

01 Jan 2003

Adaptive Critic Designs and Their Implementations on Different Neural Network Architectures

Jung-Wook Park

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Ronald G. Harley

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

J. Park et al., "Adaptive Critic Designs and Their Implementations on Different Neural Network Architectures," *Proceedings of the International Joint Conference on Neural Networks, 2003*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2003.

The definitive version is available at <https://doi.org/10.1109/IJCNN.2003.1223694>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Adaptive Critic Designs and their Implementations on Different Neural Network Architectures

Jung-Wook Park¹, G. K. Venayagamoorthy², *Senior Member, IEEE*, and Ronald G. Harley¹, *Fellow, IEEE*

¹School of Electrical and Computer Engineering
Georgia Institute of Technology
GA 30332-0250, U.S.A.

(e-mails: jungwookpark@iecc.org and ron.harley@ecc.gatech.edu)

²Department of Electrical and Computer Engineering
University of Missouri-Rolla
MO 65409-0249, U.S.A.
(e-mail: gkumar@iecc.org)

Abstract – The design of nonlinear optimal neurocontrollers based on the Adaptive Critic Designs (ACDs) family of algorithms has recently attracted interest. This paper presents a summary of these algorithms, and compares their performance when implemented on two different types of artificial neural networks, namely the multilayer perceptron neural network (MLPNN) and the radial basis function neural network (RBFNN). As an example for the application of the ACDs, the control of synchronous generator on an electric power grid is considered and results are presented to compare the different ACD family members and their implementations on different neural network architectures.

I. INTRODUCTION

The adaptive critic designs (ACDs) technique, which was proposed by Werbos [1], [2], is a novel nonlinear optimization and control algorithm based on the mathematical analysis to handle the classical optimal control problem by combining concepts of *reinforcement learning* and *approximate dynamic programming* (ADP).

Use of the ACD technique based on artificial neural networks (ANNs), allows the design of an optimal adaptive nonlinear controller and has the capability of optimization over time under conditions of noise and uncertainty [3], [4]. In other words, the ACDs can be used to maximize or minimize any utility function, such as total energy error, of a system over time in a noisy nonstationary environment.

The conventional continually on-line indirect adaptive neurocontroller for generator control was described in [5], which reported that the updates/adaptation of the parameters for the neurocontroller are carried out using a gradient descent algorithm based on the error only one time step ahead. This adaptation technique is therefore short sighted. A short-term goal does not guarantee a long-term satisfactory/optimal trajectory.

To overcome the above issue and provide strong robustness for the controller, the family of ACD techniques for infinite time horizon optimal control can be seen as alternatives where the ANNs are used as tools to identify the system and implement the ACD based control algorithms [6].

Also, without the extensive computational efforts and difficult mathematical analyses required by using the dynamic

programming (DP) in classical optimal control theory [7], the ACD technique provides an effective method to construct an optimal and robust feedback controller by exploiting backpropagation for the calculation of all the derivatives of user-defined target quantities [1] in order to minimize the heuristic cost-to-go approximation.

There are three representative optimization control techniques among the ACDs family. One is the heuristic dynamic programming (HDP), which approximates the heuristic cost-to-go function (J) *itself* by the critic network adaptation. Another is the dual heuristic programming (DHP), by which critic network performs the value iteration for *derivatives* of the heuristic cost-to-go function J with respect to the states of the plant. The other is the globalized dual heuristic programming (GDHP), which approximates both J and its derivatives by the critic network adaptation.

In the literature, there exist many ACDs based application and these have been implemented using the MLPNN for the HDP [3], [4], [6], [8]-[11], DHP [3], [4], [12], or GDHP [13] for the design of controllers. However, very few reports have appeared on implementing the HDP [14], DHP, and GDHP using the RBFNN.

In Ref. [4], the authors compared HDP and DHP based on the MLPNN. In Ref. [5], the authors compared the use of the MLPNN and RBFNN for implementing indirect adaptive control of a synchronous generator. In Ref. [6], the authors compared the HDP based on the MLPNN and RBFNN.

This paper extends the earlier works [4]-[6] by adding comparison of the performance of the DHP based on the MLPNN and RBFNN. Also, the advantages, which can be obtained through the ACDs based optimal control, are discussed with comparison of the indirect adaptive control.

