Iterative Learning Heuristic Dynamic Programming (ILHDP) design of a Steam Power Plant Controller

Udhay Ravishankar, *Member IEEE*, and Milos Manic, *Member IEEE*
University of Idaho, Idaho Falls
ravi4736@vandals.uidaho.edu, misko@ieee.org

***Abstract*-This paper presents a new dynamic programming method called the Iterative Learning Heuristic Dynamic Programming (ILHDP). The ILHDP is an Iterative Learning Control (ILC) based Neural Dynamic Programming (NDP) algorithm. The NDP aspect of the ILHDP algorithm is borrowed from traditional Adaptive Critic Design (ACD) algorithms. Typical NDP algorithms in the ACD class of algorithms train a Model Network beforehand and use a Critic Network, as the gradient approximator, trained back-and-forth with the Action Network in each iteration to converge the Action Network towards the optimal control policy. The proposed ILHDP algorithm updates the Model Network continually based on newly obtained data sampled during each Action Network optimization step on the same experiment. This process of Model Network updation ensures better gradient approximation presented by the Model Network itself. The presented ILHDP is used for the design of a Steam Power Plant controller with respect to the Active-Power-to-Frequency droop characteristics. Test results indicated that the ILHDP designed controller was capable of stabilizing the output power of the Steam Power Plant to track the load with a maximum tracking error of 0.011 for abrupt load changes as fast as 15s. The Steam Power Plant was also subjected to large transient spikes for which the designed controller proved to recover the system back to stability.**

I. INTRODUCTION

Power Grid optimization is a growing field in the realm of Smart Grid Research and Development. The demand for optimal control has heightened as the Electric Power Grid has become more complex and unmanageable under harsh load conditions. Examples of work done in the field of optimal control for Smart Grids can be found in [1] – [3]. Beyond the realm of Smart Grids, optimal control is also becoming popular in the field of Robotics, Industrial Processes, Engine Control, etc, such as in [4] – [6].

The most popular optimal controller design methods belong to a class of Adaptive Critic Design (ACD) algorithms introduced by Werbos in [7]. The reason for their popularity is that they are neural network based and hence the complex mathematics behind dynamic programming can be approximated using simple neural network properties. In a typical ACD setup, three neural networks are used that are connected in cascaded fashion starting with an Action Network followed by a Model Network and then a Critic Network. The Action Network is the approximation of the optimal control policy, while the Model Network is the approximation of the concerned system dynamics and the Critic Network is the approximation of the Hamilton-Jacobi-Bellman (HJB) equation, or its gradient, typically found in dynamic programming literature [8] – [9].

In typical ACD algorithms, the Model Network is trained beforehand from previously sampled data after which the Critic and Action Network are trained back-and-forth optimizing the Action Network in the process. The training of the neural networks is performed using the Error-Backpropagation (EBP) algorithm introduced by Werbos in 1974 [10]. The theoretical details of the different ACD dynamic programming algorithms can be found in [11]. Other developments in ACD include greedy HDP iteration method to optimize the Action Network. Examples of work published using the greedy HDP iteration method can be found in [12] – [14].

Iterative Learning Control (ILC) is also a popular optimal control design method. Its principle is based on improvising the controller using previously learned information about the system iteratively. While ACD is purely neural network based, ILC is more analytical. Examples of work published in this field can be found in [15] – [16].

The proposed ILHDP is a purely neural network optimal control design algorithm based on the principle of ILC. In the proposed ILHDP algorithm, the Model Network is continuously updated from newly obtained data sampled during the Action Network optimization process. This process of continual Model Network updation ensures better gradient approximation presented by the Model Network itself.

The ILHDP algorithm was applied to the design of an optimal Steam Power Plant neural controller. The Steam Power Plant is modeled as a third order linear system that converts power from the boiler to mechanical power on the turbine. The neural controller acts as an Automatic Generation Controller (AGC) that regulates the output power to track the load.

