No.30- (3) 2021 Journal of Petroleum Research & Studies (JPRS)

Self-Learning Controllers in the Oil and Gas Industry

Dr. Seaar Al-Dabooni*, Hussen Ali Mohammad Alshehab

Basra Oil Company (BOC)

*Corresponding Author E-mail: sjamw3@mst.edu,

Soc.engin330@boc.oil.gov.iq

Abstract

Recently, solving the optimization-control problems by using artificial intelligence has widely appeared in the petroleum fields in exploration and production. This paper presents the state-of-the-art reinforcement-learning algorithm applying in the petroleum optimization-control problems, which is called a direct heuristic dynamic programming (DHDP). DHDP has two interactive artificial neural networks, which are the critic network (provider a critique/evaluated signal) and the actor network (provider a control signal). This paper focuses on a generic on-line learning control system in Markov decision process principles. Furthermore, DHDP is a model-free learning design that does not require prior knowledge about a dynamic model; therefore, DHDP can be appllied with any petroleum equipment or devise directly without needed to drive a mathematical model. Moreover, DHDP learns by itself (self-learning) without human intervention via repeating the interaction between an equipment and environment/process. The equipment receives the states of the environment/process via sensors, and the algorithm maximizes the reward by selecting the correct optimal action (control signal). A quadruple tank system (QTS) is taken as a benchmark test problem, that the nonlinear model responses close to the real model, for three reasons: First, QTS is widely used in the most petroleum exploration/production fields (entire system or parts), which consists of four tanks and two electrical-pumps with two pressure control valves. Second, QTS is a difficult model to control, which has a limited zone of operating parameters to be stable; therefore, if DHDP controls on QTS by itself, DHDP can control on other equipment in a fast and optimal manner. Third, QTS is designed with a multi-input-multi-

output (MIMO) model for analysis in the real-time nonlinear dynamic system; therefore, the QTS model has a similar model with most MIMO devises in oil and gas field. The overall learning control system performance is tested and compared with a proportional integral derivative (PID) via MATLAB programming. DHDP provides enhanced performance comparing with the PID approach with 99.2466% improvement.

متحكمات التعلم الذاتي في صناعة النفط والغاز

الملخص

في الآونة الاخيرة، تم استعمل الذكاء الاصطناعي بشكل واسع في حل المشاكل في الصناعات النفطية سواء كانت في التنقيب او في الانتاج. هذا البحث يستعمل احدث الخوارزمية متخصصة بالذكاء الاصطناعي في التعلم الاجباري لحل مشاكل و السيطرة على المعدات النفطية، حيث ان هذه الخوارزمية تسمى خوارزمية البرمجة الديناميكية المباشرة. خوارزمية البرمجة الديناميكية تتكون من شبكتين عصبيتين احدهما تسمى الشبكة الناقدة و التي تجهز اشارة تقيم الاداء، اما الشبكة الثانية تسمى الشبكة الممثل او المسيطر والتي تجهز اشارة السيطرة للمعدات المراد السيطرة عليها.

أن خوارزمية البرمجة الديناميكية مصممة لتعليم المسيطرات بدون معرفة النموذج الرياضي للمعدات أو الاجهزة و هي تتعلم بشكل مباشر بدون التدخل البشري و بحسب مبادئ نظرية ماركوف. اي يمكن لهذا الخوارزمية ان تتكيف للسيطرة على المعدات مع تغير الظروف و ضمن شروط البيئية المختلفة و بدون الرجوع الى البشر، اي يكون التعلم ذاتي وذلك من خلال تكبير اشارة الربح (او تقليل اشارة الخطأ). في هذا البحث تم اختيار نموذج صعب السيطرة و الذي يسمى نظام الرباعي الخزانات لثلاث اسباب. اولا: ان نظام الرباعي الخزانات يوجد في معظم المجالات النفطية (النظام بشكل كامل او اجزء النظام). حيث يتكون من اربع خزانات مربوطة سوية مع مضختين كهربائيتين و عدد من الصمامات ابرزها صمامان ذي تحكم هوائي-كهربائي. ثانيا: ان نظام الرباعي الخزانات هو نظام معقد صعب السيطرة علية. لذلك اذا خوازمية البرمجة الديناميكية تمكنت من السيطرة على النظام، فيكون من السهولة للخوارزمية ان تسيطر على اي نظام ذي نموذج اسهل. ثالثا: ان نظام الرباعي الخزانات هو نظام متعدد الادخالات و الاخراجات، لذلك هو يوائم معظم الاجهزة و المعدات النفطية ذات ادخالات و اخراجات متعددة. هذا البحث تم اختباره مع اشهر مسيطر والذي يستعمل حاليا في المعامل و المنشئات النفطية او الصناعية هو مسيطر النسبي-التكاملي-التفاضلي وتم اختباره و مقارنته مع خوارزمية البرمجة الديناميكية و التي تم تنفيذها بواسطة برنامج الماتلاب (حيث ان مخرجات الموديل اللاخطي مقاربة جدا للاختبارات العملية الواقعية). حيث تبين ان خوارزمية البرمجة الديناميكية تسيطر على النموذج رباعي الخرانات بشكل افضل و اسرع من حيث عدم تجاوز القيم المطلوبة و بفترة زمنية قصيرة وكل هذا تتم من خلال التعلم الذاتي (التعلم من الصفر و بدون التدخل البشري). حيث ان النتائج المستحصلة بأستعمال خوارمية البرمجة الديناميكية قد تحسنت الى 99.2466% مقارنة مع مسيطر النسبي-التكاملي –التفاضلي.

