ISA Transactions 49 (2010) 510–518

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Stable modeling based control methods using a new RBF network

Selami Beyhan *, Musa Alcı [1]

Ege University, Electrical and Electronics Engineering, Bornova, Izmir, Turkey

ARTICLE INFO

Article history:
Received 1 October 2009
Received in revised form
26 February 2010
Accepted 13 April 2010
Available online 14 May 2010

Keywords:
Stable modeling/identification
ALR
RBF network
ARX modeling
Dynamic and static mapping
Inverse control
Sliding-mode control
PID control
Real-time control

ABSTRACT

This paper presents a novel model with radial basis functions (RBFs), which is applied successively for online stable identification and control of nonlinear discrete-time systems. First, the proposed model is utilized for direct inverse modeling of the plant to generate the control input where it is assumed that inverse plant dynamics exist. Second, it is employed for system identification to generate a sliding-mode control input. Finally, the network is employed to tune PID (proportional + integrative + derivative) controller parameters automatically. The adaptive learning rate (ALR), which is employed in the gradient descent (GD) method, provides the global convergence of the modeling errors. Using the Lyapunov stability approach, the boundedness of the tracking errors and the system parameters are shown both theoretically and in real time. To show the superiority of the new model with RBFs, its tracking results are compared with the results of a conventional sigmoidal multi-layer perceptron (MLP) neural network and the new model with sigmoid activation functions. To see the real-time capability of the new model, the proposed network is employed for online identification and control of a cascaded parallel two-tank liquid-level system. Even though there exist large disturbances, the proposed model with RBFs generates a suitable control input to track the reference signal better than other methods in both simulations and real time.

1. Introduction

To identify and control nonlinear systems accurately, there is need to employ fine artificial models [1]. The most commonly applied methods are neural networks (NNs) and fuzzy logic systems (FLSs). The known supports of these methods are their ability to learn and good performance for the approximation of the nonlinear functions. Feedforward NNs and FLSs perform highly nonlinear static mapping. However, for linear or mildly nonlinear systems, these models are not well suited, and they cause less accurate results of identification [2]. To extract the dynamics, we need to use models that combine linear and nonlinear models. In the development of NNs, numerous new static and dynamic types of NN, local and global recurrences, and other mixed structures have been developed to get better identification results [3]. To capture the change in operating conditions and noise disturbances are also important tasks of the identification. Therefore, to work online is actually a challenging problem for strongly nonlinear systems. In this work, online identification is aimed for, so that there is no explicit learning phase needed; i.e. the network is utilized for a learning-while-functioning task, instead of learning then functioning [4]. In system identification, some nonlinearity is not modeled exactly by feedforward networks. These unmodeled dynamics of nonlinear systems cause parameter drifts and even instability. To guarantee stability of identification and even fine parameter convergence, the optimization of parameters must be modified. Thus, the Lyapunov stability guaranteed learning rate is employed in the online gradient descent algorithm. This time-varying adaptive learning rate (ALR) is determined with current sensitivity of the plant input–output data and model structure. In all simulations, the identification is performed in a stable sense by assuming that the identified system is originally stable. This optimization provides parametric stability of the network. This means that the parameters do not increase or decrease abruptly to meaningless values. It is performed by using the internal dynamics of the gradient descent (GD) method. If the structure of the network is available for Lyapunov synthesis, we can show the exact stability behavior of the models. Some discrete and continuous NN stability analyses are represented in [3]. In this proposed model, bounded-input bounded-output (BIBO) stability will be shown by using the ALR in the Lyapunov stability analysis to result in boundedness of modeling parameters and errors. In addition to the modeling parameters, the PID controller parameters are updated with this ALR.

This paper is organized as follows. In Section 2, general RBF networks, the proposed RBF network and stability analysis are explained. In Section 3, inverse modeling control by the proposed network is given with simulations. In Section 4, the sliding-mode control approach with proposed network identification

* Corresponding author. Tel.: +90 232 311 1665; fax: +90 232 388 60 24.
E-mail addresses: selami.beyhan@ege.edu.tr (S. Beyhan), musa.alci@ege.edu.tr (M. Alcı).
[1] Tel.: +90 232 311 1836; fax: +90 232 388 60 24.

0019-0578/$ – see front matter © 2010 ISA. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.isatra.2010.04.005

simulations is shown. In Section 5, the PID control parameters are tuned adaptively by the RBF network to control a nonlinear system. Finally, in Section 6, the proposed model is applied to real-time control of the cascaded parallel two-tank liquid-level control system. The corresponding theorems are proved in the Appendix.

2. Radial basis function networks

Radial basis function networks (RBFNs) are one of the different functionalized type of NNs with high approximation and regularization capability. RBFs are preferred as the basic structure of neural networks because of their good local specialization and global generalization ability [5]. The design of an RBFN in its most basic form consists of three separate layers. The first layer is the input layer. The second layer is the hidden layer and it is structured with high dimension to provide better approximation. The last layer gives the output of the network. The general RBFN model has a nonlinear transformation between the input layer and the hidden layer, but a linear transformation from the hidden layer to the output layer. Some basis functions are utilized as RBFs, such as Gaussian RBFs, multi-quadratic RBFs, inverse multi-quadratic RBFs, thin-plate spline RBFs, cubic spline RBFs, and linear spline RBFs. However, Gaussian RBFs are employed frequently in networks, since they are bounded, strictly positive and continuous on $\Re^n$ [3]. Moreover, they are known to have noise suppression properties [6]. Therefore, in this study, Gaussian RBFs are structured in the proposed model.

$$R_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \tag{1}$$

where x is the input vector, c_i is the center and σ_i is the standard deviation of the Gaussian function, respectively. The optimization of the centers and standard deviation provides better approximation and interpolation capability as compared to the sigmoid functions, which will be seen later in the simulations. In this study, as stated above, the GD method is employed with ALR to train the parameters in online sense.