The rest of this paper proceeds as follows: Section II discusses the background overview of Iterative Learning Control (ILC) and Adaptive Critic Design (ACD) followed by the introduction the ILHDP algorithm in Section III. The Steam Power Plant model and its associated Grid model for loading conditions are introduced in Section IV along with the ILHDP design implementation of the Steam Power Plant controller. Test results are discussed in Section V and the paper finally concludes with future work in Section VI.

II. Background Overview of ILC and ACD

This section provides a brief overview of ILC and ACD that form the basis of the proposed ILHDP algorithm.

A. *Iterative Learning Control (ILC)*

ILC is an optimal control algorithm that improves the tracking response iteratively from previously learned information about the system. Assume a discrete-time linear dynamic system of the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

which can be rewritten as

$$y(t+1) = Ay(t) + CBu(t) \tag{2}$$

In the discrete-domain, i.e. the z-domain, eq. (2) can be written as

$$Y(z) = H(z)U(z), \quad H(z) = \frac{CB}{z-A} \tag{3}$$

Let $y_d(t)$ be the desired tracking signal and the corresponding error signal be denoted as $e(t) = y_d(t) - y(t)$. Then the optimal control signal $u^*(t)$ is derived as

$$\begin{aligned} U^*(z) &= U(z) + \frac{E(z)}{H(z)} \\ U^*(z) &= U(z) + K(z)E(z) \\ \Rightarrow u^*(t) &= u(t) + k(t) * e(t) \end{aligned} \tag{4}$$

It is clear from eq. (4) that the previously learned system information plays a vital role in the control signal optimization. The iterative equation, derived from eq. (4), can be written as

$$u_{i+1}(t) = u_i(t) + \sum_{j=0}^{r} k(j) z^{-j} e_i(t) \tag{5}$$

B. *Adaptive Critic Design (ACD)*

ACD is a neural network based optimal control algorithm that improves the tracking response iteratively from previously learned information about the system. The ACD algorithm uses a Model Network to learn the system dynamics as a function $y(t+1) = f_M(y(t),u(t))$. Once the Model Network is trained, the control policy, $u(t) = f_A(e(t))$, or the Action Network is optimized based on a Critic Network that approximates the gradient of the cost function with respect to the system output $y(t)$. For a tracking control problem, the cost function is typically in the form of the HJB equation given as

$$J(t) = \sum_{k=0}^{\infty} \alpha^k \Gamma(t+k), \quad 0 \le \alpha \le 1 \tag{6}$$

where $\Gamma(t) = e^2(t)$. The control policy update equation for the ACD algorithm is then given as

$$u_{i+1}(t) = u_i(t) + \frac{\partial J_i(t+1)}{\partial y_i(t+1)} \frac{\partial y_i(t+1)}{\partial u_i(t)} \tag{7}$$

III. Iterative Learning Heuristic Dynamic Programming (ILHDP)

From the previous section, both the ILC and ACD algorithms utilized previously learned information about the system to update the control policies towards optimality. Hence modeling the system dynamics is a vital part in optimizing the control policy. The more accurate the modeling, the better are the chances of reaching the true optimal control policy.

The ILHDP algorithm combines the attributes of the ILC and ACD algorithms. The ILHDP algorithm is a gradient-based optimal control algorithm, like the ACD algorithm, that continually updates its knowledge about the system dynamics during control policy optimization process. In the following subsection, the ILHDP algorithm will be developed for linear dynamic systems.

A. *ILHDP for Linear Systems*

Continuing from eq. (4), the z-transform of the optimal control signal, $U^*(z)$,

$$\begin{aligned} K(z) &= \frac{1}{H(z)} = \frac{z-A}{CB} \\ \Rightarrow k(t) * e(t) &= \frac{1}{CB}\big(e(t+1) - Ae(t)\big) \end{aligned} \tag{8}$$

This means that linear dynamic systems do not need high orders, r, to optimize the control signal or policy. Using ACD approach the optimizing gradient can be written as

Pseudo Code of the ILHDP:

Let the Action Network be denoted as a function $f_A(y(t))$ and the Model Network as $f_M(y(t),u(t))$.

