IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 November 2023, accepted 9 December 2023, date of publication 13 December 2023,
date of current version 20 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3342201

RESEARCH ARTICLE

Output Feedback Control for Deterministic Unknown Dynamics Discrete-Time System Using Deep Recurrent Q-Networks

ADI NOVITARINI PUTRI[1], **EGI HIDAYAT**[1], **(Member, IEEE),**
DIMITRI MAHAYANA[1], **(Member, IEEE), AND CARMADI MACHBUB**[1,2], **(Member, IEEE)**
[1]Control and Computer System Research Group, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia
[2]Institut Teknologi Sains Bandung, Bekasi 17530, Indonesia

Corresponding author: Dimitri Mahayana (dimitri@itb.ac.id)

This work was supported in part by the School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB).

ABSTRACT The current application of control theory is commonly carried out in systems with a model or known system dynamics. However, in practice this is a formidable task to achieve as not all state information can be known. The use of the Output Feedback (OPFB) scheme in the field of control systems also possesses a weakness because it requires the use of an observer. This appears rather contradictory as the use of an observer requires system dynamics information. This research proposes an optimal control scheme using Deep Recurrent Q-Networks (DRQN) to generate an optimal control signal trajectory based on a collection of input and output data from the system itself. The approach proposed in this study is based on the Q-Learning method from the Reinforcement Learning (RL) scheme. The Long-Short Term Memory (LSTM) is used to approximate the Q-function and determine the control signals for a system without a known model. The method that we proposed in this study has been tested on four case studies. The control signal trajectory generated from our proposed algorithm, is much smaller than the control signal that generated from classical Q-Learning scheme. The results of this research are certainly relevant to the aim of OPFB, namely that the controller is designed to be able to regulate (bring the state trajectory to zero) and minimize control signal energy. It is empirically discovered that the same result is proven by the norm values resulting from the Q-function trajectory. The norm of Q-function trajectory for our proposed algorithm on the 1st, 2nd, 3rd, and 4th case studies are 2.11E-08, 3.15E-06, 3.79E-09, and 1.59E-13, respectively.

INDEX TERMS Reinforcement learning, output feedback, q-learning, deep recurrent q-network, LSTM.

I. INTRODUCTION

The use of Reinforcement Learning (RL) algorithm in designing the optimal control has progressed recently [1], [2], [3]. One of the problems in optimal control is Output Feedback (OPFB) [4], [5]. This scheme allows control design without going through full state feedback. The control objectives using the OPFB scheme are (1) to fulfill the stability conditions of the closed loop system, (2) the control system is able to track the desired reference signal [2], [5].

The application of the OPFB scheme has been carried out in several studies. In the oil and gas industry, rotary drilling equipment is needed to open a borehole in a rock formation. However, this equipment has problems regarding vibration which has implications for reducing oil exploration efficiency. In research, [6] has succeeded in implementing vibrations control based on the OPFB scheme to maintain oil exploration efficiency. The authors proposed the use of OPFB due to the limited number of sensors on the equipment. The use of the OPFB method has also been implemented in Unmanned Aerial Vehicle (UAV) systems to track reference signals [7].

Solving OPFB can be executed by seeking a solution to the Hamilton-Jacobi-Bellman Equation (HJB) analytically [2], [8]. Solving the HJB equation requires a system dynamics model, which is quite complex to obtain practically. In addition, the OPFB scheme requires an observer to produce state trajectories during the learning process [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

Conversely, RL methods can be classified as model-based and model-free [9], [10]. The model-based RL method performs the role of the Dynamic Programming (DP) to find the optimal control signal. Meanwhile, the model-free RL method employs Q-learning in general. On-policy and off-policy are model-based RL method classifications viewed from a policy perspective [11].

In control system design, acquiring a mathematical model of the control presents considerable difficulty. This is due to the uncertainty factor between the model and the real system. The application of a model-free scheme, when viewed from the perspective of control system theory, has relevance to the OPFB scheme. However, using a conventional OPFB scheme calls for the role of an observer which requires information on the system dynamics model. The development of artificial intelligence approaches (e.g. RL) is also applied in designing optimal control system schemes without the need to know the system dynamics model. From the point of view of the RL method, this scheme is known as model-free. This study seeks to show that the use of a combination of artificial intelligence methods (i.e RL and ANN) can be employed to solve optimal control system problems.

The application of OPFB that requires the role of observer is contradictory to the model-free RL terminology. The use of the model-free RL method (i.e. Q-Learning) has been successfully applied to solve Linear Quadratic Tracking (LQT) problems on discrete-time (DT) systems [12], [13]. In [12], research has succeeded in utilizing neural networks to approximate value functions based on the value iteration method. However, in this research, the model-based RL method was still used. However, in [13] the calculation of the Q-function is still carried out on the basis of the model.

The implementation of a model-free RL method for a control system scheme requires significant efforts. As a result of designing a control system without a model, an estimator (for example: a Kalman filter) is required to estimate the dynamics of the state needed to design a controller. The use of the Kalman filter as an estimator is completely contradictory to the terminology of the model-free RL [14], [15]. Further research was conducted at [16] which developed a model-free RL method to solve LQT problems based on system input and output data. In [13] and [16] continue to use the state trajectory to operate the LQT scheme. The operation of the OPFB scheme on [17] has successfully implemented the Q-Learning method based on a collection of system input and output data to find the optimal solution. However, in the research [3], [17] do not implement the discount factor parameter in the RL method but the stability of the system can be maintained. In addition, [3], [17] research continues to use state trajectories in the design of optimal controllers. Model-free RL method has been employed to solve the optimal control problem in [18]. The authors' choice to employ the $\lambda-$PI method reveals a weakness since it is based on the Policy Iteration (PI) method which requires a controller initialization value that is guaranteed to stabilize the system. This proves to become a bottleneck in the process of calculating and implementing the control system.

