ISSN: 2348 - 2117

Available Online at www.ijetcr.org

International Journal of Engineering Technology and Computer Research (IJETCR) 1 (1) 2013, 20-24

ARTIFICIAL NEURAL NETWORKS STRUCTURAL DESIGN WITH DYNAMICALLY MANAGEMENT SYSTEM

[1] Nishu Sharma, [2]Ankita, [3]Smriti Kiran

[1] B.Tech- M.Tech Scholar, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India.

[2] B.Tech- M.Tech Scholar, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India.

[3] M.C.A. Scholar, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India.

Received 10 October 2013; Revised 05 November 2013; Accepted 10 November 2013

ABSTRACT

Artificial neural network has at least two physical components, namely, the processing elements and the connections between them. The processing elements are called neurons, and the connections between the neurons are known as links that are working in as dynamically management system. The field of management is currently well developed in the area of analysis and design of linear time-invariant dynamical systems. ANNs are parallel distributed processing devices, they can be implemented in parallel hardware. Therefore, as a consequence of the currently possible very fast processing capability, ANNs can be used in real-time management.

Key Words: ANN, Dynamic, Linear, Manager, BPTT

INTRODUCTION:

The field of management is currently well developed in the area of analysis and design of linear time-invariant dynamical systems. However, the area of management of nonlinear dynamical systems is much less advanced and few general results are available. Therefore each system is treated on a case-by-case basis. Among other features, the ability of feed forward ANNs to approximate arbitrary nonlinear mappings has attracted the attention of several management engineers since it provides a new approach to the difficult problem of nonlinear dynamical management [1].

In this research paper we firstly review some approaches proposed in the writing to integrate feed forward ANNs in the general management structure. The concept of feedback error- learning is then introduced and we propose a feedback-error-learning management structure. Such a management structure is then mathematically analyzed and we show that, at least for the case of single input-single output linear dynamical systems, when certain requirements are satisfied, the inverse dynamical model can be correctly identified.

In the subsequent section the technique of using a variable feedback management is proposed. Simulations of the management of a two-joint robot by an ANN are presented and we show that the use of a variable feedback management improves the performance of the neural management in relation to other trajectories that were not used during training.

1.Artificial Neural Networks and Dynamical Management:

Highlights the following features of ANN as important to their application in management:

ANNs have the theoretical ability to approximate arbitrary nonlinear mappings. There is also the possibility that such an ANN approximation is more parsimonious, i.e. it requires less parameter, than other competitive techniques such as orthogonal polynomials, splines, or Fourier series. However, this has yet to be theoretically proven [2].

Since ANNs can have multi-inputs and multi-outputs, they can be naturally used for management of multivariable systems. ANNs can be trained off-line using past data records of the system to be management or they can be adapted on-line in order to compensate for changes in the management system [4].

Since ANNs are parallel distributed processing devices, they can be implemented in parallel hardware. Therefore, as a consequence of the currently possible very fast processing capability, ANNs can be used in real-time management. Also due to their distributed organization,

*Corresponding author: Nishu Sharma | Email: nishu191994@gmail.com

ANNs have the possibility of offering, when properly trained, a good level of fault tolerance against internal damage to the network itself.

By using ANNs it may be possible to perform efficient sensor data fusion where symbolic and numerical information received from different types of sensor can be naturally integrated. Similar to performing sensor data fusion, one could perform actuator integration, where several actuators act on different systems without a one-to-one correspondence between actuator and system (each actuator provides signals to several systems and each system receive signals from several actuators). Such features could result in neural management that is robust to loss of sensors and actuators.

If one could obtain all these features simultaneously, the result would be very impressive: a real-time multi-variable nonlinear adaptive fault-tolerant management. Nature, through evolution, has managed to achieve such management in biological systems. However, the current artificial neural and non-neural systems are still far from achieving such a level of refinement.