Phase I: Train Model Network

1) Obtain training data of $y(t)$ and $u(t)$ from simulation of the system under consideration.
2) Create a mapping $y(t+1) = f_M(y(t),u(t))$.

Phase II: Train Optimal Neural Controller

1) Initialize Action Network $f_A(e(t))$ where $e(t)$ is the error state.
2) Compute the gradient $\partial y(t+1)/\partial u(t)$ from the Model Network.
3) Compute the gradient $\partial J(t+1)/\partial u(t)$ according to eq. (9).
4) Update the control policy $u(t)$ using eq. (7).
5) Simulate the system with the new control policy.
6) Sample new data from the simulation.
7) Update the Model Network with new and previous data.
8) Repeat 2 to 7 until optimal policy is reached.

Fig. 1. Pseudo Code for the ILHDP algorithm.

$$\frac{\partial J(t+1)}{\partial u(t)} = \frac{\partial y(t+1)}{\partial u(t)} e(t+1) + \frac{\partial y(t)}{\partial u(t)} e(t) \qquad (9)$$

However, since the second term on the right-hand-side of eq. (9) is zero, only the first term can be used for the control policy optimization. The pseudo-code for the ILHDP algorithm is given in Fig. 1.

IV. Steam Power Plant and Grid Modeling

This section discusses the Steam Power Plant modeling and the Grid modeling for control.

A. Steam Power Plant Modeling

The Steam Power Plant model used in this paper was adopted from [17]. In this model, the generator turbine has three sections namely, the High Pressure (HP), the Intermediate Pressure (IP) and the Low Pressure (LP) sections. The steam leaving the boiler initially enters the HP section of the turbine. After partial expansion, the steam is directed back to the boiler to be reheated from the residual energy stored in the boiler, i.e. the boiler heat energy remaining after passing the steam into the HP steam chest. The reheated steam then flows into the IP section of the turbine where the steam is expanded again. On leaving the IP section of the turbine, the steam flows into the LP section of the turbine and finally to the condenser to complete the cycle. The individual turbine sections, i.e. the HP, IP and LP, contribute 30%, 40% and 30% of the total turbine torque respectively. The block diagram of the steam system is shown in Fig. 2. Fig. 3 shows the s-domain modeling of the steam flow mechanism just described. The P_m denotes the total mechanical power on the turbine from the steam flow. The gains α, β and γ, equal to 0.3, 0.4 and 0.3 respectively, represent the individual percentage contribution to total mechanical power on the turbine. The values of T_A, T_B and T_C chosen are 0.1s, 4s and 0.3s

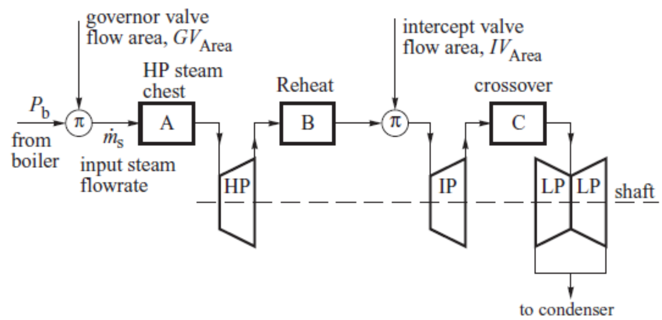

Fig. 2. Block Diagram of the steam flow. [17]

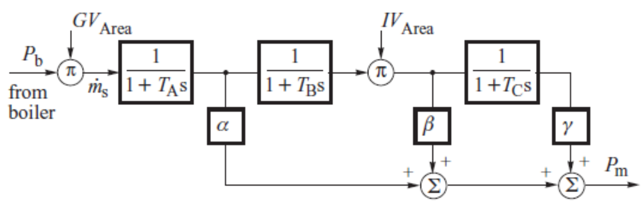

Fig. 3. S-domain modeling of the steam flow mechanism. [17]