I. Introduction

Approximate dynamic programming (ADP) is useful tool to overcome a behavior of nonlinear systems [1]. ADP has three categorizes [2]: heuristic dynamic programming (HDP), dual heuristic programming (DHP) and globalized DHP. ADP has two neural networks: actor and critic to provide optimal control signal and the long-cost value, respectively. If the action-dependent (AD) form is used in ADP (ADHDP for HDP and ADDHP for DHP). ADP is used in many real applications. For instance, [3] presents how control on turbo-generator. [4] shows the ability of DHP to solve swarm robot problems. [5] and [6] illustrated that ADHD P can obtain an optimal path by multi-robot navigation. Recently, [7] and [8] are used with Atari game to solve many hard problem with huge number of states. All previous ADP approaches are used temporal difference learning algorithm based on Markov decision process. A Markov Decision Process contains a set of model states, a set of actions, and a reward or cost function and system model. The core of Markov decision process is to find a sequence of actions for certain state that make the cost low or long-go reward high. The main purpose and aim of this paper is how using the HDP approach to control on a process of a quadruple-tank system (QTS), which is frequently used in oil and gas industrial. QTS consists of four interconnected tanks and two motor-pumps [9]. HDP is used to control voltage of two pumps to follow the desired level (set point level value) of tanks, which is a first approach appearing in the literature. This paper presents a self-learning algorithm to build a controller from scratch without human intervention to control on tanks level of QTS.

II. Devices and experiments

This section presents the aspects of HDP as in [2] and [6] with details of learning of the nonlinear QTS model as in [9].

A. Architecture of The HDP approach

The main block diagram for the featured DHDP illustrates in Figure (1).

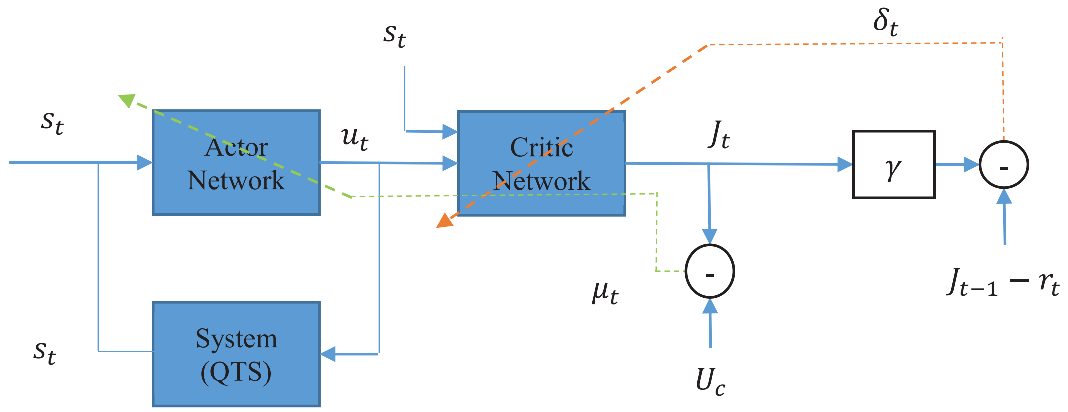

Fig. (1) Block diagram for HDP. u_t is the action vector at time t to control on the motor-pumps of QTS that comes from actor neural network (the controller), while the value function (J_t), which is single long - cost value, comes from critic neural network. s_t is the input states vector at time t, which is represented by the tank levels. A reinforcement function (r_t) can get from linear quadratic equation. The backpropagation learning path is shown by dashed lines for actor and critic networks.

As shown in Figure (1), the model produces a prediction of the next state and next reward. HDP uses to solve the Bellman's optimality equation, which is written as [6].

$$J^*(\boldsymbol{s},u) = P^u_{\boldsymbol{ss'}}\left(r_t + \gamma \max_{u\in\mathcal{A}} J^*(\boldsymbol{s'},u)\right) \qquad (1)$$

according to Markov decision process principles, the $J^*(\boldsymbol{s},u)$ is the optimal value function of the current state $\boldsymbol{s}$; $P^u_{\boldsymbol{ss'}}$ is the transition probability to move to the next state $\boldsymbol{s'}$ with action, u, that belong to $\mathcal{A}$, (in this paper, $P^u_{\boldsymbol{ss'}} = 1$) and γ is the discount factor, which is between 0 and 1. Therefore, The Bellman's optimality equation obtains as follows:

$$J^*(\boldsymbol{s},u) = \max_{u\in\mathcal{A}} [r(\boldsymbol{s},u) + \gamma J^*(\boldsymbol{s'},u)] \qquad (2)$$

The optimal control $u^*(s)$ is given as follows:

$$u^*(\boldsymbol{s}) = \underset{u\in\mathcal{A}}{argmax} [r(\boldsymbol{s},u) + \gamma J^*(\boldsymbol{s'},u)]. \qquad (3)$$

As shown in [6], DHDP consists of blocks called the action network and critic network. It also uses online learning for the neural networks. The control signal is generated from actor neural network (controller), which is evaluated by the critic neural network. Both critic and actor have one hidden layer. The temporal difference error for the critic network is defined as:

$$\delta_t = J_{t-1} - (r_t + \gamma J_t). \qquad (4)$$

And

$$E_t^c = \frac{1}{2}\delta_t^2. \qquad (5)$$

The gradient-based adaptation for the weights update rule in the critic network can be given by

$$w_{t+1}^c = w_t^c + \Delta w_t^c, \qquad (6)$$

$$\Delta w_t^c = l_t^c\left[-\frac{\partial E_t^c}{\partial w_t^c}\right], \qquad (7)$$

$$\frac{\partial E_t^c}{\partial w_t^c} = \left[\frac{\partial E_t^c}{\partial J_t}\frac{\partial J_t}{\partial w_t^c}\right], \qquad (8)$$

Where, ℓ_t^c is the learning rate of the critic network at time *t*, and (w_t^c) is the weight vector in the critic network.