2.Neural Structural Design Management:

In this, that one of the problems of applying ANNs is to decide how to construct them, e.g. to decide the number of layers, the number of units in each layer and the activation functions for each unit. When using ANNs for management, an additional problem is to decide how to integrate the ANN into the management structure, i.e. the neural management architecture. Classifies the current neural management architectures into five categories: supervised management, adaptive critic, back-propagation through time, direct inverse management and neural adaptive management [3]. In this section we explore the basic characteristics of these neural management architectures.

A. Supervised Management System:

The first neural net management was probably the one built by B. Widrow and F. W. Smith at Stanford University in 1963 [5]. It was used to management a motorized cart that had an inverted pendulum on its top. The cart moved along a one dimensional finite and straight track. The management was required to balance the inverted pendulum in the upright position, keeping the cart position within certain bounds. The management was of the bang-bang type, i.e. it could only apply a force of constant magnitude in the left or right direction.

Figure: 1 show the neural management architecture proposed by Widrow and Smith [6]. The basic idea is to use an already existing management to train the neural management. This type of neural management architecture is known as supervised management system.

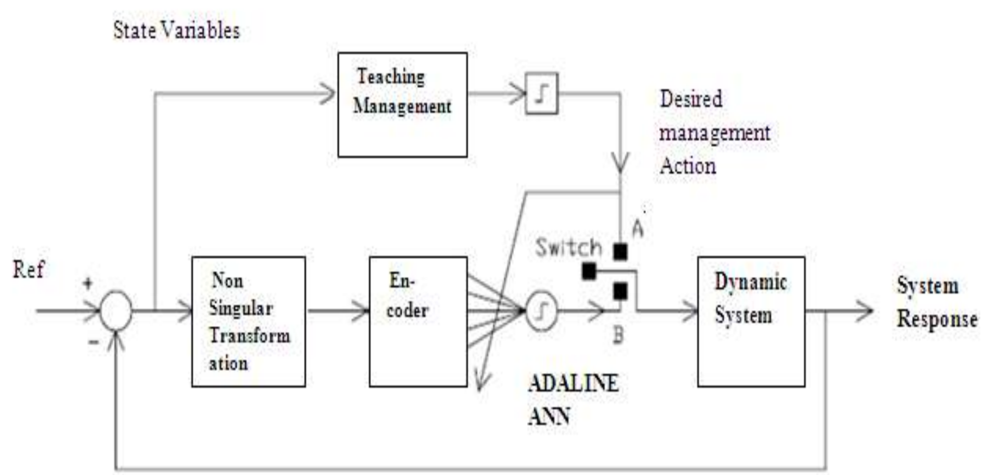

Figure 1: Supervised Management as ADALINE Manager

In their original 60's work the teaching management implemented a linear switching surface, i.e. the location of the switching surface was a linear function of the state variables of the system being management [3]. With the switch shown in figure: 1 in position A, the neural management is being trained to approximate the desired switching surface provided by the teaching management and the teaching manager still provides the management action for the system. When the switch is in position B, training stops and the ADALINE such as a TLU with bipolar output management system.

B. **Adaptive Critic and Reinforcement Learning system:**

In order to use the supervised management architecture it is necessary to have the existence of a "teacher", for instance a human expert in the particular management task that can provide the correct action for every state of the system. For this reason it is said that in supervised management the ANN "learns with a teacher". In some situations, however, no such a teacher is available. Furthermore, a series of actions have to be taken before it is possible to evaluate the success of the actions. In general, the success of such actions is simply evaluated as a "success" or as a "failure", i.e. the evaluation is not quantitative but only qualitative. In this case the ANN is said to "learn with a critic". The crucial problem in this case is known as the temporal credit assignment problem [7], i.e. which actions should receive the blame in case of failure or the credit in case of success. This is different from structural or spatial credit assignment, where the problem is to attribute the network output error to different units or weights.