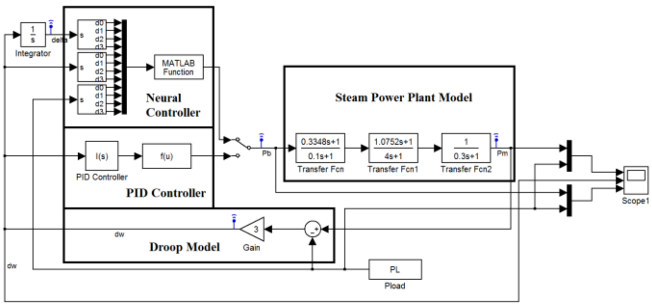

Fig. 4. Matlab/Simulink model of Steam Power Plant control.

respectively.

Typical Automatic Generation Controllers (AGC) control the power output of the plant based on droop characteristics. Droop characteristics dictate that the Active Power output (P) is related to the Frequency output of the plant (ω), while the Reactive Power output (Q) is related to the Voltage output of the plant (V). In this paper, we consider only the P-ω droop characteristics to control the Steam Power Plant. Therefore in addition to the Steam Power Plant model, the P-ω droop model was embedded. The total system, i.e. the Steam Power Plant model and the Droop model, was closed using an Integral (I) controller. This is shown in Fig. 4. All values taken in the power plant control model are normalized values with respect to the rated power outputs and frequency. In Fig. 4 the Droop Model is characterized by a 5% droop generator gain of three that outputs the actual frequency output on the transmission lines. The input to the Droop Model is the error between the electrical power output of the power plant (assuming it is equal to the mechanical power, P_m, on the turbine) and the required Active Load Power, P_{load}. The I controller then inputs the error between the actual output frequency of the plant and the reference frequency of 60 Hz, to output the required boiler power input to the steam chest consisting of the HP, IP and LP sections of the turbine.

B. Grid Modeling

The most important problem of an electric power grid is the load disturbance the grid is subjected to. It is this load disturbance that drives the grid into instability if not managed properly. In the electric power grid, this load does not only consist of the consumer loading but also consists of loading effects from other generations in the grid. Hence to evaluate

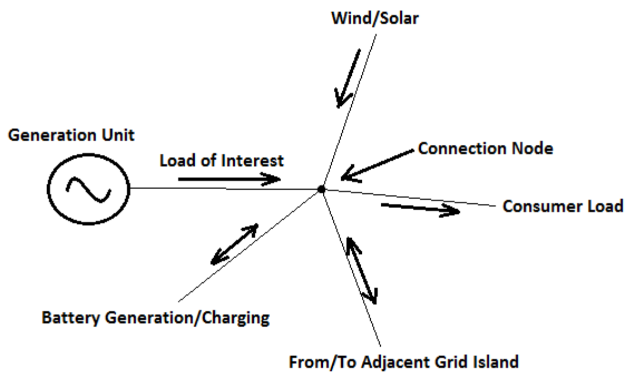

Fig. 5. Grid Model.

the controller's stability performance on its generation unit, it is important to subject the generation unit, i.e. the Steam Power Plant, to a load disturbance signal. Since a generation unit is connected to the grid at some random node, we only consider the loading effects from the generation unit to that particular node of the grid. The power flowing in and out of this node may come from Wind/Solar Generators, other adjacent grid islands, consumer load, battery generation/charging and etc. This is shown in Fig. 5

C. *ILHDP Design of the Steam Power Plant Controller*

The Steam Power Plant neural controller architecture is shown in Fig. 6. It is a Multilayer Perceptron with linear neurons in the hidden layer and a sigmoidal neuron in the output layer. The sigmoidal neuron in the output layer ensures that the boiler power input, $Pb(t)$, to the steam power system is within the rated limits. The inputs to the neural controller are the load-angle, $d(t)$, delta frequency (reference frequency minus the actual frequency), $dw(t)$ and the load power, $PL(t)$, along with their time-delayed values . The load signal, $PL(t)$, is included in the input because this signal acts as an extra uncontrollable stimulus to the Steam Power Plant. So the neural controller is trained to provide a control signal based on the current load state. Modern power grids are now being equipped with Phasor Measurement Units (PMU) that facilitate the load signal estimation. The Steam Power Plant can however estimate the load signal locally using the droop model equation as

$$\Delta\omega(t) = M_{gen}\Delta P(t)$$
$$\Rightarrow \Delta P(t) = \frac{1}{M_{gen}}\Delta\omega(t) \qquad (10)$$

with the droop constant, M_{gen}, as the generator's droop inertia. As mentioned earlier in Section IV, 5% generator droop was used.

