A New Class of Intelligent Neural Network Controllers

D. G. Lainiotis, K. N. Plataniotis
Florida Institute of Technology
Department of Electrical and Computer Engineering
150 W. University Blvd., Melbourne Fl. 32901, USA

Abstract - In this paper we present, following the Lainiotis multi-partitioning control methodology, a new class of neural networks based controllers. Moreover we investigate the performance of the new class of controllers by extensive simulations for the case of linear and nonlinear problems. The results show that the new controllers have excellent performance, achieving significant computational savings due to their massively parallel structure.

I. INTRODUCTION

In recent years, there has been a growing interest in the understanding of the adaptive control problem [1]. This is a problem of major theoretical as well as practical importance. In many practical design situations, the designer does not have at his disposal sufficient data to derive a complete mathematical model of the process he endeavors to control. Instead, uncertainty may exist with respect to certain parameters in the system description. Such systems have been termed "adaptive" in order to emphasize the uncertainty in the process description. The control problem for such a system becomes more difficult because the control input that minimizes the performance criterion, may be functionally dependent on the undetermined parameters. Various algorithms have been proposed in the past to solve the adaptive control problem. Among them the most notable the various Multi-partitioning Adaptive Controls (MACs) introduced and studied by Lainiotis [2]-[13].

Recently, neural networks [14],[15] have been applied to linear and nonlinear control problems [17]-[19]. The application of neural networks to the control problem is due to the fact and mainly because they do not require a-priori knowledge of the model. Instead in contrast to previous methodologies, neural network controllers are trained rather than designed by measurements and desired output pairs. As such, neural network controls can be "designed" whether a dynamical model is known or not. In many important control applications dynamical models of the underlying process are known from physical considerations with uncertainty to specific parameter values. In this paper we present, following the multi-partitioning approach of Lainiotis, a new class of neural network based multi-partitioning controllers. Moreover we investigate the performance of this multi-partitioning neural controllers by extensive simulations for the case of linear and nonlinear problems.

In the following section the neural algorithms and the configuration of the networks are discussed. Simulation results for linear and nonlinear models are provided in section III. Finally, section IV summarizes the results of this work.

II. ADAPTIVE SYSTEM

In general,the state space model for an adaptive stochastic nonlinear system in discrete time has the following form

$$x(k+1) = f(k, x(k), u(k), \theta(k), w(k)) \qquad (1)$$

$$z(k) = h(k, x(k), \theta(k), v(k)) \qquad (2)$$

where

- x(k) is the n dimensional state of the system which the initial value x(0) is coming from a known probability density function
- u(k) is the system's deterministic input vector
- z(k) is the system's output vector (measurement)
- f(), h() are in general arbitrary non linear functions
- w(k), v(k) represent modelling errors and stochastic disturbances and usually assumed white Gaussian noises.
- $\theta(k)$ is the unknown parameter vector, that summarizes the uncertainty in the above system

The objective for the control designer in a system described by equations (1)-(2) is to identify the system dynamics by a neural estimator, and control it to track a state command signal $\bar{x}(k)$ using a per step quadratic performance index:

$$J(k) = E(Jx(k) + Ju(k)) \qquad (3)$$

$$Jx(k) = (x(k) - \bar{x}(k))^t A_k (x(k) - \bar{x}(k)) \qquad (4)$$

$$Ju(k) = (u(k-1) - \bar{u}(k-1))^t B_{k-1} (u(k-1) - \bar{u}(k-1)) \qquad (5)$$

where E() denotes the mathematical expectation, and A,B are symmetric positive definite weighting matrices, and $\bar{u}(k)$, $\bar{x}(k)$ are the desired control and state respectively.

From the above state space model it can be seen that in a discrete adaptive system there are two mapping for a neural network to identify. One is the mapping f() which maps the past system state and the inputs (deterministic and stochastic) to the new state x(k+1). The second is mapping h() which transforms the state x(k) into the measurable output z(k).

III. NEURAL CONTROLLER

The definition of the performance index implies that we have to choose a control strategy such that the distance between the actual and the desired trajectories decreases. When multilayer perceptrons are used, the gradient descent methodology is used to obtain the control law. The chain rule is then used to calculate the impact of the control signal u on the performance index J. The problem associated with this approach is that the influence of the control input on the state of the system is difficult to be determined analytically for arbitrary plants with unknown elements in the state space model. In the bulk of the literature [17]-[19],a suboptimal approach is used to identify the plant dynamics in the presence of unknown parameters. The so called neural identifier is used to compute an estimate of the systems state when the current state and inputs are given. In the sequel the impact of the input on the succeeding state is realized by applying the well known backpropagation rule in a multilayer perceptron.

