Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 1, 2004

Neural-Network-Based Control

Luminita Giurgiu, Stefan Demeter, Feteanu Gigi[1]

Abstract - To make a system behave in a desired manner means to control it. The desired behavior determines two kinds of problems: servo problems where the desired behavior is to make the output of the system follow a reference trajectory and regulation problems where the output must be kept at a certain constant level. Neural networks are proposed as a tool for adaptive control of nonlinear dynamic system in direct control system design where the controller is a neural network and indirect control where the network only models the system to be controlled. This article follows aspects of the direct control system design.
Keywords: neural network, direct control systems

I. INTRODUCTION

The approach to the problem of control systems for processes that are nonlinear and difficult to model in a deductive fashion, has been the system identification approach to model based control of an unknown process:

- System identification to infer a neural network model of the process to be controlled from a set of data collected in an experiment with the process;
- One or more controllers are designed based on the identified model.

When the controller is a neural network we speak about direct control system design witch is often advantageous in real time platforms that prohibits complicated solutions. The implementation of such a controller is simple but the design and tuning implies the retraining of the network every time a design parameter is modified. The design is model-based because the model of the system is required in order to design the controller.

II. DIRECT INVERSE CONTROL

The neural network based controller use the inverse of the process as the controller. This concept is called direct inverse control and the principle consists in training a network to act as the inverse of the system and use this network as a controller.
If the system is described by:

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

a network is trained as the inverse of the process:

$$\hat{u}(t) = \hat{g}^{-1}[y(t+1), y(t),\ldots, y(t-n+1), u(t),\ldots, u(t-m)] \quad (2)$$

Assuming such a network obtained, the inverse model is subsequently applied as the controller for the process by substituting the output at time $t+1$ by

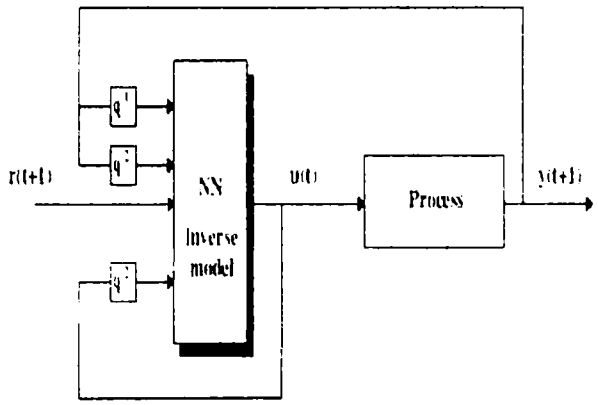

Fig. 1 Principle of direct inverse control

the desired output, $r(t+1)$, the reference. If the obtained network represents the inverse, the control input produced by it will drive the output of the system at time $t+1$ to $r(t+1)$.
An inverse model must be trained and two strategies for establishing the inverse model will be described:

1. General training where a network is trained off-line to minimize the following criterion (θ specifies the weights in the network):

$$J_1(\theta) = \sum_{t=1}^{N} [u(t) - \hat{u}(t,\theta)]^2 \quad (3)$$

2. Specialized training where a network is trained on-line to minimize the following criterion:

$$J_2(\theta) = \sum_{t=1}^{N} [r(t) - y(t)]^2 \quad (4)$$

III. GENERAL TRAINING

The way of training a network as the inverse of a system can be considered as a system identification problem: first an experiment is performed, second a

[1] Land Forces Academy "Nicolae Balcescu", Revolutiei 3-5, Sibiu, email: lumigee@actrus.ro

network architecture is selected and third the network is trained off-line.
The main difference from system identification consists in the choice of regressors and network output. The training process is executed in order to minimize the criterion:

$$J(\theta, Z^N) = \frac{1}{2N}\sum_{t=1}^{N}[u(t) - \hat{u}(t|\theta)]^2 \qquad (5)$$

The illustration of he method is made on the following benchmark system with smooth nonlinearities:

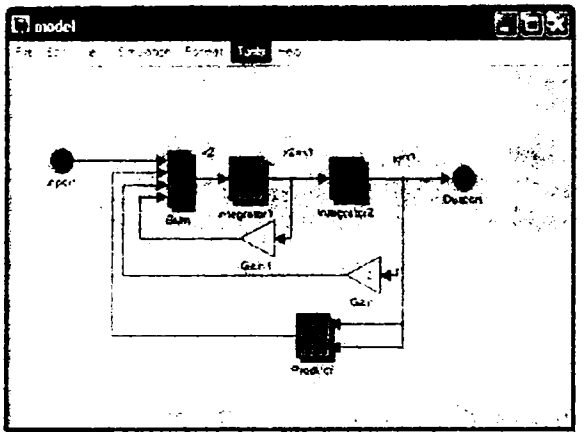

Fig. 2 The benchmark system

During the initial experiment, the input signal is applied to the benchmark system and the response produced by the system is obtained. A specific function generates a level change at random instances signal. The signal is defined in the following way:
u(0)=randn, for t>0:

$$u(t) = \begin{cases} u(t-1) & \text{with probability } \alpha \\ \text{randn} & \text{with propability } 1-\alpha \end{cases}$$

At random instances the signal is changed to a new random value.

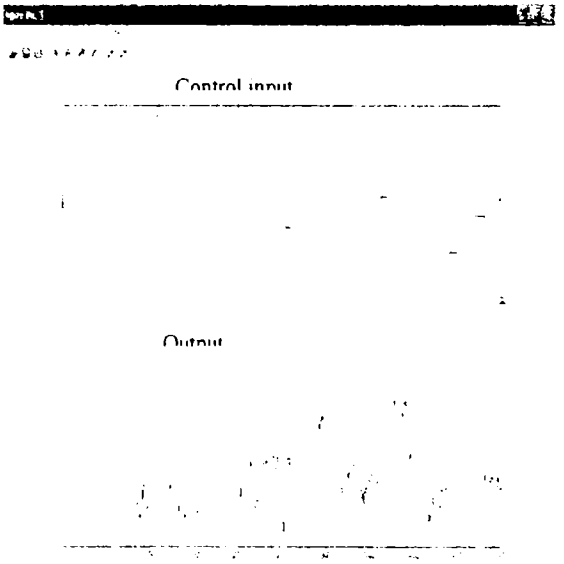

Fig. 3 Training set

The "shift frequency" is determined by 'alpha' which is a variable between 0 and 1: alpha=0 generates a constant signal while alpha=1 generates a Gaussian white noise signal. The signal is scaled so that it contains values between two desired values. The data set of corresponding iputs and outputs:

$$Z^N = \{[u(t), y(t)],\ T = 1, \ldots, N\} \qquad (6)$$

On this data set, a neural network with five *tanh*-hidden neurons and a linear output will be trained as the inverse model. The Levenberg-Marquardt algorithm is used and the following regressor structure:

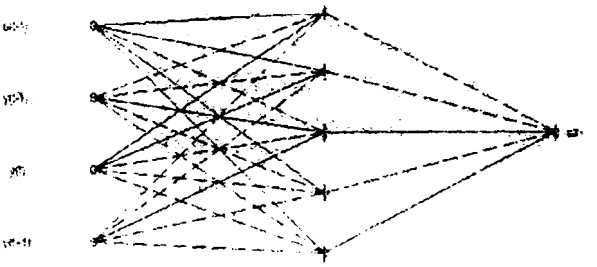

Fig. 4 Regressors structure

The obtained network will be used as controller. Figure 5 shows the response of the closed-loop system to different reference signal's changes. From the plot, we observe a small overshoot when the reference changes, indicating that the inverse model is not perfect, and another effect is that the control signal is extremely active and assume very large values.

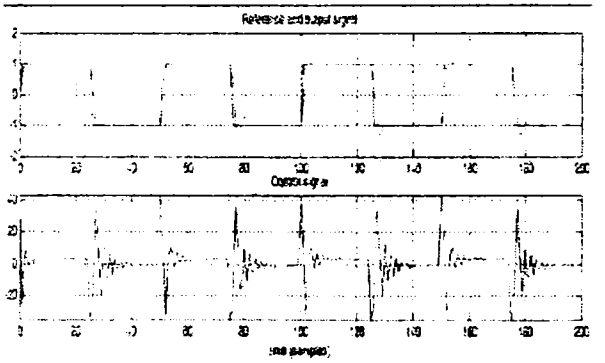

Fig.5 Direct inverse control: panel superior-reference and output signal, panel inferior-control signal