II. ADAPTIVE CRITIC DESIGNS

The adaptive critic method determines optimal control laws for a system by successively adapting two neural networks, namely, an action neural network (which dispenses the control signal) and a critic neural network (which “learns” the desired performance index for some function associated with the index).

The model dependent designs for the HDP and DHP algorithms are briefly described below.

A. Heuristic Dynamic Programming (HDP)

The critic network in the HDP approximates the heuristic cost-to-go function J itself in (1).

$$J(k) = \sum_{p=0}^{\infty} \gamma^p U(k+p) \quad (1)$$

where γ is the discount factor ($0 < \gamma < 1$) and U is the user-defined utility or cost function. The configuration of the critic network training (for value iteration to minimize the value of J) by approximate dynamic programming is shown in Fig. 1. The following error equation [3] for the training of critic network is used.

$$e_c(t) = J(\Delta \hat{Y}(t)) - \gamma J(\Delta \hat{Y}(t+1)) - U(\Delta \hat{Y}(t)) \quad (2)$$

where $\Delta \hat{Y}(t)$ is a vector of observables of the plant, which is the output vector from the model network (Fig. 2).

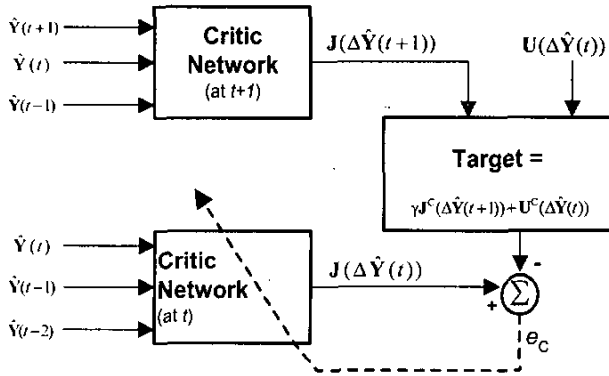


Fig. 1. Critic adaptation in HDP: The same critic network is shown for two consecutive times, $t+1$ and t . The critic's output $J(\Delta \hat{Y}(t+1))$ at time $t+1$ is required for the ADP to generate a target signal $\gamma J(\Delta \hat{Y}(t+1)) + U(\Delta \hat{Y}(t))$ for training the critic network.

The input of the action network in Fig. 2 is the output vector of the plant ($Y(t)$) and its time-delayed values. After minimizing J in (1) by the critic network, the action network is trained with the estimated output backpropagated from the critic network to obtain the converged weight for the optimal control u^* . In other words, the objective of the action network shown in Fig. 2, is to find the optimal control u^* to minimize J in the immediate future, thereby optimizing the overall cost expressed as a sum of all U over the horizon of the problem in (1). This is achieved by training the action network with an error vector $e_A(t)$ in (3).

$$e_A(t) = \frac{\partial J^c(t)}{\partial A(t)} \quad (3)$$

The derivative of the cost function $J(t)$ with respect to $A(t)$ in (3) is obtained by backpropagating $\partial J/\partial J = 1$ (recall that the HDP approximates the function J itself) through the critic network and then through the pretrained model network to the action network. This gives signals $\partial J(t)/\partial \Delta \hat{Y}(t)$ and

$\partial J(t)/\partial A(t)$ in Fig. 2 for the weights $W_A(t)$ and the output vector $A(t)$ of the action network. The expression for the weights' update in the action network is given in (4).

$$\Delta W_A(t) = -\eta_A \cdot e_A(t) \cdot \frac{\partial e_A(t)}{\partial W_A(t)} \quad (4)$$

where η_A is the positive learning rate.

The general training procedure for the model, critic, and action networks in the HDP is explained in [3] and [5].

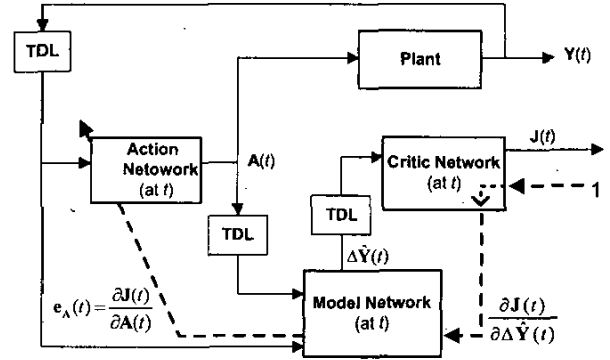


Fig. 2. The configuration for the action network adaptation in HDP.