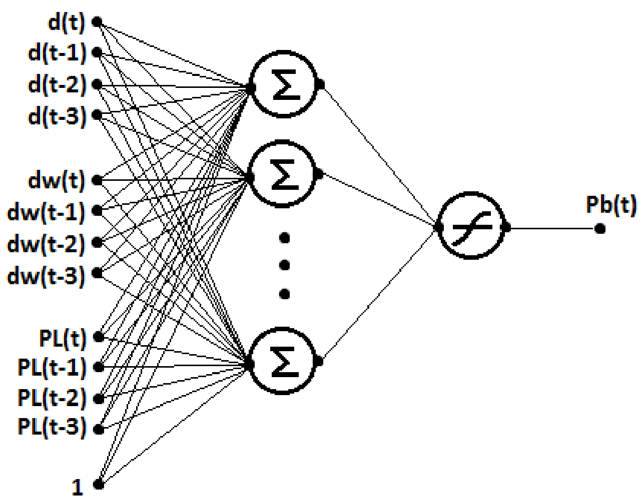

Fig. 6. Action Network used for training the optimal policy.

V. Test Results

This section shows the test results obtained from the ILHDP designed controller on the Steam Power Plant system. The controller was designed to output the control signals within the rated power limits in the per-unit (p.u.) system. The real time sampling rate of the system was taken as 0.2s.

A. *Controller response under abrupt load changes*

The Steam Power Plant system was subjected to a load signal with multiple abrupt changes in the load. The time intervals between each abrupt load change were chosen as 30s, 20s and 15s. The response of the controller under these load conditions are shown in Fig.s 7 to 9.

From Fig.s 7 to 9, it can be noted that the power output

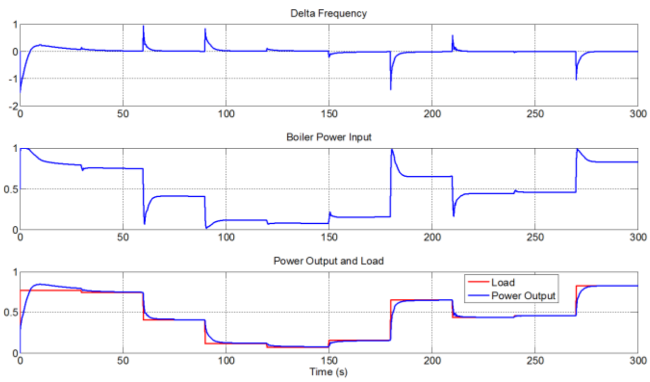

Fig. 7. ILHDP controller response for 30s abrupt load change condition.

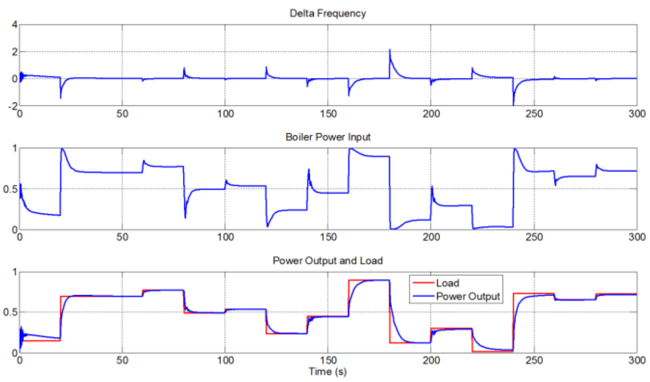

Fig. 8. ILHDP controller under 20s abrupt load change condition.