Fig. 2 illustrates the critic's neural network structure. The weight updates from hidden to output layer (Δw_t^{c2}) according to backpropagation rules are:

$$\Delta w_t^{c2} = -\ell_t^c \gamma \delta_t p^T, \qquad (9)$$

While, the weights updating from input to hidden layer (Δw_t^{c1}) are:

$$\Delta w_t^{c1} = -\ell_t^c 0.5\left[Id(n_c) - diag\left(p_j^2\right)\right][\gamma\delta_t w_t^{c2}]^T[In] \qquad (10)$$

where n_c is the total number of hidden nodes in the critic network; $p_j = \sigma\left(q_j\right)$ is the j output of the hidden nodes $q, p \in \Re^{n_c}$; $\sigma(.)$ is the sigmoid function; In is the row vector for total number of inputs to the critic network which consists of n input states concatenated with m control signals; $In \in \Re^{(n+m)}$; $Id(.)$ is the identity matrix, $diag(.)$ is a diagonal matrix.

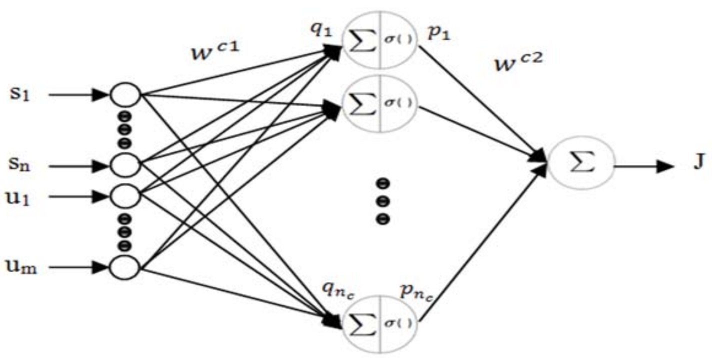

Fig. (2) Critic multilayer perceptron neural network structure (Sigmoid function is applied only for hidden nodes) for hidden nodes.

As shown in Figure (1), the error between the desired ultimate objected ($U_c = 0$) to minimize the actor error (see [2]) and the approximate value function (J_t) is backpropagated through critic network. The error function of an action network can be defined as

$$\mu_t = J_t - U_c. \quad (11)$$

Therefore, the objective function in the action network is

$$E_t^a = \frac{1}{2}\mu_t^2. \quad (12)$$

The weight updating in the action network is given as follows:

$$w_{t+1}^a = w_t^a + \Delta w_t^a, \quad (13)$$

$$\Delta w_t^a = \ell_t^a\left[-\frac{\partial E_t^a}{\partial w_t^a}\right], \quad (14)$$

$$\frac{\partial E_t^a}{\partial w_t^a} = \left[\frac{\partial E_t^a}{\partial J_t}\frac{\partial J_t}{\partial u_t}\frac{\partial u_t}{\partial w_t^a}\right], \quad (15)$$

Where ℓ_t^a the learning rate of the action is network at time t, and w_t^a is the weight vector in the action network.

Figure (3) illustrates the actor network. The weight updates from hidden to output layer (Δw_t^{a2}) according to backpropagation rules are:

$$\Delta w_t^{a2} = -\ell_t^a \mu_t [w_t^{c2}]\left(0.5\left[Id(n_c) - diag\left(p_j^2\right)\right]\right)[w_{ca}](0.5[Id(m) - diag(u_i^2)])g^T, \quad (16)$$

While, the weights updating from input to hidden layer (Δw_t^{a1}) are:

$$\Delta w_t^{a1} = -\ell_t^a \mu_t [w_t^{c2}]\left(0.5\left[Id(n_c)\ diag\left(p_j^2\right)\right]\right)[w_{ca}](0.5[Id(m) diag(u_i^2)])[w_t^{a2}] \times$$

$$\left(0.5\left[Id(n_a) - diag\left(g_j^2\right)\right]^T\right) s_t, \quad (17)$$

where n_a is the number of hidden neurons; u_j is the jth output from action network; w_{ca} is the weight values which are associated with the input states from the action network, $w_{ca} \in \Re^{n_c \times (n+1:n+m)}$ from w_t^{c1}; g_j is the jth output of the hidden nodes of the action network, $g \in \Re^{n_a \times 1}$. Both critic and action learning rate decrease with time until a certain small value as we present in the result section.

