2002년도 대한전기학회 하계학술대회 논문집 2002.7.10-12

신경망을 이용한 비선형 플랜트 최적제어에 관한 연구

*Lin Min(1), Zhao XiaoBing(1), 조현섭(2), 박왈서(3)
내몽고사범대학교(1), 청운대학교(2), 원광대학교(3)

An Optimized Controller for Nonlinear Plant Based on Neural Network

Zhao XiaoBing (1), Lin Min(1), Hyeon-Seob Cho(2), Wal-Seo Park(3)
(1)Computer Department, Inner Mongolia Normal University, Huhhot, Inner Mongolia, China, (2)Chungwoon University, (3)Wonkwang University

Abstract - Design of controller of nonlinear systems is an important part of control research.In this paper, a controller for nonlinear plants using a neural network is presented. The controller is a combination of an approximate PID controller and a neural network controller.The PID controller be used for stabilizing the process and for compensating for possible disturbances , a neural network act as feedforward controller. In this method, a RBF neural network is trained and the system has a stable performance for the inputs it has been trained for. Simulation results show that it is very effective and can realize a satisfactory control of the nonlinear system and meets the demands of the system.

Keywords RBF neural network,nonlinear system, PID controller

1. Introduction:

The control of systems with complex,unknown,and nonlinear dynamics has become a topic of considerable importance in control research. The conventional design methods of a control system often require the construction of a mathematical model describing the dynamic behavior of the plant to be controlled.When such a mathematical model is difficult to obtain due to uncertainty or complexity of systems,these conventional techniques based on a mathematical model are not well suited for dealing with. Neural networks have massive parallelism and ability to approximate arbitrary nonlinear mappings.So artifical neural network teachiques have been suggested for identification and control of nonlinear plants for which conventional techniques of control do not give satisfactory performance, such as the accuracy in matching the behavior of the physical system.Inverse models of dynamical systems play a crucial role in a range of control structures.Conceptually the simplest approach is direct inverse control.Assuming that the discrete-time models of a nonlinear system to be controlled can be described by

$$y(t+1) = f[y(t), y(t-1), \ldots, y(t-n+1), u(t), u(t-1), \ldots u(t-m+1)]$$

Then a neural network is trained as the inverse model of the nonlinear plant:

$$\hat{u}(t) = f^{-1}[y(t+1), y(t), \ldots, y(t-n+1), u(t-1), \ldots, u(t-m+1)]$$

This inverse model can then be used as feedforward controller for the plant by replacing y(t+1) with the desired output(the reference r(t+1)).

$$\hat{u}(t) = f^{-1}[r(t+1), y(t), \ldots, y(t-n+1), u(t-1), \ldots, u(t-m+1)]$$

There are two train strategies for the inverse model are available:generalized training and specialized training.In generalized training a network is trained off-line to minimize the following criterion(w specifies the weights in the network) with Levenberg-Marquardt method.

$$J_1(w) = \sum_{t=1}^{N} (u(t) - \hat{u}(t))^2$$

Specialized training is an on-line procedure related to model-reference adaptive control.The idea is to minimize the criterion:

$$J_2(w) = \sum_{t} (y_m(t) - y(t))^2$$

Specialized training is often said to be goal directed because it, as opposed to generalized training,attempts to train the network so that the output of the process follows the reference closely.For this reason,specialized training is particulary well-suited for optimizing the controller for a prescribed reference trajectory.Specialized training must be performed on-line and thus it is much more difficult to carry out in practice than generalized training.Before the actural training of the inverse model is initiated,a forward model of the plant must be trained since this is required by the scheme.The specialized inverse learning structure is shown in Figure 1.The error may then be propagated back through the forward model and then the inverse model;only the inverse network model weight are adjusted during this procedure.

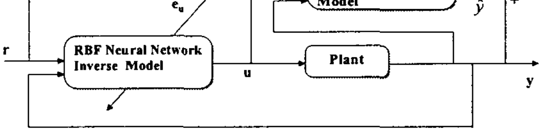

Reference model
y_m
RBF Neural Network Forward Model
e
2
-
+
$\hat{y}$
e_u
r
RBF Neural Network Inverse Model
u
Plant
y

Figure 1. The structure of specialized training

Using inverse models for feedback control leads to a dead-beat type of control which is unsuitable in many cases.If a PID controller has already been tuned for stabilizing the process,an inverse model can be used for providing a feedforward signal directly from the reference. Through combining the two controller, the PID controller is used for stabilizing the process and suppressing disturbances while the feedforward controller is used for providing a fast tracking of the reference.