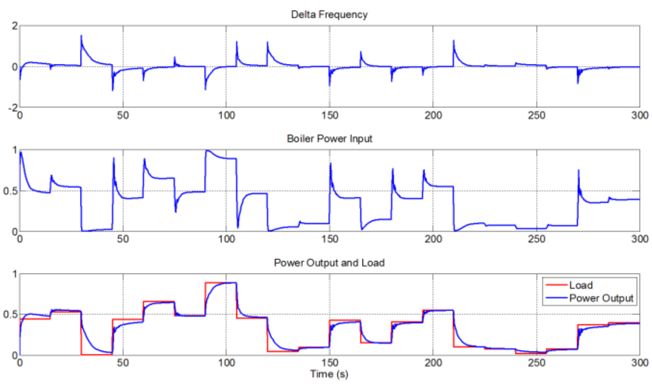

Fig. 9. ILHDP controller response for 15s abrupt load change condition.

slowly tracks the abrupt load changes. This is due to the fourth order characteristic of the Steam Power Plant and droop. The maximum tracking errors for the 30s, 20s and 15s abrupt load change conditions were 0.007, 0.0095 and 0.011 respectively.

B. Controller response under Gaussian noisy load

To simulate the loading effects of the grid on Steam Power Plant as shown in Fig. 5, a Gaussian random noise signal was added to the abrupt load change signal as discussed in the previous subsection. The Gaussian random noise was chosen since typical natural systems exhibit such characteristics. The probability density function (pdf) of a Gaussian Distribution is given by,

$$f(x,\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (5)$$

In Eq. (5), μ and σ are the parameters of the Gaussian pdf known as the mean and the variance respectively. The mean of the Gaussian random noise signal added to the abrupt load change signal is zero, but variances of 0.05 and 0.1 were chosen. The response of the controller under these load conditions are shown in Fig.s 10 and 11.

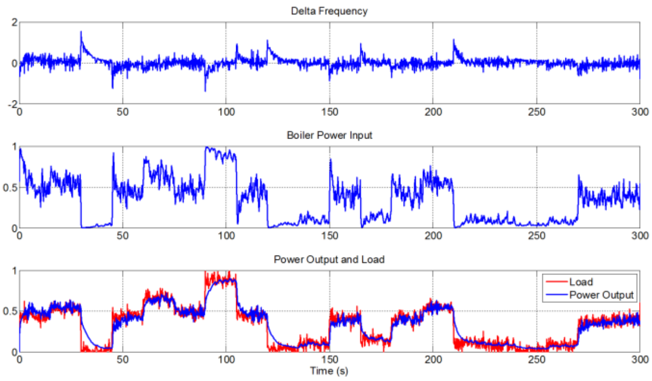

Fig. 10. DAHDP controller response for 15s abrupt load change and Gaussian random noise variance of 0.05.

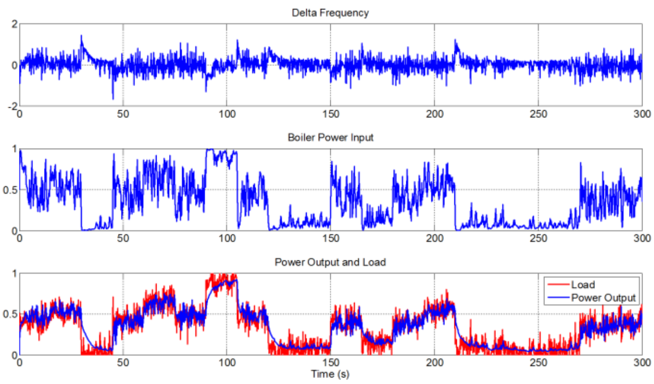

Fig. 11. DAHDP controller response for 15s abrupt load change and Gaussian random noise variance of 0.1.