B. Dual Heuristic Programming (DHP)

The critic network in the DHP approximates the derivatives of the heuristic cost-to-go function J in (1) with respect to the states of the plant. In other words, the value iteration by the critic network in the DHP is carried out with *perfect state information* of the plant, which means that the actual suboptimal path to minimize J is changed, and the corresponding optimal control u^* is determined in the different optimal policy set.

The configuration for the critic network adaptation in the DHP is shown in Fig. 3. The input and output vectors of the model, action, and critic networks are the same as those in the HDP. For the critic network adaptation in the DHP, it learns to minimize the following error measure over time:

$$\|E_c\| = \sum_t e_c^T(t) e_c(t) \quad (5)$$

$$e_c(t) = \frac{\partial J[\Delta \hat{Y}(t)]}{\partial \Delta \hat{Y}(t)} - \gamma \frac{\partial \hat{J}[\Delta \hat{Y}(t+1)]}{\partial \Delta Y(t)} - \frac{\partial U[\Delta Y(t)]}{\partial \Delta Y(t)} \quad (6)$$

After exploiting all relevant pathways of backpropagation as shown in Fig. 3, where the paths of derivatives and adaptation of the critic network are depicted by dotted and dash-dot lines, the error signal $e_c(t)$ is used for training to update the weights of the critic network.

The j^{th} component of the second term in (6) can be expressed by the output of the critic network at time $t+1$, $\lambda_j(t+1) = \partial \hat{J}[\Delta \hat{Y}(t+1)] / \partial \Delta \hat{Y}_j(t+1)$ as follows.

$$\frac{\partial \hat{J}[\Delta \hat{Y}(t+1)]}{\partial \Delta Y_j(t)} = \sum_{i=1}^n \hat{\lambda}_i(t+1) \frac{\partial \hat{Y}_i(t+1)}{\partial \Delta Y_j(t)} \quad (7)$$

$$+ \sum_{k=1}^m \sum_{i=1}^n \hat{\lambda}_i(t+1) \frac{\partial \Delta \hat{Y}_i(t+1)}{\partial A_k(t)} \frac{\partial A_k(t)}{\partial \Delta Y_j(t)}$$

where n and m are the numbers of outputs of the model and the action networks, respectively.

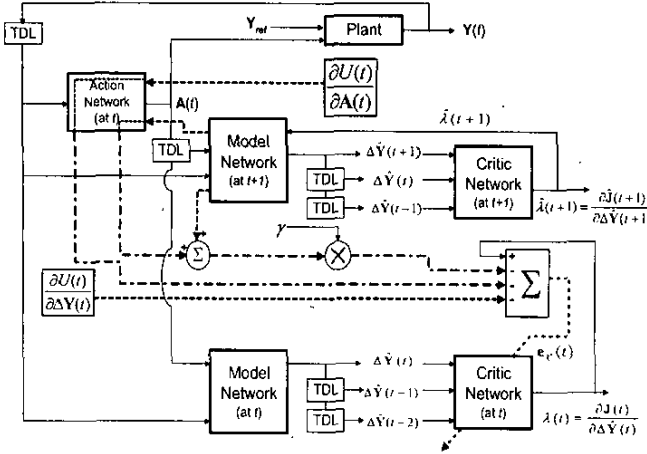


Fig. 3. Critic network adaptation in the DHP as in (8). The same critic network is shown for two consecutive times, t and $t+1$. The discount factor γ is chosen to be 0.5. Backpropagation paths are shown by dotted and dash-dot lines. The output of the critic network $\hat{\lambda}(t+1)$ is backpropagated through the model from its outputs to its inputs, yielding the first term of (6) and $\partial \hat{J}(t+1)/\partial A(t)$. The latter is backpropagated through the action network from its outputs to its inputs forming the second term of (6). Backpropagation of the vector $\partial U(t)/\partial A(t)$ through the action results in a vector with components computed as the last term of (8). The summation of all these signals produces the error vector $e_c(t)$ used for training the critic network.