In this work, the partition methodology [2]-[11], [12]-[13] of Lainiotis allow us to condition the plant on different values of the unknown parameters in the model. In this way the impact of the control signal can be calculated explicitly through the actual plant model. Having determined the impact of the control signal on the performance index the next step is to calculate the control input u. It is assumed that the control input is computed by a feed forward neural network, and modifications of the control signal can be obtained by modifying the weights of a multilayer perceptron which we call neural controller. Following the partitioning approach several neural controllers are trained independently using different variations of the actual adaptive model. Each one of the networks trained with data pairs obtained using different parameter vectors $\theta(k)$ in the nonlinear system that generates the data. Due to this training methodology each network converges to a different solution. When the training is over, a bank of different neural controllers is available to be applied to the solution of the problem. In the mean time another set of multilayer perceptrons are used to provide one step ahead predictions of the systems measurements [15],[20]-[22]. These neural emulators are used to predict the succeeding measurement of the plant, when the current state and control input are given. In the actual operation phase a nonlinear selection mechanism is used. The adaptive algorithm compares residuals between true and predicted measurements to identify at every time instant which neural controller minimizes the performance index.

The adaptive neural methodology described above uses the Lainiotis partitioning theorem [3],[4], and it is an extension of the Multi-partitioning Adaptive Lainiotis Control to neural networks [2]-[13]. In this case however, a bank of trained multi-partitioning neural controllers are used.The adaptive neural control scheme integrates the robustness of the neural controller with the effectiveness and attractiveness of the partitioning theory. In the next section simulation studies were carried out to test the effectiveness of the new methodology. To examine the feasibility of the adaptive neural scheme the algorithm is used to provide solutions in a series of linear and nonlinear problems. Moreover a comparison with the optimal control when the parameters that specify the model is completely known is also provided.

Simulation I

In this first example a linear state space model with unknown parameters is considered. It is assumed that two different variations of the model can be obtained using different values for the parameter vector. More specific the equations of the models are:

Model 1

$$x(k+1) = 0.6x(k) + 0.5u(k) + w(k) \tag{6}$$

$$z(k) = x(k) + v(k) \tag{7}$$

Model 2

$$x(k+1) = 0.9x(k) + 0.2u(k) + w(k) \tag{8}$$

$$z(k) = x(k) + v(k) \tag{9}$$

where w(k), v(k) are white Gaussian noises with variances 0.25 and 0.04 respectively and x(0) is considered Gaussian with mean value 0.5 and variance 0.25.

The performance index to be minimized is given be the following equation:

$$J(k) = 10x^2(k) + u^2(k-1) \tag{10}$$

In this experiment it is assumed that the actual model is Model 1. Following the adaptive neural algorithm discussed in section II a neural controller called matched neural controller is trained using the correct Model 1, and another one called mismatched neural controller is trained using the erroneous Model 2 and training data generated by it.Moreover one neural predictor is trained to provide one step ahead predictions of the Model 1, and another one is trained to predict the measurements of Model 2. In the actual. implementation phase the adaptive scheme selects the appropriate control law that satisfies the criterion of eq. (10). The resulting controller is called adaptive neural controller. In order to justify the performance of the proposed adaptive methodology a comparison with the optimal controller is made. More specific, we compare the neural controllers with the optimal statistical control obtained using the linear separation theorem [2]-[13]. However, it must be noted that if the model parameters are unknown, the optimal stochastic controller cannot be designed and implemented. In other words it is inaccessible to the designer.

The configuration of the neural controller as well as that of the neural emulator is summarized below:

A. Neural controller

network topology:

- three input nodes: the current and the previous measurements, and the previous control are used as input signals
- one output node: the control input for the next step.
- two hidden layers with 4-4 hidden nodes respectively.

learning parameters:

- learning rate: 0.1, momentum: 0.2

training procedure:

- backpropagation training algorithm.
- the target vector during training is the desired control signal as specified from the performance index.
- the network tries to minimize the square error between the current output and the target vector.
- each training set consists of 50 input/output pairs.
- the training procedure is terminated if the training error tolerance is less than 0.01 or if the number of iterations of the training set is more than 500.
- running each one of the candidate model a different training data set is obtained
- the test data record is generated using the Model 1,and the control signal produced by the adaptive algorithm.