C. Controller response under transient spikes

The Steam Power Plant system was subjected to a single transient spike within the data window. The transient spike

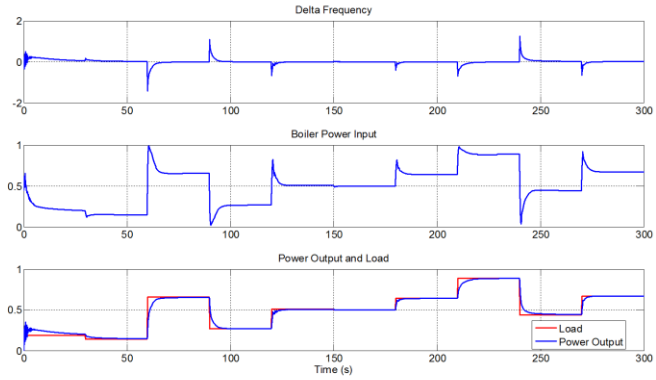

Fig. 12. ILHDP controller response for no transient spike in the load.

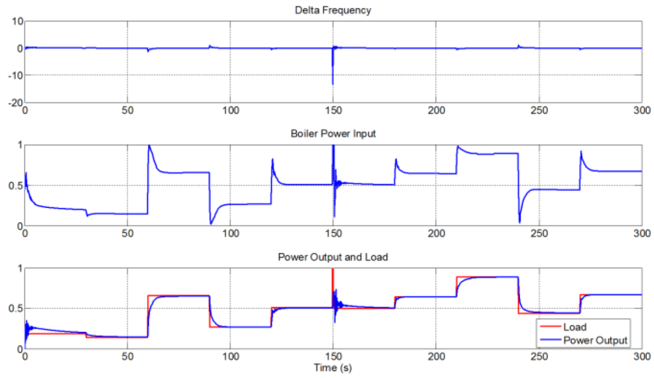

Fig. 13. ILHDP controller response for a transient spike magnitude of 5 times rated power.

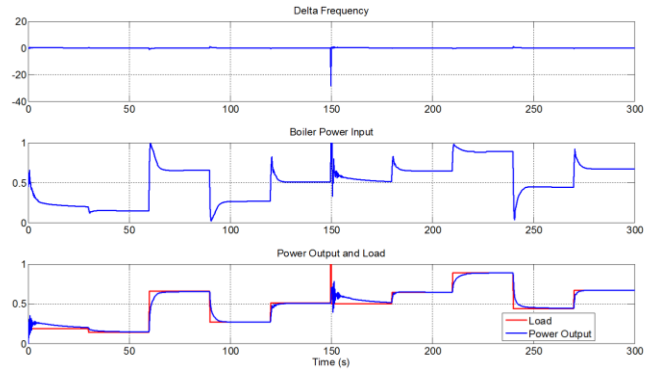

Fig. 14. ILHDP controller response for a transient spike magnitude of 5 times rated power.

magnitudes chosen were 5 and 10 times the rated power of the Steam Power Plant. The controller's response under the chosen transient spike conditions are shown in Fig.s 13 and 14. The transient spike responses are compared with a no transient spike case shown in Fig. 12. The purpose of this subjection was to study the how well the designed neural controller can bring the system back to stability under large spikes in the load. In a real electric grid, these transient spikes may occur due to lightning strikes or solar flares.

It can be noted from Fig.s 13 to 15 that when the transient spike occurs *t=150s* the drop in the delta frequency increases as the magnitude increases. This increase in delta frequency urges the neural controller to boost the boiler power input into the system to quickly stabilize it to track the actual load reference signal.

VI. Conclusion and Future Work

The Iterative Learning Heuristic Dynamic Programming (ILHDP) algorithm was introduced in this paper and was applied to the design of the Steam Power Plant neural controller. The control in this paper considered only the Active-Power-to-Frequency (P-ω) droop characteristics. The designed controller was able to stabilize the power output of the Steam Power Plant with a load tracking error no more than 0.011 for abrupt load changes as fast as 15s. The load signal was also subjected to Gaussian variance of about 0.1. The Steam Power Plant was further tested under the subjection of large transient spikes and the designed controller proved to bring the power plant back to stability.