2.Controller Design and RBF Network Train

The structure of control system is shown in Figure 2. The feedback controller can be a PID or any other conventional controller. The neural network employed in this scheme is an Radial Basis Function Network (RBFN). It produces a feedforward signal directly from the reference to optimize the control system.The structure of RBFN is showed in Figure 3. It is a network with two layers. A hidden layer of radial basis neurons and an an output layer of linear neurons.A common choice for the basis function is a Gaussion given by the equation: $G_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma^2}\right), i = 1,2,...m$

Where ci represents the center of the basis function and σ denotes its width.The norm $\|\cdot\|$ in equation can be expressed by Euclidean distance. The weights and biases of each neuron in the hidden layer define the position and width of a radial basis function. Each linear output neuron forms a weighted sum of these radial basis functions.With the correct weight and bias values for each layer,and enough hidden neurons,a RBFN can fit any function with any desired accuracy.The advantage of the RBFN is its rapid learning, generality and simplicity. RBFN finds the input to output map using local approximators.It can be trained faster than BP network and have none of BPs training problems such as saturation and local minima.

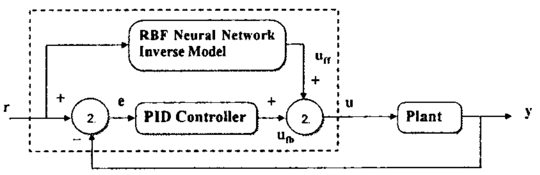


Figure 2. The structure of the control system

One of the simplest training algorithms for an RBF network is to have as many hidden units as there are training samples. The width parameter is set, heuristically, to a number between the minimum distance between points in the training set and the maximum distance between points in the training set, Since the mapping from the hidden units to the output is a linear function.

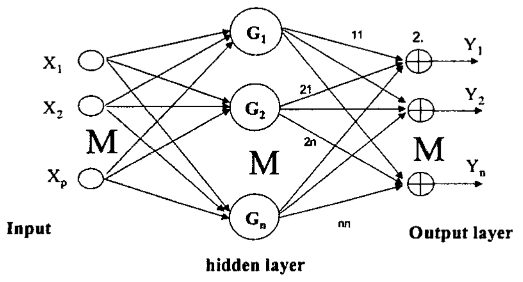


Figure 3. Radial Basis Function Network

The interpolation matrix is used where every row corresponds to the response in the hidden layer for each observation and every column to one hidden unit. This solution is exact, i.e. $\hat{y}(n) = y(n)$,for all observations in the training data set.The exact RBF training algorithm is available in MATLAB in the function NEWRBE. The exact RBF training algorithm is of course not a good choice if we have noise in our data. In this case we know (almost) for certain that the output values y(n) we have in the training data are incorrect, and we do not want to construct a model that reproduces these values exactly. Instead, we want a noisy interpolation model like the Generalized Regression Neural Network (GRNN).The MATLAB function NEWGRNN implements the GRNN algorithm,which is a normalized version of the RBF network. This normalization corresponds to a special form of kernel interpolation. Using all the training examples as centers is simple and produces an exact solution (i.e. a solution with zero training error, if this is desired). However,this produces very large networks. It is often possible to achieve almost as good training results as the exact solution, but using only a small subset of the training data. Furthermore, these smaller networks tend to produce better test results than the exact training solution.There are different ways to select this small subset of the training data. One is to choose a random subset of the training data, another is to iteratively select the observation that minimizes the training error the most. The latter method is effectively done using the orthogonal least squares technique.A random subsets of the training data is very simple:Instead of using all the training input data as center points, we select a small subset of them as center points. The main reason for this is to lower the number of internal units in the network, e.g. for real-time operation.Radial basis functions are often referred to as local mappings. This means that typically only a few of the hidden units will respond when a new input is presented to the network. Each hidden unit only has a local area in input space within which it responds. For this reason, it usually requires quite many units to construct a good mapping.On the other hand,this also means that two hidden units seldom interfere with each other when an RBF net is trained.

Taking an overlapping Gaussian activation function for kernel units supposedly provides a smoother response and better generalization.but in our case the amount of interference was so high and we obtained a better performance with non-overlapping regions. During the training stage,each time only one kernel unit responds and one weight is adjusted,This results in a shorter training time compared with Multi-layer Perceptron (MLP) type networks. Because this method does not need prior knowledge about the transfer characteristics of the computing devices,it is not affected by the effects of neuron to neuron variations. Generally optimization methods based on parameter perturbation are bound to failure when many parameters are involved in perturbation and that is because of the moving target effect of the other parameters.It is not the case in the proposed system for which only one or few related parameters are active at each time.