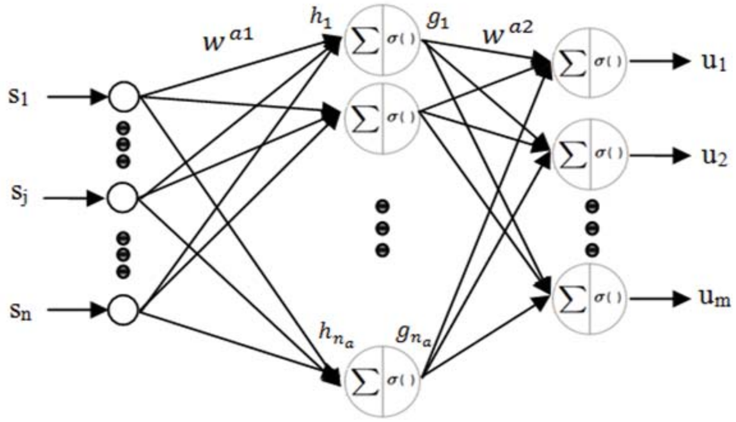

Fig. (3) Action multilayer perceptron neural network structure (sigmoid function is applied for all nodes)

B. *Architecture of The QTS approach*

Figure (4) illustrates a schematic diagram of the QTS. Authors in [9] derived accurate mathematical model based on both physical and experimental data. They demonstrates that the outputs from the model and the outputs from the real process are closed in various situations. Two pumps is used to control on the level in the lower two tanks by input voltages ($\upsilon_1$and υ_2). The voltage from level measurement devices are represented the output ($y_1$and y_2). Low of Bernoulli and mass balances deferential equations are given as follows [9]:

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}\upsilon_1,$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}\upsilon_2, \qquad (18)$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}\upsilon_2,$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}\upsilon_1,$$

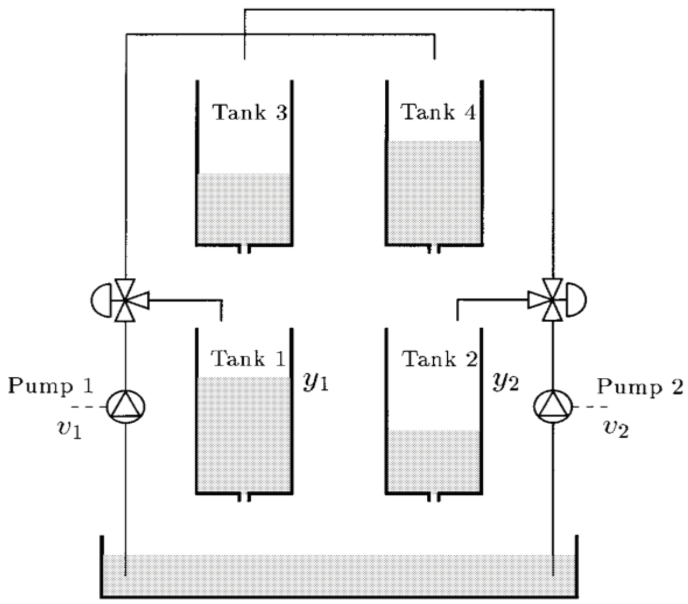

Fig. (4) Schematic diagram of the QTS [9].

Where, Table (1) presents the values and descriptions of parameters. Equation (18) converts to multi-inputs multi-outputs (MIMO) nonlinear state space representation with two inputs (pumps voltages) and two outputs (Tank1 and Tank2 levels), which is demonstrated in equation (19). In this paper, the system model of DHDP is represented by (19). The Runge-Kutta 4.5 method is used to solve the differential equation of QTS model. MATLAB V2018b is used to implement the entire structure of HDP.

Table (1) The parameters for differential equation QTS model

State variables	Value	Description
A_1	$28\ cm^2$	Cross-section of Tank1
A_2	$32\ cm^2$	Cross-section of Tank2
A_3	$28\ cm^2$	Cross-section of Tank3
A_4	$32\ cm^2$	Cross-section of Tank4
a_1	$0.071\ m^2$	Cross-section of outlet hole of Tank1
a_2	$0.057\ cm^2$	Cross-section of outlet hole of Tank2
a_3	$0.071\ cm^2$	Cross-section of outlet hole of Tank3
a_4	$0.057\ cm^2$	Cross-section of outlet hole of Tank4
h_i	------ cm	Liquid level of Tank *i*
γ_1	0.7	Constant of the three-way valve1
γ_2	0.6	Constant of the three-way valve2
v_1	------ V	Required voltage for pump1
v_2	------ V	Required voltage for pump2
k_1	$3.33\ \ cm^3/Vs$	Converter for input 1
k_2	$3.35\ \ cm^3/Vs$	Converter for input 2
g	$981\ cm/s^2$	The acceleration of gravity

$$\frac{dx}{dt}=\begin{bmatrix} \frac{-1}{T_1} & 0 & \frac{A_3}{A_1T_3} & 0 \\ 0 & \frac{-1}{T_2} & 0 & \frac{A_4}{A_2T_4} \\ 0 & 0 & \frac{-1}{T_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{T_4} \end{bmatrix}x+\begin{bmatrix} \frac{\gamma_1k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix}u,\ y=\begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \end{bmatrix}x, \qquad (19)$$

