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 applied 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.

Metamodules training that in the field of oil and gas.

The abstract

In the latest era, the dual control is used in most industrial sectors in the fields of oil and gas. Therefore, the intelligent dual control is used in the dual control of the dual control model. This control is the control of the dual control model in the dual control system and is compared with the PID control via MATLAB programming. DHDP provides enhanced performance comparing with the PID approach with 99.2466% improvement.

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

Approximate dynamic programming (ADP) is a useful tool to overcome a behavior of nonlinear systems [1]. ADP has three categories [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 a turbo-generator. [4] shows the ability of DHP to solve a swarm robot problem. [5] and [6] illustrated that ADHDP can obtain an optimal path by multi-robot navigation. Recently, [7] and [8] are used with Atari game to solve many hard problems with a 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 a 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).
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].

\[
J^*(s, u) = P_{ss'}^u (r_t + \gamma \max_{u \in \mathcal{A}} J^*(s', u))
\]  

(1)

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

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

(2)

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

\[
u^*(s) = \arg \max_{u \in \mathcal{A}} [r(s, u) + \gamma J^*(s', u)].
\]  

(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:

\[
\delta_t = J_{t-1} - (r_t + \gamma J_t).
\]  

(4)
And
\[ E_t^c = \frac{1}{2} \delta_t^2. \]  

(5)

The gradient-based adaptation for the weights update rule in the critic network can be given by
\[ w_{t+1}^c = w_t^c + \Delta w_t^c, \]  

(6)

\[ \Delta w_t^c = \ell_t^c \left[ -\frac{\partial E_t^c}{\partial w_t^c} \right], \]  

(7)

\[ \frac{\partial E_t^c}{\partial w_t^c} = \left[ \frac{\partial E_t^c}{\partial \ell_t^c} \frac{\partial \ell_t^c}{\partial w_t^c} \right], \]  

(8)

Where, \( \ell_t^c \) 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 (\( \Delta w_t^{c2} \)) according to backpropagation rules are:
\[ \Delta w_t^{c2} = -\ell_t^c y \delta_t p^T, \]  

(9)

While, the weights updating from input to hidden layer (\( \Delta w_t^{c1} \)) are:
\[ \Delta w_t^{c1} = -\ell_t^c 0.5 \left[ Id(n_c) - \text{diag}(p_j^2) \right][y \delta_t w_t^{c2}]^T [In] \]  

(10)

where \( n_c \) is the total number of hidden nodes in the critic network; \( p_j = \sigma(q_j) \) is the j output of the hidden nodes \( q, p \in \mathbb{R}^{n_c} \); \( \sigma(.) \) 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; \( In \in \mathbb{R}^{(n+m)} \); \( Id(.) \) is the identity matrix, \( \text{diag}(.) \) 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

\[ \mu_t = J_t - U_c. \]  

(11)

Therefore, the objective function in the action network is

\[ E_t^a = \frac{1}{2} \mu_t^2. \]  

(12)

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

\[ w_{t+1}^a = w_t^a + \Delta w_t^a, \]  

(13)

\[ \Delta w_t^a = \ell_t^a \left[ -\frac{\partial E_t^a}{\partial w_t^a} \right]. \]  

(14)

\[ \frac{\partial E_t^a}{\partial w_t^a} = \left[ \frac{\partial E_t^a}{\partial J_t} \frac{\partial J_t}{\partial u_t} \frac{\partial u_t}{\partial w_t^a} \right]. \]  

(15)

Where \( \ell_t^a \) 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 \((\Delta w_t^{a2})\) according to backpropagation rules are:

\[ \Delta w_t^{a2} = -\ell_t^{a2} \mu_t [w_t^{c2}] \left( 0.5 \left[ Id(n_c) - diag(p_f^2) \right] \right)[w_{ca}] \left( 0.5 \left[ Id(m) - diag(u_t^2) \right] \right) g^T. \]  

(16)

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

\[ \Delta w_t^{a1} = -\ell_t^{a1} \mu_t [w_t^{c1}] \left( 0.5 \left[ Id(n_c) \right] \right)[w_{ca}] \left( 0.5 \left[ Id(m) \right] \right) [w_t^{a2}] \times \]  

\[ \left( 0.5 \left[ Id(n_a) - diag(g_f^2) \right] \right)^T s_t, \]  

(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, \( w_{ca} \in \mathbb{R}^{n_a \times (n+1:n+m)} \) from \( w_t^{c1} \); \( g_f \) is the jth output of the hidden nodes of the action network, \( g \in \mathbb{R}^{n_a \times 1} \). Both critic and action learning rate decrease with time until a certain small value as we present in the result section.
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 ($u_1$ and $u_2$). 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]:

\[
\begin{align*}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1}{A_1}k_1u_1, \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2}{A_2}k_2u_2, \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2, \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1.
\end{align*}
\]
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