2.1. Proposed RBF network

The RBF network introduced has one hidden layer neural network (NN) with all its parameters being adaptable. The network parameters are optimized by the gradient descent (GD) method with ALR whose stable convergence behavior is proved by Lyapunov stability analysis. The simplified modeling scheme with two inputs and one output model is represented in Fig. 1. After the realization of the construction, the network can be designed with different numbers of inputs and outputs. The idea of construction of this RBFN is to combine the power of the models, which have different mapping abilities. These models are the auto-regressive with exogenous input (ARX) model, the nonlinear static NN model and the nonlinear dynamic NN model. Therefore, the proposed RBFN is constructed in two parts. The first part is the linear ARX modeling part. There exist past values of inputs and outputs. The second part is the nonlinear static NN part and the locally recurrent dynamic NN part, which are excited with the same ARX terms. In Fig. 1, the ARX inputs are given as u_{k-1}, y_{k-1} and y_{k-2}. However, the u_{k-1} and y_{k-1} inputs are used to excite the static and dynamic parts of the model. These are known NN models; however, they have not been introduced together previously. Using a suitable optimization, the network is seen as a well-defined alternative model for the identification. The dynamic NN part is called a "Block-Diagonal Neural Network" in [7]. The ARX modeling part itself is not adequate model to identify nonlinear systems online well. However, the proposed mixed structure has fine approximation capability. The general output formula of the network is given by

$$\hat{y}(k+1) = \sum_{i=1}^{nd} \alpha_i u(k-i) + \sum_{j=1}^{np} \beta_j y(k-j) + \sum_{k=1}^{nk} \gamma_k f\left(\frac{\|x_k^1 - c_k\|^2}{2\sigma_k^2}\right) + \sum_{l=1}^{nr} \xi_l f\left(\frac{\|x_l^2 - c_l\|^2}{2\sigma_l^2}\right) \tag{2}$$

where nd and np are selected delays of the inputs and outputs. In addition, nk and nr are the numbers of RBFs used for the static and dynamic parts of the network, respectively. The output of the network is implicitly found as

$$\hat{y}(k+1) = W^T(k)\phi(k) \tag{3}$$

where α_i, β_j, γ_k and ξ_l are the parameters of the vector $W^T(k)$ and $\phi(k)$ is the input vector. The RBF inputs for the static part, which is superscripted as 1, are as follows:

$$\begin{aligned} x_1^1 &= w_{1,1}^1 u_{k-i} + w_{2,1}^1 y_{k-j} \\ x_2^1 &= w_{1,2}^1 u_{k-i} + w_{2,2}^1 y_{k-j} \end{aligned} \tag{4}$$

for dynamic part, which is superscripted as 2, they are

$$\begin{aligned} x_1^2 &= w_{1,1}^2 u_{k-i} + \hat{y}_1^2(k-1) + \hat{y}_2^2(k-1) \\ x_2^2 &= w_{1,2}^2 u_{k-i} + \hat{y}_1^2(k-1) + \hat{y}_2^2(k-1) \end{aligned} \tag{5}$$

where subscripts 1 and 2 show the first and second RBF of the blocks. The weights; i.e., $w_{1,2}^1$ is the weight from input 1 to the second RBF of block 1.

2.2. Stability analysis

The stability of the modeling is as important as the controller stability. To extract the nonlinear dynamics of the system with well-optimized parameters in a stable sense is necessary for modeling and identification. Identification stability is first related to the optimization method convergence to a local or global minimum. The convergence does not take into consideration the parameters' magnitude or other characteristics. Therefore, while training the NNs and FLSs, the convergence of the error to the minimum does not show the stability of the system unless one is using some other stability conditions. Therefore, there has to be a law for stability in optimizing the parameters. This law is derived by using the Lyapunov stability, input-to-state stability (ISS), bounded-input bounded-output stability (BIBO), a passivity approach, etc. Previously, constant stability guaranteed learning rates have been used in backpropagation algorithms. Nevertheless, these learning rates are chosen heuristically, but they do not provide the good convergence in the algorithm. However, if it is determined by online current knowledge of the inputs and system structure, the change in model parameters depends on the current change in the dynamics of the network. The general GD optimization [4,6] of parameters is given by

$$W(k+1) = W(k) - \eta(k)\frac{\partial E(k)}{\partial W(k)} \tag{6}$$

where $W(k)$ and $\eta(k)$ are current parameters vector and time-varying learning rate, respectively and $E(k)$ is the identification cost function. In this work, the following ALR in Theorem 1 is used in the gradient descent algorithm.

Theorem 1. *If the adaptive learning rate of the GD learning (6) is chosen as*

$$\eta(k) = \frac{\mu}{1 + \phi(k)^T\phi(k)} \quad (0 < \mu \le 1), \tag{7}$$

then the global convergence of identification error and parameters is guaranteed, where $\phi(k)$ is the Jacobian of the output $\hat{y}(k)$ with respect to parameters vector $W(k)$.