Where $x=[h_1 \quad h_2 \quad h_3 \quad h_4]^T$, $y=[y_1 \quad y_2]^T$, $u=[v_1 \quad v_2]^T$ and the time constant is defined as follows:

$$T_i=\frac{A_i}{a_i}\sqrt{\frac{2h_i}{g}}, i=1,2,3, and\ 4.$$

III. Plots and Discussion of Simulation Results

In this section, the comparison between the proportional integral derivative (PID) as in [10], and our approach (DHDP). These PID gains are given in the PID transfer function:

$$u_x(s) = K_p + K_i \frac{1}{s} + K_d \frac{N}{1+N\frac{1}{s}}, \tag{20}$$

Where K_p is proportional gain, K_i is integral gain, K_d is derivative gain, and N is the first-order derivative filter gain (for reducing noise and distortions). In this paper, we used two PID controllers (one for pump1 and the other for pump2). The values for these gains are taken from [10] with improvement by using try-and-error method, which are K_p = 3, K_d = 1.2, K_i = 0.1, and N = 108 for the PID of pump1, the PID gains for pump2 are K_p = 2.7, K_d = 1.2, K_i = 0.0675, and N = 100. The basic HDP parameters are described as follows: the discount rate is 0.95; critic learning rate is 0.05 and the actor learning rate is 0.01; the training for either network will be terminated if the error drops under 1e – 2 or if the number of iterations meets the stopping threshold. The number of neurons in the hidden layer is 24 for critic network and 20 for actor network. Figure (5) shows the states of level tank1 after using PID and HDP during 1000 sec. Clearly, the HDP approach has better performance comparing with PID with fast response and no overshoot. Moreover, the level state in tank 1 has better steady-state complaining with PID as shown in zoom-in of Figure (5). Similarly, Figure (6) shows the states of level tank2 after using PID and HDP during 1000 sec. whereas, the HDP approach has better performance comparing with PID with fast response and small value of overshoot. Figure (7) presents the summation of errors of two level states during time. Clearly, the HDP approaches have small error comparing with PID controller. Figure (8) shows the average of level errors over learning iterations (2000 times) with zoom-in for last iteration with 5 different runs. The controller of HDP (actor network) is taken for last iteration, which is semi-optimal controller, because of last error.

IV. Technical and Economic Feasibility

The mean-squared-error with the PID approach is 0.3849, while the mean-squared-error with the HDP approach is 0.0029. That means, the improvement percentage is 99.2466%, which yields a very efficiency of using electrical power. However, the HDP approach has better results and more reliable to use, but HDP requires building two neural networks and high-speed computer for training and leaning the critic and actor networks. Because of most equipment in our company has programmable logic control (PLC) devices, the neural network block is already existed in the toolbox of PLC programing. Therefore, this project can apply in real by installing PLC or (remote terminal unit – RTU) near to any equipment with HDP toolbox connected to the sensors and actuators of certain equipment. At first time, the HDP toolbox in PLC or RTU are learnt by itself to build suitable robust controller (actor network). Then, the HDP controller is used during normal situations, while if any hard sadden events happen to the equipment that change the internal model (the PID controller cannot handle it), the HDP toolbox starts learning from scratch again to overcome the new situations.