When viewed from the perspective of computer science, the Q-Learning method can also be developed by involving deep learning into Deep Q Networks (DQN) [19]. The DQN was first applied in computer vision to provide an action based on the case study of the Atari problem [19]. Atari is originally a home video game console developed in 1977 and sold for over a decade [20]. Atari is a single game screen of 160 pixels wide and 210 pixels high, with a 128-colour palette, 18 actions can be input to the game via a digital joystick. The digital joystick is represent the three positions of the joystick for each axis, plus a single button. The Atari 2600 hardware limits the possible complexity of games, which is believed to strike the perfect balance: a challenging platform offering conceivable near-term advancements in learning, modeling, and planning [20]. The use of the Deep Recurrent Q-Networks (DRQN) method is preceded by [21], this is because the type of environment required is partially observer.

This paper is a further research from [22]. In [22], we proposed the combination of model-based RL and KalmanNet to adapt the conventional Linear Quadratic Gaussian (LQG) scheme. The proposed algorithm in [22] is formulated to solve the regulator problem for stochastic and DT linear systems. Contributions to this research include:

- Proposing a new framework, DRQN based on LSTM Networks to solve the OPFB problem for deterministic discrete-time systems. We employ the DRQN to discover the optimal controller gain based on measured data of the system.
- Implementing the LSTM network to adapt the OPFB based Q-Learning algorithm to carry out the policy evaluation and policy improvement stage to obtain the control signal trajectories. LSTM is designed to handle sequential data, making them well-suited for tasks where the input or output data has a temporal or sequential structure [23].
- The advantage of the control scheme proposed in this study, when compared to [22], is that it no longer requires information regarding the dynamical model of the plant (A, B, C, D) in designing the optimal control. Additionally, it is no longer necessary to assign an observer as a state estimation method.

The remaining section of this paper covers the development of the proposed algorithm. Section II comprises the definition of the problem of conventional OPFB and OPFB based on classical Q-Learning. Section III discusses the DRQN based on LSTM as our proposed solution. The application of the proposed solution to the design data driven OPFB scheme for cart-pole system, batch distillation column, and an unstable plant are included in Section IV, which also covers the simulations and evaluations of several test schemes. This arrangement is aimed at ensuring that our proposed algorithm provides the most optimal results empirically. Lastly, Section V contains the conclusions of this research.

A. NOTATIONS

In this research, k, n, m, p, and N are represent the time-step, state-order, input-order, output-order, and time-horizon, respectively. The control gains obtained from the conventional OPFB, classical Q-Learning, and DRQN based on LSTM are denoted as $K^{\mathcal{M}}$, $K^{\mathcal{L}}$, and $K^{\mathcal{D}}$, respectively. The control signal in general form is denoted as u_k. The Riccati solutions achieved from conventional OPFB and DRQN based on LSTM are denoted as P and $P^{\mathcal{D}}$, respectively. The discount factor is denoted as γ.

II. OPFB BASED ON CLASSICAL Q-LEARNING

In control system design, developing a mathematical model of the control is rather complex. This is due to the uncertainty factor between the model and the real system. The application of a model-free scheme when observed from the point of view of control system theory has relevance to the OPFB scheme. However, using a conventional OPFB scheme calls for the role of an observer requiring some information on the system dynamics model. The development of RL is also utilized to design optimal control system schemes without the system dynamics model. Q-Learning is one of the model-free RL methods devised to solve the optimal control problem. In Section II-B, a brief introduction of the Q-learning method based on measured data is presented.

A. CONVENTIONAL OPFB

The dynamics of a DT linear system is formulated in Eq. (1) and (2) where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$ and A, B, and C are the state, input, and output matrices, respectively.

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

$$y_k = Cx_k \tag{2}$$

The control signal could be obtained by state feedback control laws of the form in Eq. (3).

$$u_k = K^{\mathcal{M}}\hat{x}_k \tag{3}$$

Meanwhile, observer calculations can be carried out as in Eq. (4).

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k) \tag{4}$$

From Eq. (3) and (4), the $K^{\mathcal{M}}$ and L are the controller and observer gain, respectively.

The observer gain L for deterministic case must be selected so that the observer poles $(A - LC)$ can be arbitrarily located in a unity circle for the DT system [24]. In this research, a Luenberger observer is employed to obtain the observer gain for the model-based approach. The Luenberger observer is an applicable method to locate the observer gain in the sense of a deterministic case [24].

The selection of the controller gain matrices is uncovered by solving ARE formulated in Eq. (5) where P is the Riccati solution [3].

$$K^{\mathcal{M}} = (R_u + B^T PB)^{-1} B^T PA \tag{5}$$

The OPFB scheme can be used to solve this problem by involving the role of observer [24], [25]. The OPFB scheme has a weakness, namely the dynamics model must be known. Of course this contradicts the terminology of the model-free RL method.

B. OPFB VIA Q-LEARNING BASED ON INPUT-OUTPUT DATA

Consider a DT linear system could be formulated in Eq. (1) and (2). Beneath the controllability and observability assumption on (A, B) and (A, C), respectively, we would like to discover the control signal that minimizes the cost function. The reward function for quadratic cases can be formulated in Eq. (6) where $R_y \geq 0$ and $R_u > 0$ are output and control weight matrices, respectively. This reward function is inspired from [3] and [17] but modified using the output signal. It is assumed that the state information is immeasurable. Thus, the output signal for reward computation is utilized [15].

$$r(y_k, u_k) = y_k^T R_y y_k + u_k^T R_u u_k \tag{6}$$

On the basis of the reward function in Eq. (6), the Q-function is defined in Eq. (7).