By using (7), each of j components of the vector $e_c(t)$ from (6) is determined by

$$e_{c_j}(t) = \frac{\partial J[\Delta \hat{Y}(t)]}{\partial \Delta Y_j(t)} - \gamma \frac{\partial \hat{J}[\Delta \hat{Y}(t+1)]}{\partial \Delta Y_j(t)} \quad (8)$$

$$- \frac{\partial U[\Delta Y(t)]}{\partial \Delta Y_j(t)} - \sum_{k=1}^m \frac{\partial U[\Delta Y(t)]}{\partial A_k(t)} \frac{\partial A_k(t)}{\partial \Delta Y_j(t)}$$

The adaptation of the action network in Fig. 3 is illustrated in Fig. 4, which propagates $\hat{\lambda}(t+1)$ back through the model network to the action network. The goal of this adaptation is expressed in (9) [3], and the weights of the action network are updated by (10).

$$\frac{\partial U[\Delta Y(t)]}{\partial A(t)} + \gamma \frac{\partial \hat{J}[\Delta \hat{Y}(t+1)]}{\partial A(t)} = 0 \quad \forall t. \quad (9)$$

$$\Delta W_A(t) = -\eta_A \left[\frac{\partial U[\Delta Y(t)]}{\partial A(t)} + \gamma \frac{\partial \hat{J}[\Delta \hat{Y}(t+1)]}{\partial A(t)} \right]^T \frac{\partial A(t)}{\partial W_A(t)} \quad (10)$$

where η_A is a positive learning rate and W_A contains the weights of the DHP action network.

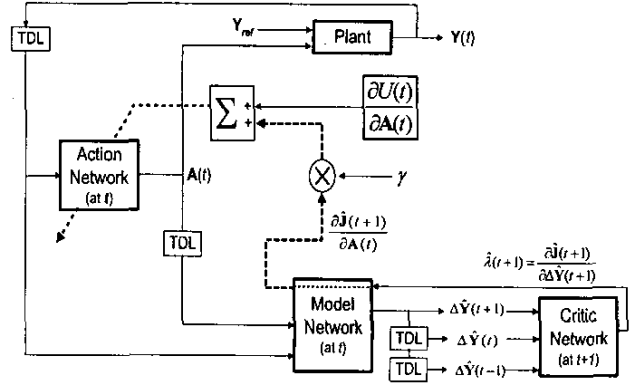


Fig. 4. Action network adaptation in the DHP: Backpropagation paths are shown by dotted lines. The output of the critic network $\hat{\lambda}(t+1)$ at time $(t+1)$ is backpropagated through the model network from its outputs to its inputs (output of the action), and the resulting vector multiplied by the discount factor ($\gamma = 0.5$) and added to $\partial U(t)/\partial A(t)$. Then, an incremental adaptation of the action network is carried out by (9) and (10).

The structures and equations for the MLPNN and RBFNN used in this paper appear in [5], [6], and [16].

III. CASE STUDY: SYNCHRONOUS GENERATOR CONTROL

A. Plant Modeling

The synchronous generator, turbine, exciter and transmission system connected to an infinite bus form the plant (dotted block) in Fig. 5 that has to be controlled [5], [6].

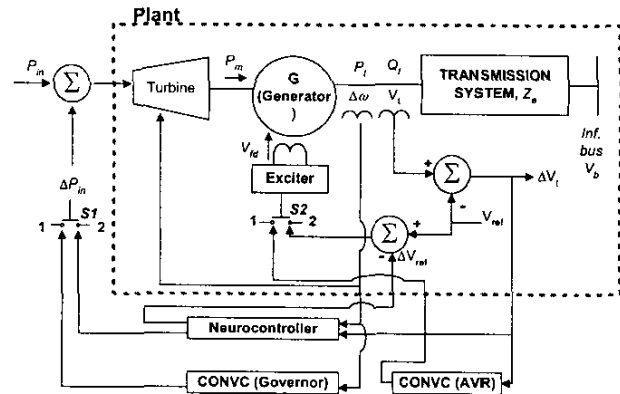


Fig. 5. Plant model used for the control of a synchronous generator connected to an infinite bus.

