49, pp. 1147–1152, July, 2004.
- [10] L. A. Feldkamp, D. V. Prokhorov and T. M. Feldkamp, "Simple and conditioned adaptive behavior from Kalman filter trained recurrent networks", *Neural Networks*, vol. 16, pp. 683–689, 2003.
- [11] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots", *IEEE Transactions on Control System Technology*, vol. 18, pp. 1199-1206, September, 2010.

- [12] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
- [13] P. J. Moylan and B. D. O. Anderson, "Nonlinear regulator theory and an inverse optimal control problem," *IEEE Trans. Automat. Contr.*, vol. 18, pp. 460–465, 1973.
- [14] R. A. Freeman and P. Kokotović, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Boston, MA: Birkhäuser, 1996.