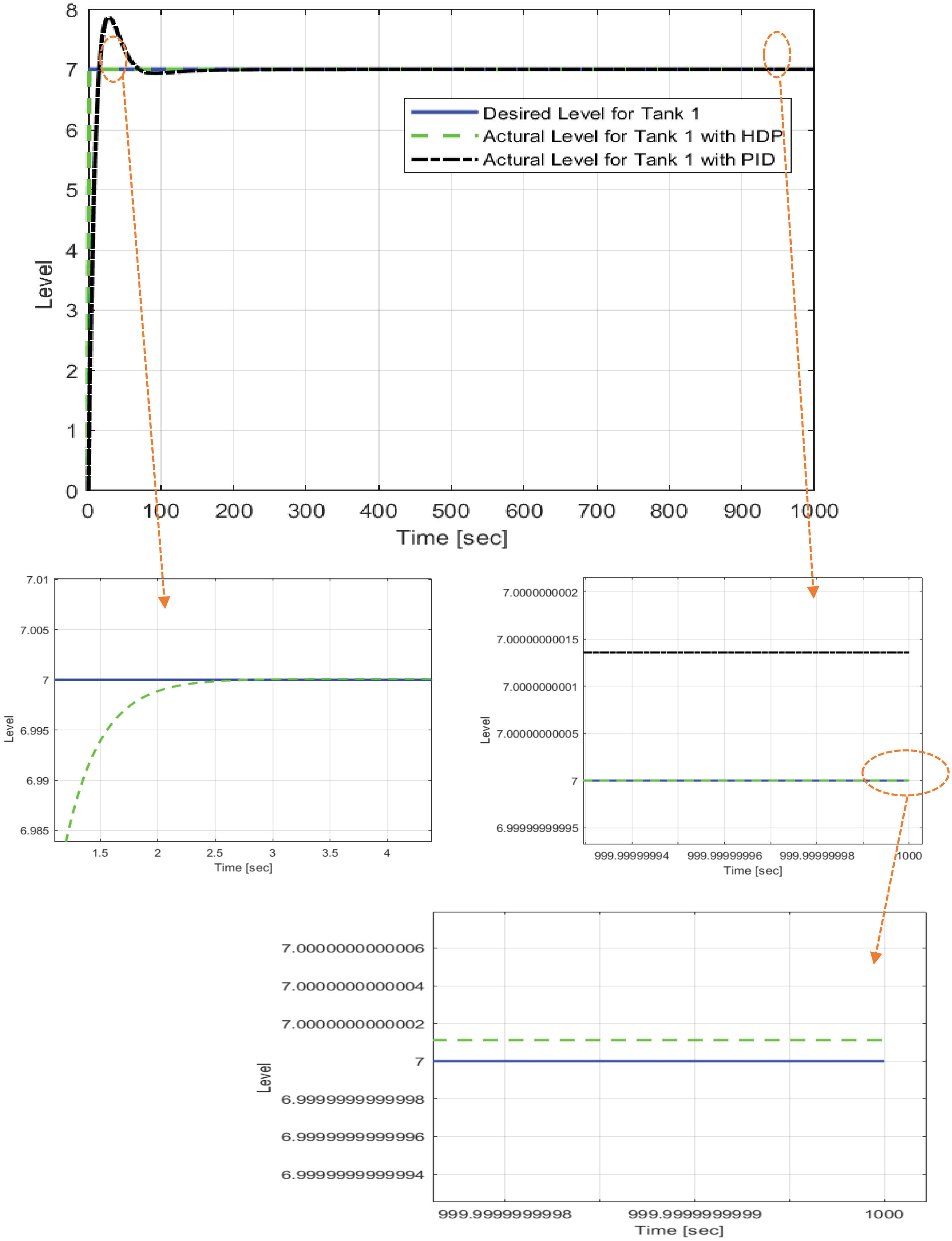

Fig. (5) The level of Tank 1 state coming from PID and HDP approaches with zoom-in.

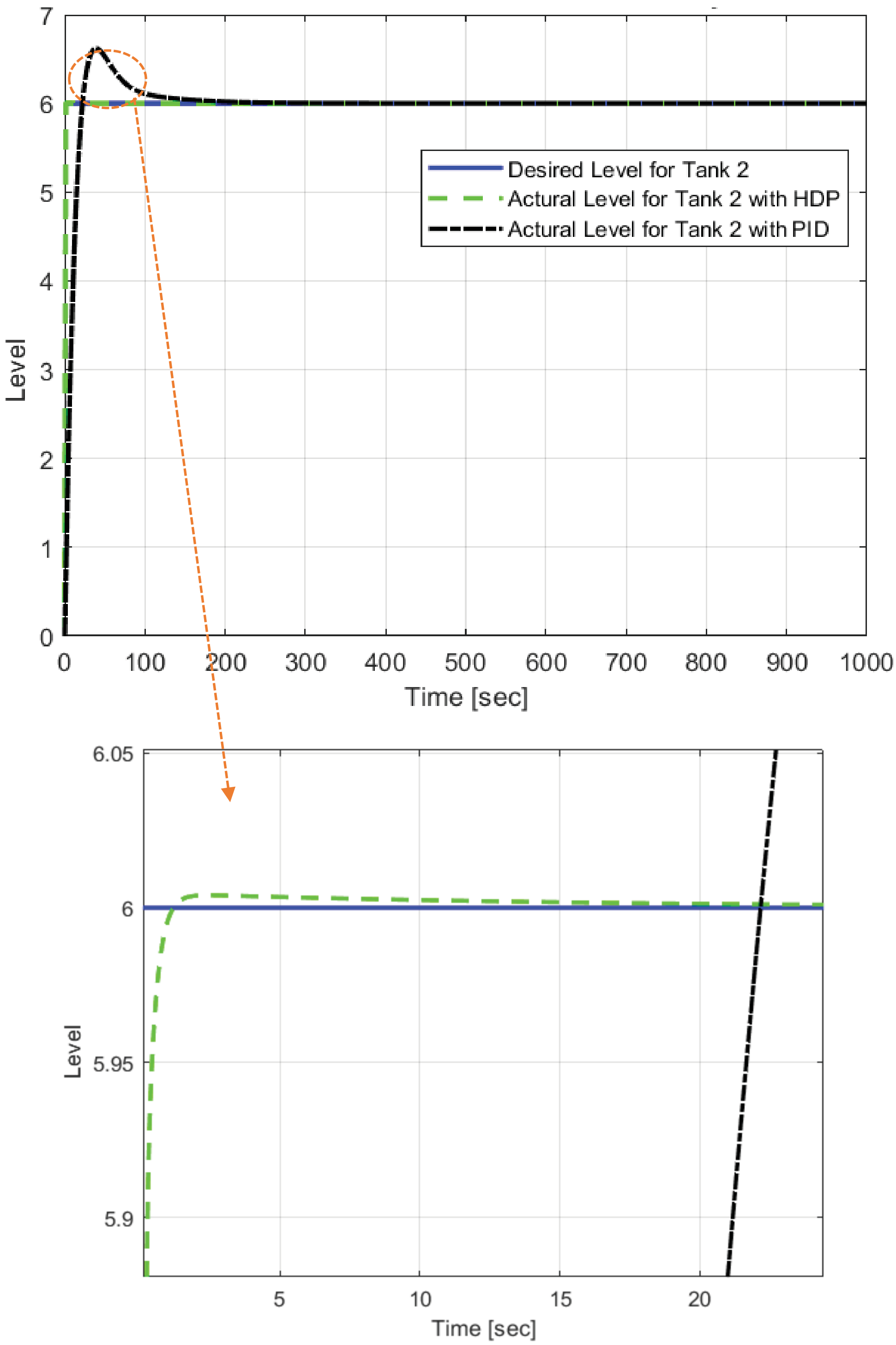

Fig. (6) The level of Tank 2 state coming from PID and HDP approaches with zoom-in.