<table>
<thead>
<tr>
<th>State variables</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>28 cm$^2$</td>
<td>Cross-section of Tank1</td>
</tr>
<tr>
<td>$A_2$</td>
<td>32 cm$^2$</td>
<td>Cross-section of Tank2</td>
</tr>
<tr>
<td>$A_3$</td>
<td>28 cm$^2$</td>
<td>Cross-section of Tank3</td>
</tr>
<tr>
<td>$A_4$</td>
<td>32 cm$^2$</td>
<td>Cross-section of Tank4</td>
</tr>
<tr>
<td>$a_1$</td>
<td>0.071 m$^2$</td>
<td>Cross-section of outlet hole of Tank1</td>
</tr>
<tr>
<td>$a_2$</td>
<td>0.057 cm$^2$</td>
<td>Cross-section of outlet hole of Tank2</td>
</tr>
<tr>
<td>$a_3$</td>
<td>0.071 cm$^2$</td>
<td>Cross-section of outlet hole of Tank3</td>
</tr>
<tr>
<td>$a_4$</td>
<td>0.057 cm$^2$</td>
<td>Cross-section of outlet hole of Tank4</td>
</tr>
<tr>
<td>$h_i$</td>
<td>----- cm</td>
<td>Liquid level of Tank $i$</td>
</tr>
<tr>
<td>$y_1$</td>
<td>0.7</td>
<td>Constant of the three-way valve1</td>
</tr>
<tr>
<td>$y_2$</td>
<td>0.6</td>
<td>Constant of the three-way valve2</td>
</tr>
<tr>
<td>$v_1$</td>
<td>----- V</td>
<td>Required voltage for pump1</td>
</tr>
<tr>
<td>$v_2$</td>
<td>----- V</td>
<td>Required voltage for pump2</td>
</tr>
<tr>
<td>$k_1$</td>
<td>3.33 cm$^3$/Vs</td>
<td>Converter for input 1</td>
</tr>
<tr>
<td>$k_2$</td>
<td>3.35 cm$^3$/Vs</td>
<td>Converter for input 2</td>
</tr>
<tr>
<td>$g$</td>
<td>981 cm/s$^2$</td>
<td>The acceleration of gravity</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\frac{dx}{dt} &= \begin{bmatrix}
\frac{-1}{T_1} & 0 & \frac{A_3}{A_1T_3} & 0 \\
0 & \frac{-1}{T_2} & 0 & \frac{A_4}{A_2T_4} \\
0 & 0 & \frac{-1}{T_3} & 0 \\
0 & 0 & 0 & \frac{-1}{T_4}
\end{bmatrix} x + \begin{bmatrix}
\frac{y_1k_1}{A_1} & 0 \\
0 & \frac{y_2k_2}{A_2} \\
0 & \frac{(1-y_1)k_1}{A_3} \\
0 & \frac{(1-y_2)k_2}{A_4}
\end{bmatrix} u, \quad y = \begin{bmatrix} 0.5 & 0 & 0 & 0 \end{bmatrix} x, \\
& (19)
\end{align*}
\]

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:

\[
T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_i}{g}}, \quad i = 1, 2, 3, \text{ and } 4.
\]

E26
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:

\[ u_x(s) = K_p + K_i \frac{1}{s} + K_d \frac{N}{1 + N^2 s^2}, \]  

(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

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDP</td>
<td>Heuristic Dynamic Programming</td>
</tr>
<tr>
<td>QTS</td>
<td>Quadruple Tank System</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multi-Inputs Multi-Outputs</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>ADP</td>
<td>Approximate Dynamic Programming</td>
</tr>
<tr>
<td>DHP</td>
<td>Dual Heuristic Programming</td>
</tr>
<tr>
<td>AD</td>
<td>Action-Dependent</td>
</tr>
<tr>
<td>ADDHP</td>
<td>Action-Dependent Dual Heuristic Programming</td>
</tr>
<tr>
<td>ADHDP</td>
<td>Action-Dependent Heuristic Dynamic Programming</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Control</td>
</tr>
<tr>
<td>RTU</td>
<td>Remote Terminal Unit</td>
</tr>
</tbody>
</table>
References


10. E. G. Kumara, B. Mithunchakravarthib and N. Dhivyac, “Enhancement of PID Controller Performance for a Quadruple Tank Process with Minimum and Non-