$$Q(y_k, u_k) = r(y_k, u_k) + \gamma Q(y_{k+1}, u_{k+1}) \tag{7}$$

1) THE AUGMENTED SYSTEM BASED ON MEASURED DATA

The control signal based on Q-Learning that we used in this section is denoted as u_k. The dynamics of the system in the finite time horizon $[k - N, k]$ can be expressed as in Eq. (8a) [26], [27] and can be simplified to Eq. (8c) with the U_N is defined in Eq. (8b) and $\bar{u}_{k-1,k-N} = \begin{bmatrix} u_{k-1} & u_{k-2} & u_{k-3} & \dots & u_{k-N} \end{bmatrix}^T$ [1], [16]. The U_N is called as controlability matrices [3], [16].

$$x_k = A^N x_{k-N} + \begin{bmatrix} B & AB & \dots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-N} \end{bmatrix} \tag{8a}$$

$$U_N = \begin{bmatrix} B & AB & A^2B & \dots & A^{N-1}B \end{bmatrix}^T \tag{8b}$$

$$x_k = A^N x_{k-N} + U_N \bar{u}_{k-1,k-N} \tag{8c}$$

The system output equation can be formulated into Eq. (9a) by deriving a formula like Eq. (9b)-(9d).

$$\bar{y}_{k-1,k-N} = V_N x_{k-N} + T_N \bar{u}_{k-1,k-N} \tag{9a}$$

$$\bar{y}_{k-1,k-N} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-N-1} \\ y_{k-N} \end{bmatrix} \tag{9b}$$

$$V_N = \begin{bmatrix} CA^{N-1} \\ CA^{N-2} \\ \vdots \\ CA \\ C \end{bmatrix} \tag{9c}$$

$$T_N = \begin{bmatrix} 0 & CB & CAB & \dots & CA^{N-2}B \\ 0 & 0 & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9d}$$

The Eq. (8c) can be simplified into Eq. (10a) where matrices M and $\bar{z}_k$ is defined in Eq. (10b) and (10c), respectively [1], [17].

$$x_k = M\bar{z}_k \tag{10a}$$

$$M = \begin{bmatrix} U_N - A^N V_N^+ T_N & A^N V_N^+ \end{bmatrix} \tag{10b}$$

$$\bar{z}_k = \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \end{bmatrix} \tag{10c}$$

Assumption 1: The pair of (A, B) is controllable and (A, C) is observable.

If Assumption 1 holds, then V_N is full rank, then the pseudo-inverse of the V_N observability matrices can be expressed as in Eq. (11)

$$V_N^+ = (V_N^T V_N)^{-1} V_N^T \tag{11}$$

The state trajectory x_{k-N} can be formulated as Eq. (12) with V_N^+ as in Eq. (11) is assumed to be full column rank [1].

$$x_{k-N} = V_N^+(\bar{y}_{k-1,k-N} - T_N \bar{u}_{k-1,k-N}) \tag{12}$$

Substitute Eq. (12) to (8c) becomes Eq. (13).

$$x_k = (A^N V_N^+)\bar{y}_{k-1,k-N} + (U_N - A^N V_N^+ T_N)\bar{u}_{k-1,k-N} \tag{13}$$

So that the state equation can be stated as in Eq. (14) where the matrices M_y and M_u respectively represent Eq. (16) and (15).

$$x_k = M_y \bar{y}_{k-1,k-N} + M_u \bar{u}_{k-1,k-N} \tag{14}$$

$$M_y = A^N V_N^+ \tag{15}$$

$$M_u = (U_N - A^N V_N^+ T_N) \tag{16}$$

The Bellman equation could be formulated in Eq. (17) where $\gamma \in \begin{bmatrix} 0, 1 \end{bmatrix}$ is the discount factor.

$$V(x_k) = y_k^T R_y y_k + (u_k)^T R_u u_k + \gamma V(x_{k+1}) \tag{17}$$

For the quadratic case, the value function of an augmented states based on the shape of the performance index could be formulated in Eq. (18). The quadratic form for the value function is a common approximation used in optimal control and RL because it simplifies the problem and leads to tractable solutions.

$$V(x_k) = x_k^T \bar{P} x_k \tag{18}$$

Substitute Eq. (10a) to (18) such that the value function could be represent in Eq. (19).

$$V(x_k) = (M\bar{z}_k)^T \bar{P}(M\bar{z}_k) = \bar{z}_k^T P \bar{z}_k \tag{19}$$

Eq. (19) with $P = M^T \bar{P} M$. However, one thing that is quite important is that the matrices P depends on the dynamics of the system (A, B, C). We could know that P found by solving the Least-Square (LS) method, is equal to $M^T \bar{P} M$ where the proof was developed in [16].

2) BELLMAN EQUATION BASED ON MEASURED DATA

The Bellman equation for this OPFB scheme could formulated in Eq. (20) by subtitute the Eq. (19) into (17).

$$\bar{z}_k^T P \bar{z}_k = y_k^T R_y y_k + (u_k)^T R_u u_k + \gamma \bar{z}_{k+1}^T P \bar{z}_{k+1} \tag{20}$$

The Hamiltonian function could be formulated in Eq. (21)

$$H(\bar{z}_k, u_k) = y_k^T R_y y_k + (u_k)^T R_u u_k + \gamma \bar{z}_{k+1}^T P \bar{z}_{k+1} - \bar{z}_k^T P \bar{z}_k \tag{21}$$

The notation $\bar{z}_{k+1}$ can be defined as in Eq. (22a) and P can be formulated in Eq. (22b).

$$\bar{z}_{k+1} = \begin{bmatrix} u_k \\ \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \end{bmatrix} \tag{22a}$$

$$P = \begin{bmatrix} p_0 & p_u & p_y \\ (p_u)^T & p_{22} & p_{23} \\ (p_y)^T & p_{32} & p_{33} \end{bmatrix} \tag{22b}$$

Based on the Eq. (19), it can be seen that the P is related to M. Matrices M is also related to model-based state equations. Minimizing the Hamiltonian function in Eq. (21) with respect to u_k. Thus, the control law could be formulated in Eq. (23c) [1], [17].