B. Neural emulator

network topology:

- two input nodes: the current and the previous measurements are used as input signals.
- one output node: the one step ahead estimate of the measurement.
- two hidden layers with 5-2 hidden nodes respectively

learning parameters:

- learning rate: 0.1, momentum: 0.2

training procedure:

- backpropagation training algorithm.
- the target vector during training is the actual measurement which is generated running the equations of one model.
- the network tries to minimize the square error between the current output and the target vector.
- each training set consists of 50 input/output pairs.
- the training procedure is terminated if the training error tolerance is less than 0.01 or if the number of iterations of the training set is more than 1000.

The stability of the dynamics must be also known a priori, and a training set must be carefully selected so all the important nodes of the system are persistently excited.

Since there is no theoretical analysis to justify the performance of the neural controller Monte Carlo techniques are used to verify the controllers. The figure of merit used to compare performance is the quadratic index of eq. 10 averaged over 250 MC runs. Samples of the simulation results are shown in Fig. 1-3.

Simulation II

To demonstrate the effectiveness of the propose methodology we repeat the experiment using this time a nonlinear adaptive state space model. The model has a partially known plant equation and a nonlinear measurement equation. It is assumed that two different variations of the model can be obtained using different values for the parameter vector. More specific the equations of the models are:

Model 1

$$x(k+1) = 0.6x(k) + 0.5u(k) + w(k) \tag{11}$$

$$z(k) = x(k) - 0.1x^2(k) + v(k) \tag{12}$$

Model 2

$$x(k+1) = 0.9x(k) + 0.2u(k) + w(k) \tag{13}$$

$$z(k) = x(k) + 0.2x^3(k) + v(k) \tag{14}$$

where w(k), v(k) are white Gaussian noises with variances 0.25 and 0.04 respectively and x(0) is considered Gaussian with mean value 0.5 and variance 0.25. Once again the actual model is Model 1 and the performance index to be minimized is given in eq. (10). The configuration for the neural controllers and emulators is identical to that of Simulation I The results of this experiment are summarized in Fig. 4-6.

Simulation III

In this last example a nonlinear state space model with unknown parameters is considered again. It is assumed that two different variations of the model can be obtained using different values for the parameter vector. More specific the equations of the models are:

Model 1

$$x(k+1) = 0.6x(k) + (-0.05)x^2(k) + 0.5u(k) + w(k) \quad (15)$$

$$z(k) = x(k) + v(k) \quad (16)$$

Model 2

$$x(k+1) = 0.9x(k) + 0.01x^3(k) + 0.2u(k) + w(k) \quad (17)$$

$$z(k) = x(k) + v(k) \quad (18)$$

where w(k), v(k) are white Gaussian noises with variances 0.25 and 0.04 respectively and x(0) is considered Gaussian with mean value 0.5 and variance 0.25. The performance index to be minimized is the same as the previous two simulations. The results from this last experiment are given in Fig. 7-9.

IV. CONCLUSIONS

The problem of adaptive control via neural networks was addressed in this paper. A new class of adaptive neural controllers was proposed and applied in linear and nonlinear problems. The neural controller provides reliable and consistent solutions to the problem of controlling the state of a stochastic model.The multipartitioning adaptive algorithm successfully detected the actual data generation model, and selected the appropriate trained neural controller. More over the neural controller with its capability to provide accurate solutions, and its massively parallel structure and high speed, constitutes a new promising tool in systems theory.

RREFERENCES

[1] B.D.O Anderson, J.B. Moore **Optimal Filtering**, Prentice Hall, 1979.

[2] D.G. Lainiotis, *"Optimal Nonlinear Estimation"*, International Journal of Control, Vol. IJC-14, pp. 11137-1148, 1971.

[3] D.G. Lainiotis, *"Optimal Adaptive Estimation: Structure and Parameter Adaptation"*, IEEE Transactions on Automatic Control, Vol. AC-16, pp. 160-170, 1971.

[4] D.G. Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, I-Estimation", Proceedings of IEEE, Vol. 64, pp. 1126-1142, 1976.

[5] D.G. Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, II-Control", Proceedings of IEEE, Vol. 64, pp. 1182-1198, 1976.