In the plant, P_i and Q_i are the real and reactive power at the generator terminal, respectively, Z_e is the transmission line impedance, P_m is the mechanical input power to the generator, V_{fd} is the exciter field voltage, V_b is the infinite bus voltage, $\Delta \omega$ is the speed deviation, ΔV_t is the terminal voltage deviation, V_t is the terminal voltage, ΔV_{ref} is the reference

voltage deviation, V_{ref} is the reference voltage, ΔP_{in} is the input power deviation, and P_{in} is the turbine input power.

The position of the switches $S1$ and $S2$ in Fig. 5 determines whether the optimal neurocontroller, or the conventional controller (CONVC) consisting of governor and AVR, is controlling the plant. Block diagrams, time constants, and gains for the CONVC (AVR/exciter and turbine/governor systems) are given in [6].

B. Simulation Results

With the fixed parameters after off-line training by the same procedures (explained in [3] and [6]) for the model, critic, and action networks in the HDP and DHP, the dynamic performances of the following nonlinear optimal neurocontrollers are evaluated and compared with the CONVC.

- Neurocontrollers designed by the HDP using the MLPNN and RBFNN are called the MHDPC and RHDPC, respectively.
- Neurocontrollers designed by the DHP using the MLPNN and RBFNN are called the MDHPC and RDHPC, respectively.

The following two different types of disturbances are applied to the plant for the tests of improvement of system damping and transient stability.

- A three phase short circuit at the infinite bus in Fig. 5: At $t = 0.3$ s, a temporary three phase short circuit is applied at the infinite bus for 100 ms from $t=0.3$ s to 0.4 s for the plant operating at the steady state condition.
- $\pm 5\%$ step changes in the reference voltage of the exciter in Fig. 5: The synchronous generator of the plant is operating at a steady state condition. At $t = 1$ s, a 5% step increase (ΔV_{ref}) in the reference voltage of the exciter is applied. At $t=12$ s, the 5% step increase is removed, and the system returns to its initial operating point.

The results in Figs. 6 to 8 show that the optimal neurocontrollers improve the damping of low-frequency oscillations more effectively compared to the CONVC (in Fig. 6), and that the RHDPC outperforms the MHDPC for the dynamic transient response (for the new reference value), i.e. the RHDPC has a faster rising time than the MHDPC (Figs. 7 and 8). Note that the increased damping is important for generators in power system networks. From Fig. 6, it is shown that two optimal neurocontrollers (MDHPC/RDHPC) based on the DHP improve the damping of low-frequency oscillations more effectively than the MHDPC/RHDPC and the CONVC.

The results in Figs. 7 and 8 for a step change show that the DHP based neurocontrollers outperform the HDP based neurocontrollers. Also, the RHDPC has a faster rise time than the MHDPC. Especially, the performance of the MDHPC is significantly improved compared to that of the MHDPC.

Moreover, the MDHPC shows a slightly faster rise time and smaller overshoot than the RDHPC.

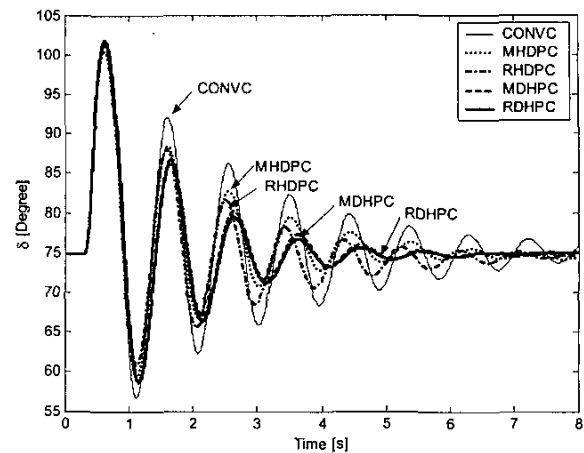


Fig. 6. Three phase short circuit test: Rotor angle.