$$\frac{\partial H(\bar{z}_k, u_k)}{\partial u_k} = R_u u_k + \gamma(p_0)u_k + \gamma((p_u)\bar{u}_{k,k-N+1} + (p_y)\bar{y}_{k,k-N+1}) = 0 \tag{23a}$$

$$u_k(R_u + \gamma p_0) + \gamma((p_u)\bar{u}_{k,k-N+1} + (p_y)\bar{y}_{k,k-N+1}) = 0 \tag{23b}$$

$$u_k = -\gamma(R_u + \gamma p_0)^{-1}((p_u)\bar{u}_{k,k-N+1} + (p_y)\bar{y}_{k,k-N+1}) \tag{23c}$$

Based on Eq. (23c) it implies that the control signal u_k depends on the value of the previous control signal, output and reference. Meanwhile, the control gain $K^{\mathcal{L}}$ is represented in Eq. (24).

$$u_k = \underbrace{-\gamma(R_u + \gamma p_0)^{-1} \begin{bmatrix} p_u & p_y \end{bmatrix}}_{K^{\mathcal{L}}} \underbrace{\begin{bmatrix} \bar{u}_{k,k-N+1} \\ \bar{y}_{k,k-N+1} \end{bmatrix}}_{\bar{z}_k} \tag{24}$$

In the next sub-section, we will discuss the use of DRQN algorithm to calculate the value of the controller gain to calculate the control strategy. The use of the Q-Learning method can be used to solve regulatory and tracking issues to

OPFB. The following is an algorithm for using the Q-learning method to overcome the OPFB problems.

III. DEEP OUTPUT FEEDBACK CONTROL

The novelty proposed appears as in Fig. 1. The system to be controlled has training data consisting of input and output signal denoted as u_k and y_k. The training dataset in a sample range from $k = 1$ to N are stored a $\bar{u}_{k-1,k-N}$ for the input signal and $\bar{y}_{k-1,k-N}$ for output signal. The dataset is stored as a $\mathbb{D}$ which then is applied to carry out the training stage. The sequences of $\bar{z}_k$ obtained from the $\mathbb{D}$ is the time series data type as input signal for the LSTM network. The LSTM networks have been appeared to memorize long-term conditions more easily than the basic RNN structures, to begin with on manufactured datasets designed for testing the capacity to memorize long-term conditions [28]. The Q-Learning method in this study is a baseline with the aim that the target data used is $r(y_k, u_k)$ as formulated in Eq. (6).

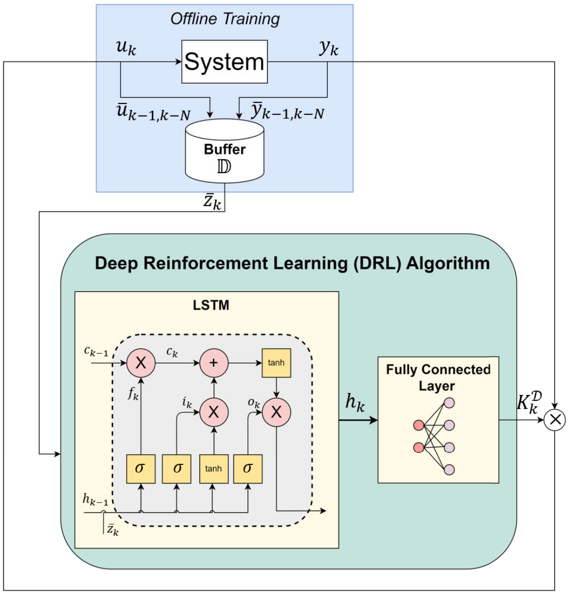


FIGURE 1. Proposed algorithm.

The implementation of DRQN based on LSTM scheme is summarized in Fig. 2. The testing for each case study is divided into three main stages. The 1st stage is responsible for creating a dataset via the *GenerateSeq()* function. As a test, the dataset in this study is given through a simulation for four case studie based on their mathematical model. Thus, the system matrices (A, B, C) are known. The *InputOutputSeq()* function is applied to generate the controllability matrix U_N, observability matrices V_N, and the Toeplitz matrix T_N which formulated in Eq. (8b), (9c), and (9d), respectively. Additionally to that, this function is also necessary to produce shifting matrices $\bar{z}_k$ like Eq. (22a).

The 2nd stage covers the implementation of the framework proposed in this study. At this stage, the calculation of the control signal trajectory u_k is rooted in the input and output signal dataset of the system, denoted as $\bar{u}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$, respectively. The use of the LSTM network to adapt the role of Q-learning in calculating control signals is summarized in Algorithm 1. At the 2nd stage, it is imperative to use LSTM to implement Algorithm 1. The LSTM parameters used in this study are selected with the final optimal hyperparameters, specifically:

- Backpropagation method: ADAM
- Number of hidden units: 30
- Maximum Epoch items: 1000
- Mini batch size: 128
- Gate activation function: sigmoid

In this research, we use the LSTM stability proposition, which is defined and proven in [29].

At the 3rd stage, the control signals obtained through DRQN based on LSTM are implemented into the system. All initial conditions are expressed as zero in the value. The tests at this stage is carried out to compare the three methods of the control, output, and Q-function trajectory that are obtained from the model-based Linear Quadratic Regulator (LQR), Q-Learning, and DRQN based on LSTM.

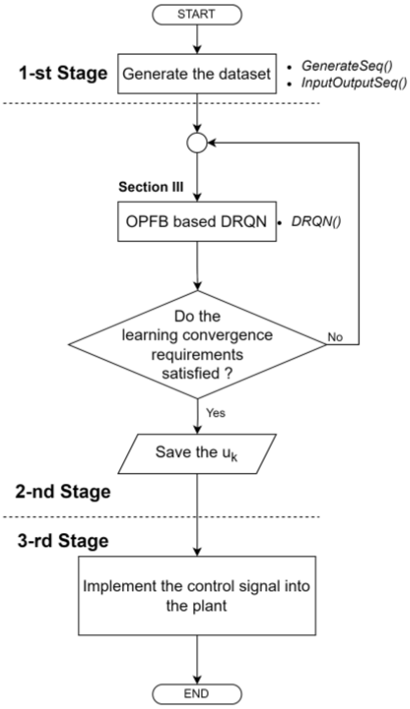


FIGURE 2. Flowchart implementation.