The idea of the adaptive critic neural architecture is based on the principle of reinforcement learning, a term borrowed from the theories of animal learning, i.e. reward the correct actions and punish the wrong actions. Once a series of actions is taken and the result is evaluated as success or failure, the basic idea of selective bootstrap adaptation is as follows:

1. if the result is evaluated as a success, the actions are rewarded by applying the LMS algorithm to each action using as desired output the output already used in that action. They called this positive bootstrap adaptation or learning by reward.

2. if the result is evaluated as failure, the actions are punished by applying the LMS algorithm to each action using as desired output the inverse of the output used in that action. They called this negative bootstrap adaptation or learning by punishment.

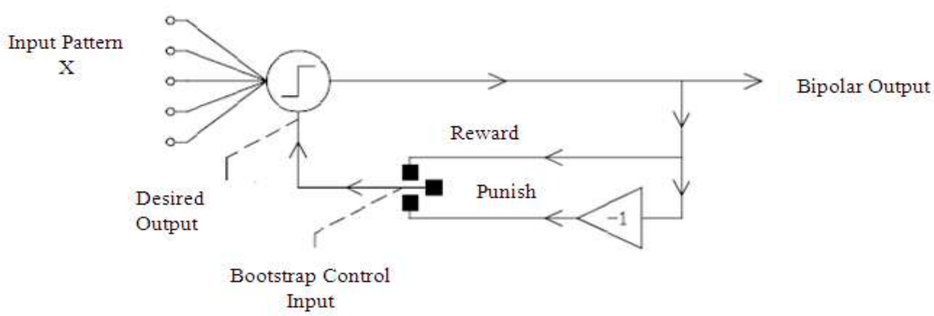

Figure 2: Bootstrap Adaptation

4. Back-Propagation through Time:

In some situations the user can specify a cost function to be minimized or maximized that, as in the case of the adaptive critic, depends on a series of actions. However, since a cost function is specified, after the actions are executed, the user can evaluate the performance of the management quantitatively. Furthermore, if a model of the plant and the system being management can be developed, then the BPTT method is used to calculate the derivative of the cost function with respect to current action. This derivative is then used to update the ANN responsible for generating the actions, i.e. the ANN management. Figure: 3 show the location of the ANN management in relation to the plant. The plant may be nonlinear and assumed to the perform the nonlinear mapping $\mathbf{X}_{k+1} = \mathbf{A}(\mathbf{X}_k, \mathbf{U}_k)$ where **x** are the plant states and **u** the plant inputs.

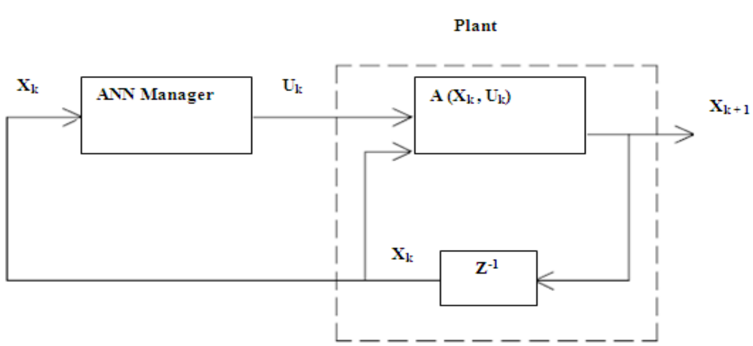

Figure: 3: ANN manager for BPTT

In order to exemplify the technique, Nguyen and Widrow [8] used the BPTT method to train an ANN to back-track a truck with a trailer to a specific point of a loading dock, with the requirement that the back of the trailer is as parallel as possible to the loading dock.

5. Direct Inverse Management:

In this neural management architecture the basic idea is to train an ANN as the inverse dynamical model of the plant. After being trained the ANN is simply used as a feed forward management such that the composition of ANN and plant act as the identity mapping [9]. The assumption in this case is that, at least in the space where the ANN is being used, the inverse dynamical model of the plant is uniquely defined and is stable.