[6] D.G. Lainiotis, J.G. Deshpande, T.N. Upadayay, "Optimal Adaptive Control: A Nonlinear Separation Theorem" Int. Journal of Control, IJC-15, pp. 877-878,1972.

[7] D.G. Lainiotis,T.N. Upadayay, J. G. Deshpande "A Nonlinear Separation Theorem" in Symp. of Nonlinear Estimation, pp184-187, 1971.

[8] D.G. Lainiotis,T.N. Upadayay, J. G. Deshpande "Optimal adaptive control of linear systems" IEEE Decision and Control Conference, pp. 247-252, 1971.

[9] J.G. Deshpande, T.N. Upadayay, D.G. Lainiotis, "Adaptive Control of Linear Stochastic Systems" Automatica, Vol. 9, pp 107-115,1973.

[10] T.N. Upadayay, D.G. Lainiotis, "Joint adaptive plant and measurement control of linear stochastic systems" Seventh Annual Princeton Conf. Information Sciences and Systems,1973

[11] T.N. Upadayay, D.G. Lainiotis, "Joint adaptive plant and measurement control of linear stochastic systems" IEEE Trans. on Automatic Control, pp. 567-571, 1973.

[12] G.N. Saridis **Self Organizing Control of Stochastic Systems**, Marcel Dekker, N.Y. 1973.

[13] K. Watanabe **Adaptive Estimation and Control: The Partitioning Approach**, Prentice Hall, 1992.

[14] D.E. Rumelhart, J.L. McClelland, **Parallel Distributed Processing: Explorations into the Microstructure of Cognition, Vol. 1**, M.I.T. Press, 1986.

[15] B. Kosko **Neural Networks for Signal Processing**, Prentice Hall, 1992.

[16] L.V. Fausett **Fundamentals of Neural Networks:Architectures,Algorithms,and Applications**, Prentice Hall, 1993.

[17] W.T. Miller III, R.S. Sutton, P.J. Werbos, **Neural Networks for Control**, MIT Press,1990.

[18] D.R Hush, B.G. Horne *"Progress in Supervised Neural Networks"*, IEEE Signal Processing Magazine, pp 8-39,1993.

[19] T.T. Ho, H.T. Ho, J.T. Balasiewicz, "Stochastic Neural Adaptive Control for Nonlinear Time Varying Systems", Intelligent Engineering Systems through Artificial Neural Networks, Vol. 2, pp 531-536, ASME Press, 1992.

[20] D.G.Lainiotis,K.N. Plataniotis, D. Menon, C.J. Charalampous, *"Heave Compensation via Neural Networks"*, Intelligent Engineering Systems through Artificial Neural Networks, Vol. 3, pp 143-148, ASME Press, 1993.

[21] D.G.Lainiotis,K.N. Plataniotis, D. Menon, C.J. Charalampous, *"Adaptive Heave Compensation via Neural networks"*, IEEE OCEANS'93 Proceedings, Vol. I, pp 243-248, 1993.

[22] D.G.Lainiotis,K.N. Plataniotis, D. Menon, C.J. Charalampous, *"Neural Network Application to Ship Position estimation"*, IEEE OCEANS'93 Proceedings, Vol. I, pp 243-248, 1993.