A. APPROXIMATE Q-LEARNING

The Q-function for OPFB problem was formulated in Eq. (7) where k is denoted for the time-step. In this research, we does not employing the discount factor because it does not incur bias from the excitation noise [17]. If the control signal u_k is obtained from the use of approximation scheme, then the quadratic Q-function can be expressed as in Eq. (25a) where $P^{\mathcal{D}}$ is the Riccati solution in the sense of approximation scheme. The $P^{\mathcal{D}}$ could be represent into matrix form in Eq. (25b).

$$Q(y_k,u_k)=\frac{1}{2}\begin{bmatrix}u_k\\ y_k\end{bmatrix}^T(P^{\mathcal{D}})\begin{bmatrix}u_k\\ y_k\end{bmatrix} \tag{25a}$$

$$Q(y_k,u_k)=\frac{1}{2}\begin{bmatrix}u_k\\ y_k\end{bmatrix}^T\begin{bmatrix}P^{\mathcal{D}}_{uu} & P^{\mathcal{D}}_{uy}\\ P^{\mathcal{D}}_{yu} & P^{\mathcal{D}}_{yy}\end{bmatrix}\begin{bmatrix}u_k\\ y_k\end{bmatrix} \tag{25b}$$

The process of calculating the control signal u_k based on the minimization concept is obtained through Eq. (26a)-(26c).

$$\frac{\partial}{\partial u_k}Q(y_k,u_k)=0 \tag{26a}$$

$$2P^{\mathcal{D}}_{uu}u_k+2P^{\mathcal{D}}_{uy}y_k=0 \tag{26b}$$

$$u_k=-(P^{\mathcal{D}}_{uu})^{-1}P^{\mathcal{D}}_{uy}y_k \tag{26c}$$

Meanwhile, the Q-function in Eq. (25b) is equivalent with Eq. (30). Basically, Q-learning learns the Q-function using Temporal Difference (TD) method. The TD method are based on Bellman equation and solve equations without using system dynamics knowledge but using the observed data along a single trajectory of the system [25], [30]. TD method is used to update the value of a system trajectory based on the Bellman equation. The main idea of TD method is update the value estimate to make the TD error small [25], [30]. The TD-Error formula for Q-Learning is formulated in Eq. (27) with $\gamma = 1$.

$$e_k=-Q(y_k,u_k)+r(y_k,u_k)+Q(y_{k+1},u_{k+1}) \tag{27}$$

Substitute Eq. (30) into (27) became Eq. (28). The policy evaluation step in RL method is based on the Bellman TD error in Eq. (28). Eq. (28) implies that the Bellman equation could performed using only the measured data, not the state [1].

$$e_k=-z_k^TP^{\mathcal{D}}z_k+y_k^TR_yy_k+u_k^TR_uu_k+z_{k+1}^TP^{\mathcal{D}}z_{k+1} \tag{28}$$

The TD method will be used to estimate the kernel matrix $P^{\mathcal{D}}$ without requiring information about system dynamics (A, B). However, the process of estimating the kernel matrices $P^{\mathcal{D}}$ can be reached through the measurement data trajectories of y_k and u_k from the system.

The Q-function can be expressed as a hypothesis function as in Eq. (29) where the vector $P^{\mathcal{D}}$ is the unknown parameter while the $\phi(z_k)$ is the vector basis.

$$Q(z_k)=(P^{\mathcal{D}})^T\phi(z_k) \tag{29}$$

In discrete time-LQR case, the $\phi(z_k)$ is a quadratic form and consist of output signal y_k and control signal u_k.

The computation of Q-Function in Eq. (25a and (25b) could be formulated in Eq. (30) where the Kronecker product $\otimes$ and $\text{vec}(P^{\mathcal{D}})$ the vector formed by stacking the column of $P^{\mathcal{D}}$ matrix [25].

$$Q(y_k,u_k)=\frac{1}{2}z_k^TP^{\mathcal{D}}z_k=\frac{1}{2}\text{vec}^T(P^{\mathcal{D}})(z_k\otimes z_k) \tag{30}$$

Kronecker product could improve the computation process in the sense of control system and system identification [31].

The Q-function consists of the control signal u_k as an argument so that $\partial((P^{\mathcal{D}})^T\phi(z_k))/\partial u_k$ could be explicitly computed using Eq. (31a) [25]. The chain rule is used to solve the derivative process (see Eq. (31a)-(31b)).

$$\frac{\partial Q(y_k,u_k)}{\partial u_k}=\left(\frac{\partial z_k}{\partial u_k}\right)^T\left(\frac{\partial \phi(z_k)}{\partial z_k}\right)^T\left(\frac{\partial (P^{\mathcal{D}})^T\phi(z_k)}{\partial \phi(z_k)}\right)^T \tag{31a}$$

$$\frac{\partial Q(y_k,u_k)}{\partial u_k}=\left(\frac{\partial z_k}{\partial u_k}\right)^T\left(\frac{\partial \phi(z_k)}{\partial z_k}\right)^TP^{\mathcal{D}} \tag{31b}$$

The first section in Eq. (31a) could be solved using Eq. (32) where the control, output signal, and augmented state are denoted as $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$, and $z_k \in \mathbb{R}^{m+p}$, respectively.

$$\left(\frac{\partial z_k}{\partial u_k}\right)^T=\begin{bmatrix}I_m & 0_{m\times p}\end{bmatrix}^T \tag{32}$$