It is possible to use a state-space formulation or an input-output formulation. In the former case the ANN inputs are the current state vector $\mathbf{x}_k$ and the desired state at the next time step $\mathbf{x}^d_{k+1}$. In the latter case the ANN inputs are: 1 and the desired plant output at the next time step $\mathbf{y}^d_{k+1}$ **: 2** the current and past plant outputs $\mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_{k-Ny}$**:3** the past plant inputs $\mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k-Nu}$. In both cases the ANN output is $\mathbf{u}_k$, the input that should be applied at the plant at time step k. Note that plant input and output at any particular time step can be vectors [10].

6. Neural Adaptive Management:

The neural adaptive management architecture basically follows the same designs as conventional linear adaptive management, with the difference that the linear mappings used in the latter case are replaced by ANNs. Possible designs arise in self-tuning and model-reference adaptive management [11].

Neural adaptive management based on the model-reference approach, most of them for single-input single-output plants, but also some examples for the multi-input multi output case.

The aim in model-reference adaptive management is to make the plant behave like a reference model, which must be supplied by the designer. First, an ANN learns to emulate the plant in the identification stage. This identification is initially carried out off-line. Later, the parameters of a second ANN, which acts as the management, are updated using the model of the plant generated by the first ANN. The identification process continues also during the management stage, so that the model represented by the ANN emulator is fine-tuned on-line.

7. CONCLUSION:

The field of Management is currently well developed in the area of analysis and design of linear time-invariant dynamical systems. ANNs can be trained off-line using past data records of the system to be management or they can be adapted on-line in order to compensate for changes in the management system. When using ANNs for management, an additional problem is to decide how to integrate the ANN into the management structure.

8. REFERENCES:

1. Hunt, K. J., Sbarbaro, D., Zbikowski, R. & Gawthrop, P. J. (1992,November). Neural Networks for Control Systems - A Survey,*Automatica*, *28* (6), 1083-1112.
2. Sontag, E. D. (1993, 6 July). Some Topics in Neural Networks andControl, Department of Mathematics, Report Number LS93-02, Rutgers University, New Brunswick, USA.
3. Werbos, P. J. (1990). Overview of Designs and Capabilities, in W. T. Miller, R. S. Sutton & P. J. Werbos (Eds.), *Neural Networks for Control*, pp. 59-65, Cambridge, USA: Bradford Books/MIT Press.
4. J.R.Haritsa, M.O.Ball and N.Roussopoulos, A. Datta and S.Baras,"Managing Networks using Database Technology," Selected Areas inCommunication, IEEE Journal, vol.11, pp.1360-1372, December 1993.
5. Nascimento Jr., Cairo L. & McMichael, Daniel (1991, April). Application of Neural Networks in Control, Control Systems Centre, Dept. of Electrical Engineering and Electronics, CSC Report 726, UMIST, Manchester, UK.
6. Widrow, B. (1987, May). The Original Adaptive Neural Net Broom-Balancer, in *Proc. Of the 20th IEEE Int. Symp. on Circuits and Systems*, pp. 351-357.
7. Hertz, J. A., Krogh, A. S. & Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*, Lecture Notes Volume I in the Santa Fe Institute Studies in the Sciences of Complexity, Redwood City, USA: Addison-Wesley Publishing Co.
8. Nguyen, D. & Widrow, B. (1990, April). Neural Networks for Self-Learning Control Systems, *IEEE Control Systems Magazine*, *10* (3).
9. Zurada, Jacek M. (1992). *Introduction to Artificial Neural Systems*, New York, USA: West Publishing Company.
10. Narendra, K. S. & Parthasarathy, K. (1990, March). Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Transactions on Neural Networks*, *1* (1), 4-27.
11. Narendra, K. S. (1990). Adaptive Control Using Neural Networks, in W. T. Miller, R. S. Sutton & P. J. Werbos

(Eds.), *Neural Networks for Control*, pp. 115-142, Cambridge, USA: Bradford Books/MIT Press.

12. Irwin W. Sandberg, James T. Lo, Craig L. Fancourt, José C. Principe, Shigeru Katagiri, Simon Haykin "Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives" Willy publication- February 2001.