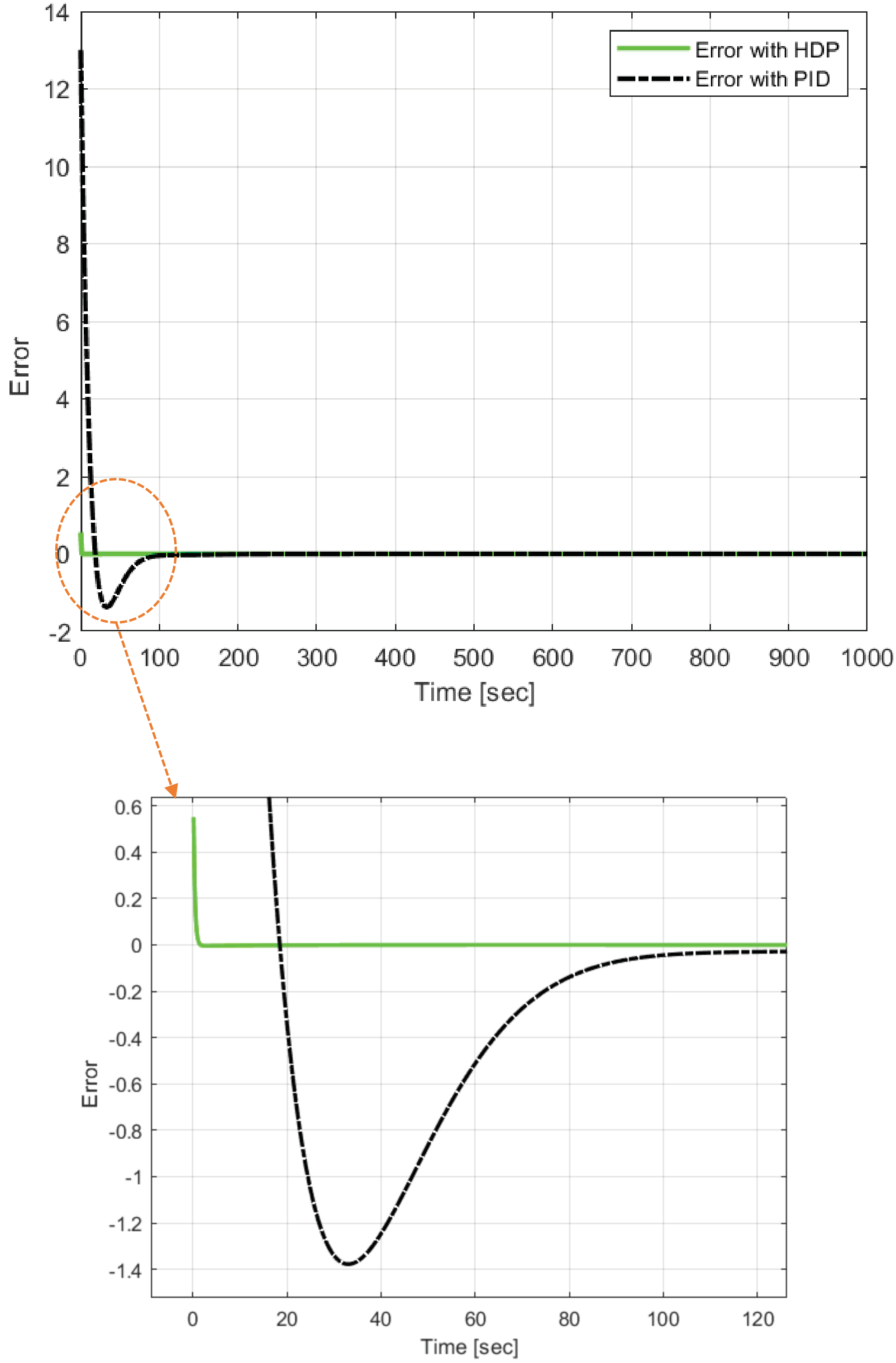

Fig. (7) The summation of errors for both level states of PID and HDP approaches.