Eq. (33) is used to solve the second section from Eq. (31a) where the computation of the gradient $\nabla\phi$ are crucial to be done.

$$\frac{\partial \phi(z_k)^T}{\partial z_k}=\nabla\phi^T \tag{33}$$

The basis vector $\phi(z_k)$ that called as quadratic polynomial function consists of the pairwise product from component z_k such that could be formulated in Eq. (34) [25].

$$\phi(z_k)=z_k\otimes z_k;\ \phi(z_k)\in\mathbb{R}^{(m+p)^2} \tag{34}$$

Thus, the gradient of ϕ^T could be formulated using Eq. (35).

$$\frac{\partial \phi(z_k)^T}{\partial z_k}=I_{m+p}\otimes z_k+z_k\otimes I_{m+p} \tag{35}$$

Using these equations we could approximate the Q-function parameters without knowing the system matrices A, B. The next section would be tell us about LSTM network that we use as a function approximation.

B. LSTM

Fig. 3 illustrates the way to approximate the Q-function via the LSTM network. From Eq. (21), it can be concluded that computation of Q-function is dependent on $\bar{z}_k$ and $\bar{z}_{k+1}$. For that reason the recurrent scheme is employed to store the previous information to approximate Q-function. The fully connected layer in Fig. 3 is applied to estimate the controller gain $K_k^{\mathcal{D}}$. LSTM is one of the schemes proposed by Hochreiter in 1997 to overcome the vanishing gradient problem in the RNN structure [23]. This is because the initial idea of LSTM is to decide when to forget or store information to the next stage [28]. The LSTM architecture consists of two layers (state), namely hidden layers and cells [23], [32]. All outputs from each block are reconnected to the input blocks and all gates [28]. Unit cells and gates appear to be in a hidden state [23]. Each existing cell has several gates that are used to

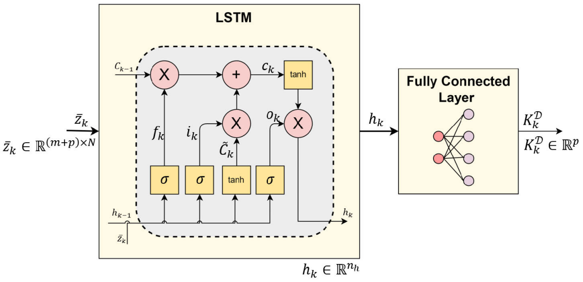


FIGURE 3. The details of DRQN.

regulate information (input) [32]. Function ⊙ is element-wise product. Weights in LSTM layer defined below:

- Input weights: $W_f,\ W_i, W_c,\ W_o \in \mathbb{R}^{n_h\times(m+p)}$
- Recurrent weights: $R_f,\ R_i, R_c,\ R_o \in \mathbb{R}^{n_h\times n_h}$
- Bias weights: $b_f,\ b_i,\ b_o$

In forget and input gate which denoted as f_k and i_k, the formulas are respectively described in Eq. (36) and (37). Input and recurrent data from the previous state are added up. A Hadamard product of two vectors is represented by ⊙. The function $g(.)$ in Eq. (38) and (41) are hyperbolic tangent function.

$$f_k = \sigma\left(W_f\bar{z}_k + R_f h_{k-1} + b_f\right) \tag{36}$$

$$i_k = \sigma\left(W_i\bar{z}_k + R_i h_{k-1} + b_i\right) \tag{37}$$

$$\tilde{C}_k = g\left(W_c\bar{z}_k + R_c h_{k-1}\right) \tag{38}$$

Connections between the cell to all gates are added to the architecture to make precise timing easy to learn [23]. Eq. (39) describes the formulation in a cell.

$$c_k = i_k \odot \tilde{C}_k + f_k \odot c_{k-1} \tag{39}$$

The output gate denoted as o_k, formulation represented in Eq. (40). While the block output is denoted in Eq. (41) and the h_k represents the output of LSTM network.

$$o_k = \sigma\left(W_o\bar{z}_k + R_o h_{k-1} \odot c_k + b_o\right) \tag{40}$$

$$h_k = o_k \odot g\left(c_k\right) \tag{41}$$

This scheme for training the LSTM is an extension of the standard back-propagation algorithm known as Back-Propagation Through Time (BPPT) [23]. The use of LSTM in this study is called stable in the sense of NN learning if Eq. (42)-(44)are fulfilled and let $||f||_\infty = \sup_k ||f_k||_\infty$. The computation of supremum function in this research was developed in Appendix VI. Meanwhile, the proof has already been developed in [29]. The term stable in this research refers to: when the gradient does not explode or converge to a stationary point [28], [29]. The exploding gradient problem refers to increasing the norm of the gradient during the learning process. This can cause network output to grow exponentially [33].

$$||R_i||_\infty, ||R_o||_\infty < (1 - ||f||_\infty) \tag{42}$$

$$||R_c||_\infty < 0.25(1 - ||f||_\infty) \tag{43}$$

$$||R_f||_\infty < (1 - ||f||_\infty)^2 \tag{44}$$

After implementing the Algorithm 1, the next step is computing the control signal trajectories using Eq. (45).

$$u_k = -K_k^{\mathcal{D}} y_k \tag{45}$$

Algorithm 1 OPFB Based on LSTM

```
Initialization:
  Input: Data-driven augmented system sequences $\bar{z}_k \in \mathbb{R}^{(m+p)N}$
  Target: Reward signal $r(y_k, u_k) \in \mathbb{R}^N$ (see the Eq. (6))
  $\mathcal{E}$: Number of epoch
  $i$: Iteration index
  while $i < \mathcal{E}$ do
      Compute the forget, input, cell, and output gate which formulated in Eq. (36), (37), (40) and (39), respectively.
      Compute the hidden state using Eq. (41).
      Define the hidden state as Q-function, denoted as $Q^{\mathcal{D}}$
      Update the weights and bias using BPTT
      Check the learning stability of LSTM networks based on Eq. (42)-(44)
      Compute the Fully Connected (FC) layers with the dataset $Q^{\mathcal{D}}$ and controller gain that obtain from model-based solution (Eq. 5)
      Compute the iteration $i = i + 1$
  end while
Output: Define the output from fully connected layer as the controller gain $K_k^{\mathcal{D}}$
```

In the [34], the RL based RNN scheme was operated using two step procedures, namely system identification and control. In this research, the network is simplified by only using a single LSTM influenced by [35]. If the action space is relatively small but the state space is partially observable, a direct value function approximation approach can be adopted with the basis of RNN [35].