It is obviously that in terms of frequency content and amplitude size, the control signal and the signal applied to the system during the experiment have very different shapes since the inverse model is extrapolating in the transient periods.
The general training criterion that expresses the objective to minimize the difference between network output and a sequence of true control inputs is not really relevant, so the trained inverse model should be validated in terms of performance of the final closed-loop system and the goal is that the system output follows the reference closely. The

minimizing criterion will be goal directed and has the type expressed in the relation (5).
To fine-tune the network directly on the square wave reference trajectory will be used the second method of specialized training.
Considering that the system output, y(t), depends on the output of the inverse model, u(t-1), the minimization of this criterion is very difficult to be off-line considered. In the on-line approach the network is trained to minimize:

$$J_t(\theta, Z^N) = J_{t-1}(\theta, Z^{t-1})[r(t) - y(t)]^2 \qquad (7)$$

A recursive gradient method is considered and assuming that J_{t-1} has already been minimized, the weights are adjusted according to:

$$\hat{\theta}(t) = \hat{\theta}(t-1) - \mu \frac{de^2(t)}{d\theta} \qquad (8)$$

where $e(t)=r(t)-y(t)$ and:

$$\frac{de^2(t)}{d\theta} = -\frac{dy(t)}{d\theta} e(t) \qquad (9)$$

The gradient $\frac{dy(t)}{d\theta}$ will be calculated by:

$$\frac{dy(t)}{d\theta} = -\frac{\partial y(t)}{\partial u(t-1)} \frac{du(t-1)}{d\theta} = \frac{\partial y(t)}{\partial u(t-1)} \times$$
$$\left[\frac{\partial u(t-1)}{d\theta} + \sum_{i=1}^{n} \frac{\partial u(t-1) dy(t-i)}{\partial y(t-i) d\theta} + \sum_{i=1}^{n} \frac{\partial u(t-1) du(t-i)}{\partial u(t-i) d\theta} \right] \qquad (10)$$

Since that the Jacobians of the system $\frac{\partial y(t)}{\partial u(t-1)}$ are required, a forward model of the system is identified to provide the Jacobians $\frac{\partial y(t)}{\partial u(t-1)} \cong \frac{\partial \hat{y}(t)}{\partial u(t-1)}$.

The forward model is obtained by system identification. The model is inferred from the same data set as was used for training the inverse model.
The specialized training considers a recursive Gauss-Newton algorithm in updating the weights of the network. While updating the weights the reference trajectory is applied.
When training is finished, the entirely closed-loop system is simulated again. Figure 8 shows the final controller performance. It seems from response that perfect model following, was achieved with specialized training.
The characteristics of direct inverse control can be expressed in few advantages: intuitive simple, simple to implement, the controller can be optimized for a specific reference trajectory with specialized training and a few disadvantages: problems for systems not being one-to-one, doesn't work for system with unstable inverse, generally presents high