For future work, the Reactive-Power-to-Voltage (Q-V) droop characteristics will be tackled using the ILHDP algorithm. This completes the control requirements of the Steam Power Plant system with the currently designed Active-Power-to-Frequency (P-ω) droop neural controller. Similar controllers can be built for various other types of power plants.

Acknowledgment

The authors acknowledge support for this work from Idaho National Laboratory through the U.S. Department of Energy Office of Electrical Delivery and Energy Reliability under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

References

[1] R. L. Welch and G. K. Venayagamoorthy, “HDP based Optimal Control of a Grid Independent PV System,” *IEEE Power Engg. Society General Meeting*, 2006, pp. 1 – 6.

[2] S. Mohagheghi, G. K. Venayagamoorthy and R. G. Harley, “Adaptive Critic Design based Neuro-Fuzzy Controller for a Static Compensator in a Multimachine Power System,” *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1744 – 1754, Nov. 2006.

[3] S. Ray, G. K. Venayagamoorthy, B. Chaudhuri and R. Majumder, “Comparison of Adaptive Critic based and Classical Wide-Area Controller for Power Systems,” *IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 38, no.4, pp. 1002 – 1007, Aug. 2008.

[4] W-S Lin and P-C Yang, “DHP Adaptive Critic Motion Control of Autonomous Wheeled Mobile Robot,” *Proc. IEEE Symposium on Approx. Dynamic Programming and Reinforcement Learning*, 2007, pp. 311 – 317.

[5] B. Yang and D. Cao, “Action-Dependent Adaptive Critic Design based Neurocontroller for Cement Preclaciner Kiln,” *Intl. Journal on Computer and Information Security*, pp. 60 – 67, 2009.

[6] H. Javaherian, D. Liu, Y. Zhang and O. Kovalenko, “Adaptive Critic Learing Techniques for Automotive Engine Control,” *Proc. American Control Conf.*, 2004, pp. 4066 – 4071.

[7] P. Werbos, “A menu of designs for reinforcement learning over time,” in *Neural Networks for Control*. Cambridge, MA: MIT Press, 1990, pp. 67 – 95.

[8] J. Si, A. G. Barto, W. B. Powell and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. New York: Wiley, Jul. 2004.

[9] S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, 2006.

[10] P. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” Ph.D. dissertation, Committee on Applied Mathematics, Harvard Univ., Cambridge, MA, 1974.

[11] D. V. Prokhorov and D. C.Wunsch, II, “Adaptive critic designs,” *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 997 – 1007, Sep. 1997.

[12] A. Al-Tamimi, F. L. Lewis and M. A-Khalaf, “Discrete-Time HJB Solution using Approximate Dynamic Programming: Convergence Proof,” *IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 38, no. 4, pp. 943 – 949, Aug. 2008.

[13] H. Zhang, Q. Wei and Y. Luo, “A Novel Infinite-Time Optimal Tracking Control Scheme for a Class of Discrete-Time Nonlinear Systems via the Greedy HDP Iteration Algorithm,” *IEEE Trans. on Systems, Man and Cybern. – Part B: Cybern.*, vol. 38, no. 4, pp. 937 – 942, Aug. 2008.

[14] D. Liu, D. Wang and D. Zhao, “Adaptive Dynamic Programming for Optimal Control of Unknown Nonlinear Discrete-Time Systems,” *IEEE Symposium on Approx. Dynamic Programming and Reinforcement Learning*, 2011, pp. 242 – 249.

[15] D. Meng, Y. Jia, J. Du and F. Yu, “Data-Driven Control for Relative Degree Systems via Iterative Learning,” *IEEE Trans. on Neural Networks*, vol. 22, no. 12, pp. 2213 – 2225, Dec. 2011.

[16] D. Shen and Z. Hou, “Iterative Learning Control with Unknown Control Direction: A Novel Data-Based Approach,” *IEEE Trans. on Neural Networks*, vol. 22, no. 12, 2237 – 2249, Dec. 2011.

[17] J. Machowski, J. W. Bialek and J. R. Bumby, *Power System Dynamics: Stability and Control*. Wiley, 2008.