IV. STABILITY ANALYSIS

In this research, we define the combination of the state and controller as augmented state. The augmented state for every sample k is denoted as ζ_k and formulated in Eq. (46) where the state $\zeta_k = \begin{bmatrix} x_k & u_k \end{bmatrix}^T$.

$$\zeta_{k+1} = \Lambda_k \zeta_k \tag{46}$$

where the Λ_k is formulated in Eq. (47) and the state, input, and output matrix are denoted as A, B, and C, respectively.

$$\Lambda_k = (A - BK_k^{\mathcal{D}}C) \tag{47}$$

The computation of $K_k^{\mathcal{D}}$ could be represented into nonlinear function of the control and output signals are denoted as u_k and y_k, respectively.

We can view the $K_k^{\mathcal{D}}$ as a time varying matrix that has a specific value in a specific time index k. By using this point of view, we can perform the stability analysis of the closed-loop system as a *finite-time stability analysis for linear time-varying system*. We will discuss the stability property in Definition 1.

Definition 1: Let $\Phi(k, 0)$ is the evolution operator of Eq. (46), i.e. $\Phi(k, 0) = \Lambda_k \Lambda_{k-1} \ldots \Lambda_0$. We say that Eq. (46) is stable in a finite horizon for $k = 0, 1, \ldots, N-1$ if and only if $||\Phi(k, 0)||$ is bounded for $k = 0, 1, \ldots, N-1$.

A necessary and sufficient condition for stable in a finite horizon, is given by the following proposition.

Proposition 1: The system (46) is stable in a finite horizon $k = 0, 1, \ldots, (N-1)$, if and only if $||\Lambda_k||$ is bounded for all $k = 0, 1, \ldots, (N-1)$ [36].

Proof 1: It is obvious from the definition of the evolution operator $\Phi(k, 0)$.

Moreover, observe that by following the design procedure which satisfies the learning convergence requirements in Eq. (42)-(44), the $\Phi(k, 0)$ is bounded and it implies the norm of $||\Lambda_k||$ will be bounded for $k = 0, 1, \ldots, N-1$. Therefore under Proposition 1, our closed-loop system will be stable in finite horizon according to the Definition 1.

V. SIMULATION STUDY

In this research, the DRQN based on LSTM is tested using four case studies: cart-pole, batch distillation, an unstable plant, and satellite attitude system. All these simulations are uploaded in[1] using MATLAB. In each case study, the proposed algorithm is tested by variating the discount factor and compare three methods, specifically:

- 1st method: Model-based (blue graph)
- 2nd method: Q-Learning (red graph)
- 3rd method: DRQN based on LSTM (magenta graph)

The proposed method for the first and second case studies have been published on [22]. The third case study is inspired by [17]. The simulation specifies the use-defined performance index as $R_y = 10$ and $R_u = 0.1$.

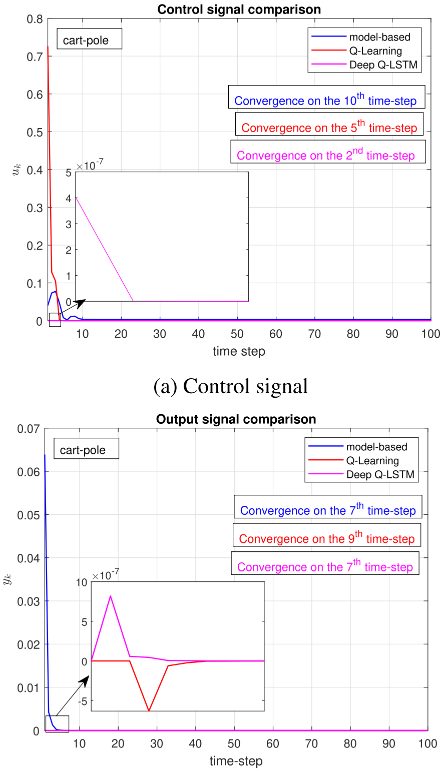
TABLE 1. Nomenclature of Cart-Pole system.

Symbol	Description	Value	Unit
g	Gravity	9.8	m/s^2
l	Pole length	0.5	m
m_p	Pole's mass	0.1	kg
m_c	Cart's mass	1	kg
m_t	Total mass	1.1	kg

A. CASE STUDIES

1) 1ST CASE STUDY: CART-POLE SYSTEM

One of the classic control problems is a cart-pole system. The objective of this case study is to apply the forces u_k to a cart moving along a track and keep the pole hinged to the cart. This model is selected deliberately for its simplicity in demonstrating the aims of this research. The dynamics of the cart-pole system represented in Eq. (48) with the parameter value summarized in Table 1 from the technical detail in.[2] The plant dynamics is formulated in Eq. (48) in which the state variables x_1, x_2, x_3, and x_4 are cart's position, cart's velocity, pole's position, and pole's velocity, respectively.