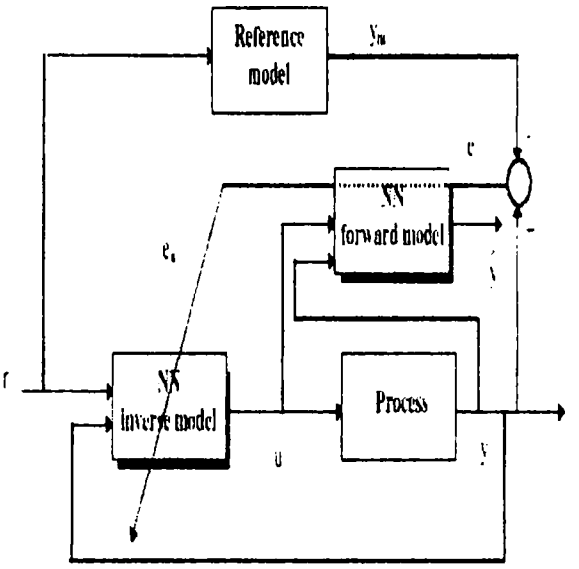

Fig. 6 Specialized training principle

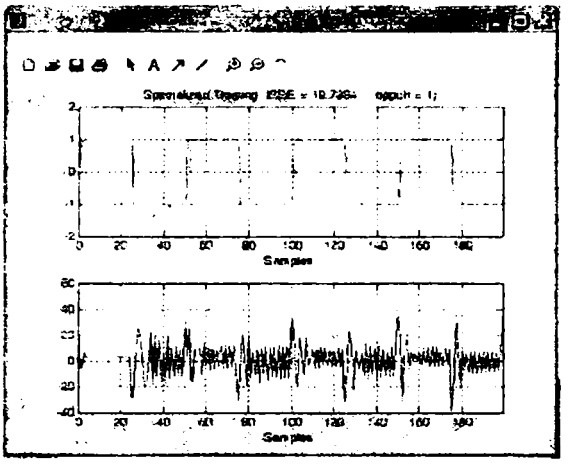

Fig. 7 Specialized training process: epoch 1

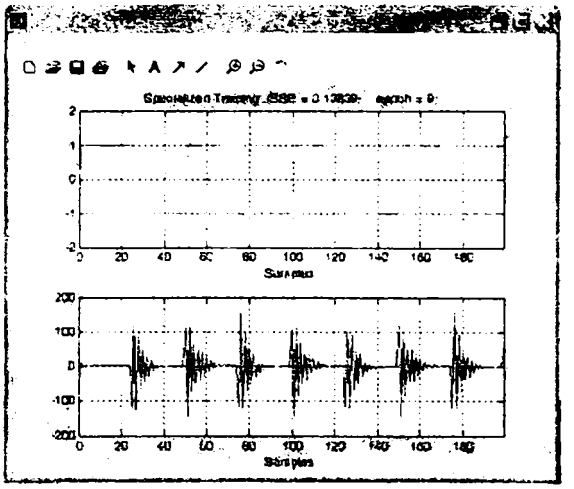

Fig. 8 Specialized training process: epoch 9

sensitivity to disturbances and noise.

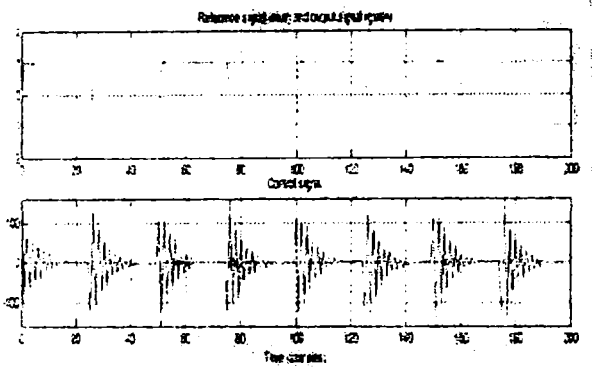

Fig. 9 Final controller performance

IV. CONCLUSIONS

Direct inverse control with neural networks is concretized in a scheme characterized by the controller being a network trained as the inverse of the system. Notice that the transfer function for the closed-loop system consisting of control and system equaled the time delay of the system. The network training lies on two different methods. First generalized training where the network is trained off-line to minimize the mean square error between a control signal applied to the system in an initial experiment and the control signal produced by the neural network and second specialized training where the objective is to minimize the mean square error between reference signal and the output of the system. The specialized training is done in an on-line manner using a recursive training algorithm.

It was concluded that generalized training should be used mainly for initialization the network. The obtained network is subsequently fine-tuned with specialized training. It is also emphasized that a carefully prepared and well-performed experiment for collecting data is of vital importance and a smooth low order model will frequently result in a superior closed loop performance.

REFERENCES

[1] Lennart Ljung, System Identification For Use with MATLAB, The Mathworks,Inc, 1997

[2] M. Norgaard, O, Ravn, Neural Networks for Modelling and Control of Dynamic Systems,Springer-Verlag London Berlin Heidelberg, 2003

[3] Magnus Norgaard, Neural Network Based System Identification Toolbox for use with MatLab (version 2), Technical Report 00-E-891, Department of Automation Technical University of Denmark, 2000

[4] Magnus Norgaard, Neural Network Based Control System Design Toolkit for use with MatLab (version 2), Technical Report 00-E-892, Department of Automation Technical University of Denmark, 2000

[5] D, Zaharie , Curs de retele neuronale artificiale, http://www.math.uvt.ro, 2003

[6] D, Nastac, Retele neuronale artificiale, Editura Printech, Bucuresti, 2002

[7] D, Dumitrescu, Principiile inteligentei artificiale, Editura Albastra, Cluj Napoca, 2003.