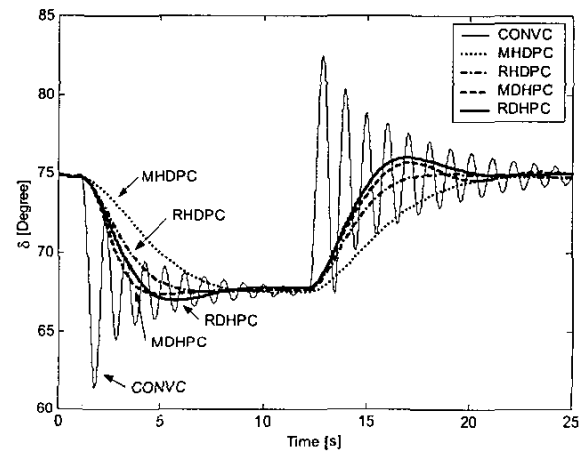


Fig. 7. $\pm 5\%$ Step changes in reference voltage of exciter: Rotor angle.

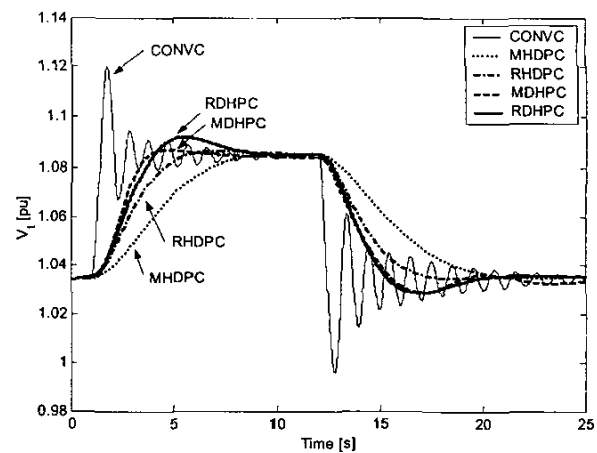


Fig. 8. $\pm 5\%$ Step changes in reference voltage of exciter: Terminal voltage.

C. Adaptive Critic Optimal control vs. Indirect Adaptive Control

The robustness of a controller [7], [17], [18] is judged by how well it controls a process even during uncertain changing system configurations. Therefore, the HDP and DHP techniques, which are based on the *infinite horizon optimal control* method, provide robust feedback control; this powerful dynamic control capability of adaptive critic optimal controllers has been evaluated in the previous subsection B. This robustness comes because the parameters for the critic and action networks are only trained off-line (not on-line), and remain *fixed* during real-time control.

In contrast, the model reference indirect adaptive control (MRAC) shown in [5] depends on the outputs of the desired response predictor (DRP). The response of this controller therefore varies according to the design of DRP using information from the changing system outputs.

Also, the parameters for the ANN identifier and controller in indirect adaptive control must be updated on-line at every time step in order to force the plant outputs back to the response of the DRP.

The results of the indirect adaptive control scheme [5] are compared in Figs. 9 to 11 with the responses of the MDHPC and RDHPC. In these figures, the neurocontrollers designed by the indirect adaptive control (IAC) scheme using the MLPNN and RBFNN are called the MIAC and RIAC, respectively.

From Fig. 9, the DHP neurocontrollers still have a better damping performance compared to the MIAC and RIAC in the case of a severe disturbance (three phase short circuit).

On the other hand, the results in Figs. 10 and 11 show that the MIAC is less damped and slightly oscillatory with respect to the DHP controllers. The RIAC response lies between that of the MIAC and the DHP.

Whether the ACD family of controllers or the IAC family of controllers give better results for large or small disturbances, depends on choices such as the utility function for the ACD family, and the DRP for the IAC family. The purpose of this paper is not to claim that one of these families will always perform better than the other one, but to show that the ACD has a comparable performance to the IAC, even with fixed control parameters in real-time operation.

More detailed explanations of the indirect adaptive control and the ACD based optimal control methodologies for the synchronous generator control are explained in [5] and [6], respectively.

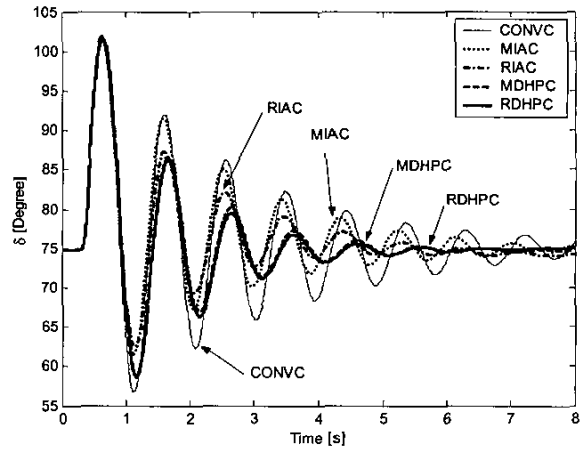


Fig. 9. Three phase short circuit test: Rotor angle.