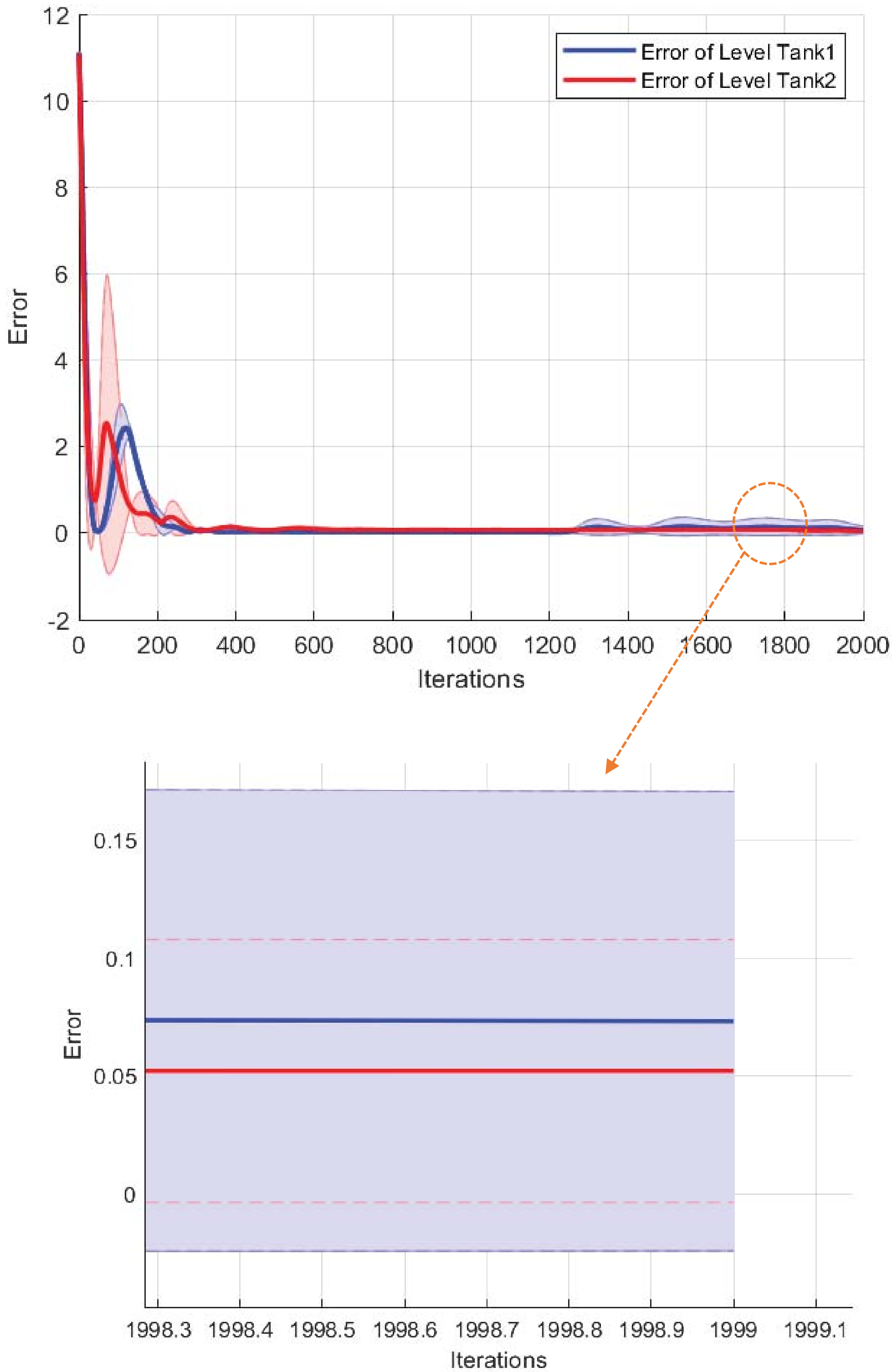

Fig. (8) The average summation error for both level states over iteration of HDP approaches. The solid lines is the mean of runs, while the shaded color is the standard deviation of the runs

V. Conclusion

This paper has presented DHDP for controlling on the well-known device using in the oil and gas industrial, which is QTS. The performance of HDP was excellent during time compared to PID controller. Merging neural network with oil and gas field presents improvement the generalization ability of the system with dealing with dynamic change in the environment. A significant advantage to boost the efficiency of control the level of tanks is demonstrated in this paper.

Nomenclature

HDP	Heuristic Dynamic Programming
QTS	Quadruple Tank System
MIMO	Multi-Inputs Multi-Outputs
PID	Proportional Integral Derivative
ADP	Approximate Dynamic Programming
DHP	Dual Heuristic Programming
AD	Action-Dependent
ADDHP	Action-Dependent Dual Heuristic Programming
ADHDP	Action-Dependent Heuristic Dynamic Programming
PLC	Programmable Logic Control
RTU	Remote Terminal Unit

References

1. X. Zhong, Z. Ni, and H. He, “A Theoretical Foundation of Goal Representation Heuristic Dynamic Programming,” *IEEE Trans. Neural Netw. Learn. Syst*., vol. 27, no. 12, pp. 2513 - 2525, Dec. 2016.
2. S. Al-Dabooni, and D. Wunsch, “The Online Model-Free N-Step HDP with Stability Analysis," published, accepted in May 25, 2019 for *IEEE Transaction on Neural Network and Learning Systems*.
3. G. K. Venayagamoorthy, R. G. Harley, and D. Wunsch, “Dual heuristic programming excitation neurocontrol for generators in a multimachine power system,” *IEEE Trans. Applications Industry*, vol. 39, no. 2, pp. 382-394, Mar. 2003.
4. N. Zhang, and D. Wunsch, “A Comparison of Dual Heuristic Programming (DHP) and neural network based stochastic optimization approach on collective robotic search problem,” *IEEE Trans. Neural Netw. and Learn Syst.*, vol. 1, pp. 248-253, Jul. 2003..
5. C. Lian, and X. Xu, “Motion planning of wheeled mobile robots based on heuristic dynamic programming,” *IEEE* Proc. *World Congress Intelligent Control and Automation (WCICA)*, pp 576-580, Jul. 2014.
6. S. Al-Dabooni, and D. Wunsch, “Heuristic dynamic programming for mobile robot path planning based on Dyna approach, “*IEEE/INNS, International Joint Conference on Neural Networks (IJCNN)*, pp. 3723-3730, Jul. 2016.
7. V. Mnih, A. P. Badia , M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” International Conference on Machine Learning, pp. 1928-1937, Jul. 2016.
8. O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, and J. Quan, “StarCraft II: A New Challenge for Reinforcement Learning,” arXiv preprint arXiv:1708.04782. Aug. 2017.
9. K. H. Johansson, “The quadruple-tank process: A multivariable laboratory process with an adjustable zero.” IEEE Transactions on control systems technology, vol. 8, no. 3, pp. 456 – 465, May 2000.
10. E. G. Kumara, B. Mithunchakravarthib and N. Dhivyac, “Enhancement of PID Controller Performance for a Quadruple Tank Process with Minimum and Non-

Minimum Phase Behaviors” ScienceDirect, 2nd International Conference on Innovations in Automation and Mechatronics Engineering, ICIAME 2014.