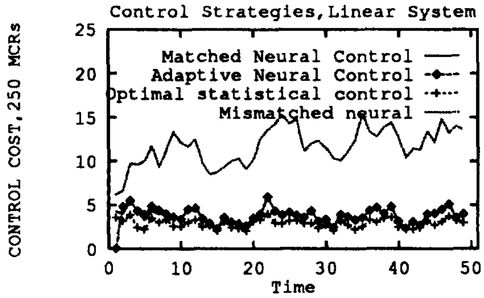

Fig. 1. Simulation I. Linear state space model

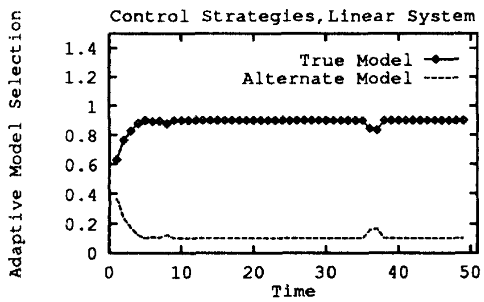

Fig. 2. Simulation I. Linear state space model

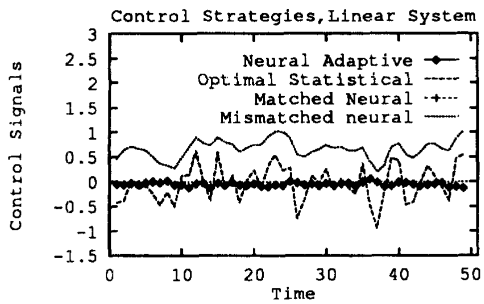

Fig. 3. Simulation I. Linear state space model

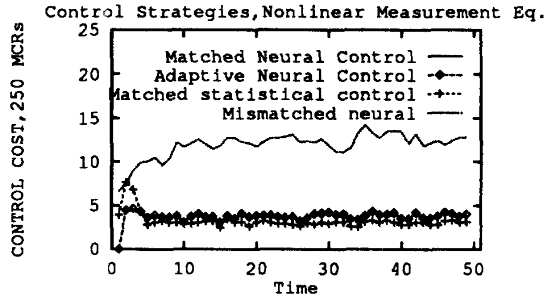

Fig. 4. Simulation II. Nonlinear state space model. Nonlinear measurement equation

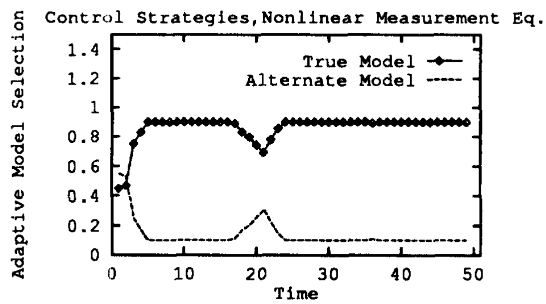

Fig. 5. Simulation II. Nonlinear state space model. Nonlinear measurement equation

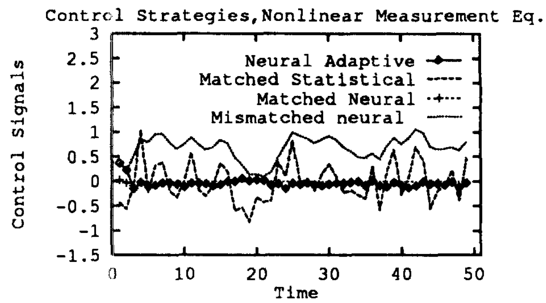

Fig. 6. Simulation II. Nonlinear state space model. Nonlinear measurement equation

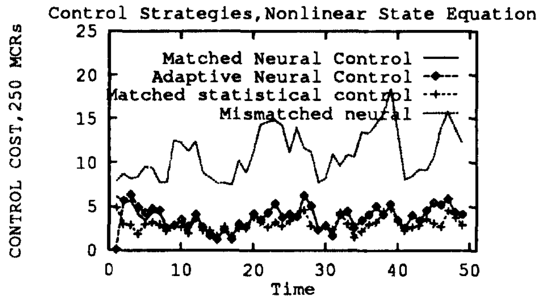

Fig. 7. Simulation III. Nonlinear state space model. Nonlinear state equation

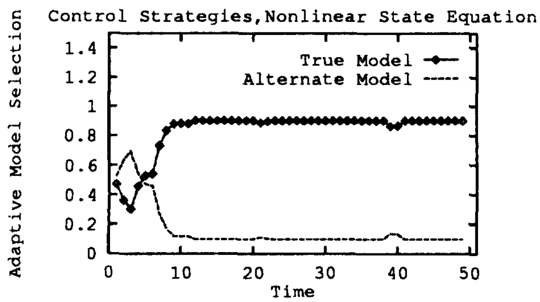

Fig. 8. Simulation III. Nonlinear state space model. Nonlinear state equation

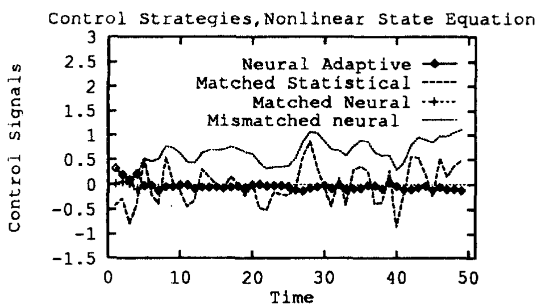

Fig. 9. Simulation III. Nonlinear state space model. Nonlinear state equation