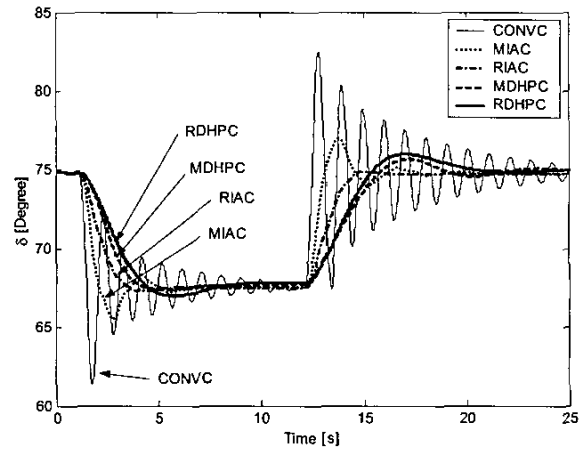


Fig. 10. $\pm 5\%$ Step changes in reference voltage of exciter: Rotor angle.

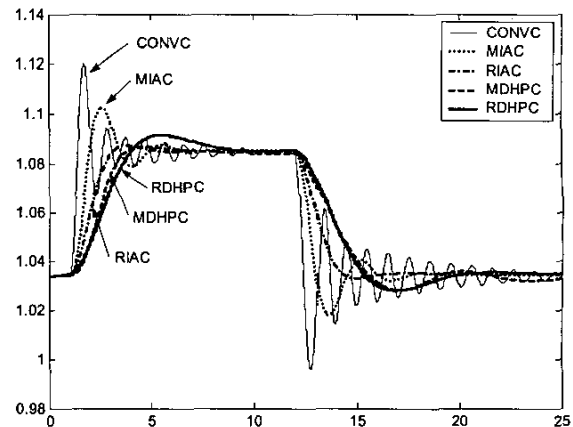


Fig. 11. $\pm 5\%$ Step changes in reference voltage of exciter: Terminal voltage

IV. CONCLUSIONS

This paper has presented the design of optimal neurocontrollers based on the heuristic dynamic programming (HDP) and dual heuristic programming (DHP) for the control of a synchronous generator in an electric power grid. To implement the HDP/DHP algorithm, the multilayer perceptron neural network (MLPNN) and radial basis function neural network (RBFNN) were used as function approximators. The comprehensive comparisons were carried out based on time-domain simulations to evaluate the dynamic transient and damping performances of the HDP/DHP based optimal neurocontrollers using the MLPNN/RBFNN. From the case study illustrated, the following conclusions can be drawn:

- When using the HDP, the RBFNN is preferred as function approximators for the model, critic, and action networks than the MLPNN.
- The DHP algorithm provides the effective dynamic and more robust control capability than the HDP.
- With DHP control designs, either the RBFNN or the MLPNN can be used as function approximators. However, the MLPNN is easy for hardware implementation because the RBFNN requires the feature extraction techniques to determine the centers of the RBF units, which is the most important characteristic of the RBFNN. In other words, the off-line computation to determine the centers of the RBF units must be paid the careful attention; otherwise the on-line updates for the centers of the RBF units require the highly expensive computational efforts.

Generally, the proposed neurocontrollers have *fixed parameters* for their model, action, and critic neural networks, which are trained off-line based on the infinite horizon optimal control approach. This means that there are no adaptive parameters in a real-time operation. Therefore, they provide a robust feedback with a powerful dynamic control capability under uncertain environments, and the possible instability issue associated with artificial neural networks (ANNs) based controllers can be avoided.

Investigations are continuing into more detailed treatment of different optimality conditions according to approximations for the value iteration J in the HDP and DHP.

ACKNOWLEDGMENT

Financial support by the National Science Foundation (NSF), USA under Grant No. ECS-0080764 for this research is gratefully acknowledged.