3.Simulation Study

We consider a nonlinear dynamic plant which is governed by the difference equation:

$$y(t+1) = f[y(t), y(t-1), u(t), u(t-1), u(t-2)]$$

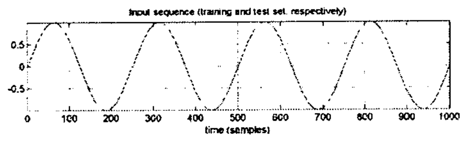
with an arbitrary unknown function:

$$f[.] = 1 + \cos\{6\pi[y^2(t) + y^2(t-1)]\} + e^{-u(t)} + u^2(t-1) + u^2(t-2).$$

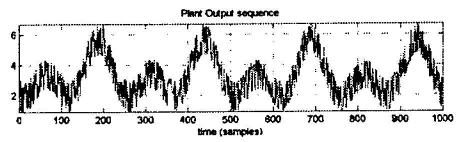
The input to the nonlinear system is r(t)=sin(2πt/250) in the interval [-1,1].The simulation results obtained for this case are shown in Figure 4. Figure 4a show the input sequence (a half for training the inverse mode and a half for testing it,respectively). Figure 4b show plant corresponding responses. Figure 4c show the correlation function of error response and Figure 4d show the actual control signal (u) and the prediction signal (uhat) as well as the error signal(u-uhat). Figure 4 (a) The input to the nonlinear system r(t). (b) Plant output.(c)Correlation function of error response. (d)Actual control signal (u) , prediction (uhat) and error signal(u-uhat).

4. Conclusions

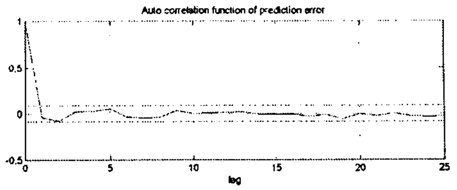
A PID feedback controller for nonlinear plants combined with neural feedforward controller optimized the existing control system,which is guaranteed to perform stably for the inputs that it has been trained for. The RBF network differ from classical multi-layer perceptrons.It have more effective ability to learning. Using this method of control makes it different from conventional control methodologies. It makes a structure beyond the capabilities of backpropagation based on neural networks.The structure used here decreased in training time.It can also be viewed as a gain scheduling adaptive controller which can work for any unknown plant with no attempt to linearize the system at each region.



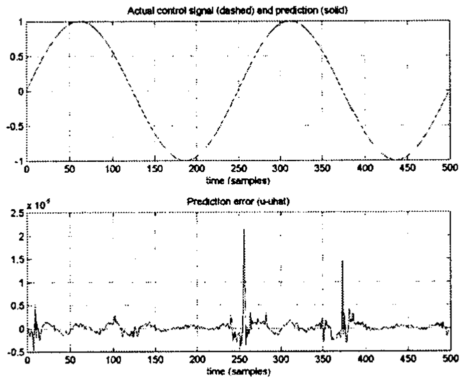
(a)



(b)



(c)



(d)

References

[1]J.Moody and C.J.Darken, Fast learning in Networks of Locally-Tuned Processing Units ,Neural Computation, Vol.1,pp.281-194,1989.

[2]K.J.Hunt,D.Sbarbaro,R.Zbikowski,P.J.Gawthrop, Neural Networks for Control Systems-A Survey ,Automatica, Vol.28,No.6,pp.1083-1112,1992.

[3]K.S.Narendra and K.Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks , IEEE Trans on Neural Network Vol.1 No.1,pp.4-27,1990

[4]M.M.Gupta and D.H.Rao, Dynamic Neural Units in the Control of Linear and Nonlinear Systems. ,In Proceedings of the International Joint Conf. On Neural Networks, pp.100-105, June 1992.

[5]G.A.Montague,M.J.Willis and A.J.Morris, Artificial Neural Network Model Based Control ,Automatic Control Conference,1994

[6]Howard Demuth,Mark Beale, Neural Network ToolBox for Use with MATLAB The MathWorks Inc.,1994

[7]Chen S,Billing S A, Grant P M.Recursive Hybrid Algorithm for Nonlinear System Identification using Radial Function Network ,Int.J Control,1992,55(5): 1051-1070

[8]O.Sorensen(1994): Neural Networks in Control Applications ,Ph.D.Thesis.Aalborg University,Department of Control Engineering,1994.