V. REFERENCES

- [1] W.T. Miller, R.S. Sutton, and P.J. Werbos, "Neural Networks for Control," MIT Press, Cambridge, Massachusetts, 1990, ISBN 0-262-13261-3.
- [2] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. White and D. Sofge, Eds. New York: Van Nostrand Reinhold, pp.493-526, 1992.
- [3] D.V. Prokhorov, and D.C. Wunsch, "Adaptive critic designs," *IEEE Trans. on Neural Networks*, Vol.8, No.5, pp. 997-1007, Sept. 1997.
- [4] G.K. Venayagamoorthy, R.G. Harley, and D.C. Wunsch, "Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator," *IEEE Trans. on Neural Networks*, Vol.13, No.3, pp. 764-773, May. 2002.
- [5] Jung-Wook Park, R.G. Harley, and G.K. Venayagamoorthy, "Comparison of MLP and RBF Neural Networks Using Deviation Signals for Indirect Adaptive Control of a Synchronous Generator," in *Proc. of the International Joint Conference on Neural Networks, IJCNN02*, Hawaii, pp. 919-924, May 2002.
- [6] Jung-Wook Park, R.G. Harley, and G.K. Venayagamoorthy, "Adaptive Critic Based Optimal Neurocontrol for Synchronous Generator in Power System Using MLP/RBF Neural Networks," *37th IEEE IAS Annual Meeting*, Vol. 2, pp.1447-1454, October 2002. (appear in *IEEE Trans. on Industry Applications*, 2003.)
- [7] Dimitri P. Bertsekas, "Dynamic Programming and Optimal Control", Athena Scientific, Belmont, Massachusetts, 2001, ISBN 1-886529-08-6.
- [8] Derong Liu, Xiaoxu Xiong, and Zhang Yi, "Action-Dependent Adaptive Critic Designs," in *Proc.2001 IEEE IJCNN*, Washington DC, Vol.2, pp. 990-995, July 2001.
- [9] S.N. Balakrishnan and Han Dongchen, "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Trans. on Control System Technology*, Vol.10, No.4, pp. 481-489, July. 2002.
- [10] Jennie Si and Yu-Tsung Wang, "On-Line Learning Control by Association and Reinforcement," *IEEE Trans. on Neural Networks*, Vol.12, No.2, pp. 264-276, March. 2001.
- [11] Timothy Draelos, David Duggan, Michael Collins, and Donald Wunsch, "Adaptive Critic Designs for Host-Based Intrusion Detection," in *Proc. of the IJCNN02*, Hawaii, pp. 1720-1724, May 2002.
- [12] G.K. Venayagamoorthy, R.G. Harley, and D.C. Wunsch, "Dual heuristic programming excitation neurocontrol for generators in a multimachine power system," appear in *IEEE Trans. on Industry Applications*, March/April. 2003.
- [13] G.K. Venayagamoorthy, R.G. Harley, and D.C. Wunsch, "Adaptive Critic Based Neurocontroller for Turbogenerators with Global Dual Heuristic Programming," in *Proc.2002 IEEE PES Winter Meeting*, Singapore, Vol.1, pp. 291-294, January 2000.
- [14] Chun-shin Lin, Yi-Hsun Ethan Cheng, and Hyongsuk Kim, "Radial Basis Function Networks for Adaptive Critic Learning," in *Proc.1994 IEEE WCCI IJCNN*, Orlando, Vol.2, pp. 903-906, July 1994.
- [15] Jung-Wook Park, R.G. Harley, and G.K. Venayagamoorthy, "Comparison of MLP and RBF Neural Networks Using Deviation Signals for On-Line Identification of a Synchronous Generator," in *Proc.2002 IEEE PES Winter Meeting*, New York, Vol.1, pp. 274-279, January 2002.
- [16] Z. Uykan, C. Guzelis, M.E. Celebi, and H.N. Koivo, "Analysis of input-output clustering for determining centers of RBFN," *IEEE Trans. on Neural Networks*, Vol.11, No.4, pp. 851-858, July 2000.
- [17] W.L. Brogan, "Modern Control Theory," Prentice Hall, Englewood Cliffs, New Jersey, 1991, ISBN 0-13-589763-7.
- [18] John Gregory and Cantian Kin, "Constrained Optimization in the Calculus of Variations and Optimal Control Theory," Van Nostrand Reinhold, New York, 1992, ISBN 0-442-00722-1.