(a) Control signal

(b) Output signal

FIGURE 4. Performance comparison in which the blue, red, and magenta graph represent the control and output signal trajectories for the 1st case study obtained from model-based, Q-Learning, and DRQN based on LSTM, respectively (a) control signal trajectories (b) output signal trajectories.

$$x_{k+1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m_p}{m_c}g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m_p+m_c}{lm_c}g & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ \frac{1}{m_c} \\ 0 \\ \frac{1}{lm_c} \end{bmatrix} u_k$$

$$y_k = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} x_k \tag{48}$$

It's worth noting that in this first case study, the system applied is detectable or not fully observable but fully controllable (see Eq. (48)). The control signal obtained from the DRQN based on LSTM is shown in Fig. 4a which is denoted in the magenta graph holds the fastest convergence time. This is due to the use of the DRQN method which utilizes the use of an LSTM network to search for optimal control signals. Empirically, it can be seen that the use of the DRQN method is computationally faster than conventional optimal control solutions. The fastest convergence time is closely correlated to the maximum peak of the trajectories control signal in Table 2 for the 1st case study. This is due to the assistance of the LSTM network which carries out the learning process offline so that the energy produced by the control signal is minimized. The lowest point maximum value of control signal is generated by the third method, specifically

[1] https://github.com/adinovitarini/DRQN based on LSTM.

[2] https://github.com/openai/gym/blob/master/gym/envs/classic_control/cartpole.py

the DRQN based on LSTM. Fig. 4b shows the same result, which is also shown in the output signal trajectories in the first case study. The fastest convergence is achieved from the use of the third method, specifically DRQN based on LSTM which is denoted in the magenta graph.

2) 2ND CASE STUDY: BATCH DISTILLATION SYSTEM

The operation of the batch distillation process could be reviewed in Fig. 5. The boiler consists of a certain amount of solvent (water and ethanol) which is denoted as the amount of solvent (M_B), concentration (X_B), and composition of steam in boiler (Y_B) [37]. The temperature in the boiler will be increased to a certain value, wherein in this study the temperature in the boiler was set to around 78^0 to 80^0 Celsius. This is because the purpose of this heating is to separate the vapor phase of ethanol from water. Where the boiling point of ethanol is at 78^0. Then the solvent vapor has then flowed into condenser 1 and condenser 2. In the initial phase, ethanol with a lower boiling point will evaporate more than water. The amount of ethanol will decrease as the boiling point of the solvent continues to rise and only water will remain in the boiler. Whereas, the distillate concentration which remains in the product tank is denoted with X_D. To regulate the amount of reversal mixture which is distributed to the distillate, we have to control the reflux valve. It could be done by controlling the amount of on or off (duty cycle) of the reflux valve. To implement this idea of the closed-loop system, a controller is needed in this system to keep the results of the distillate concentration as desired. In the schematic above, vapor (V), reflux (R), distillate (D), R_0 (constant) is the initial condition for reflux flow rate when the valve is closed. The reflux ratio is developed with a range of $0 - 1$ which represents the 0% until 100% PWM. The identification system for the second case study was already published on [38].The state, input, and output matrices is define in Eq. (49).

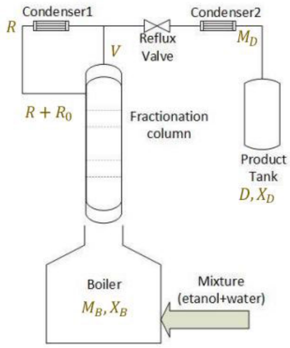


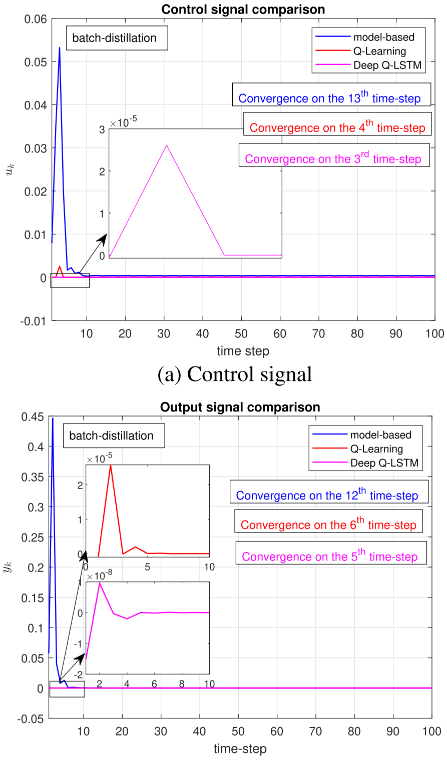
FIGURE 5. Batch distillation column schematic diagram.

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k
\end{aligned}
\tag{49}
$$

The state, input, and output matrices denoted as A, B, and C are defined in Eq. (50).

$$
\begin{aligned}
A &= \begin{bmatrix} 1.14 & -0.78 & -0.41 & -0.93 \\ 1.05 & 1.02 & 0.52 & 0.55 \\ -0.77 & 0.74 & -0.83 & 2.68e-03 \\ 1.18 & 0.95 & -0.65 & -0.79 \end{bmatrix} \\
B &= \begin{bmatrix} -1.37 \\ 0.45 \\ 1.08 \\ -0.38 \end{bmatrix} \\
C &= \begin{bmatrix} 0.73 & -0.78 & 0.97 & -0.34 \end{bmatrix}
\end{aligned}
\tag{50}
$$

In this 2nd case study, the system is fully observable and controllable (see Eq. (49) and (50)). The control signal trajectories in Fig. 6 shows the fastest convergence time achieved when using the DRQN based on LSTM which is illustrated with a magenta graph. In addition. In addition, the maximum peak of the control signal generated from the DRQN based on LSTM is also the smallest (see Table 2). Meanwhile, the output signal returns the same result in Fig. 6b. The implementation of the DRQN based on LSTM also proves that the resulting output signal holds the fastest convergence time (see the magenta graph). In this case study, comparisons can be made with previous research [22]. In this research, the design of the control method is based on the information model of the system. Meanwhile, in this research, we do not need a plant model. We only rely on system input and output signals to generate optimal control signals.



(a) Control signal

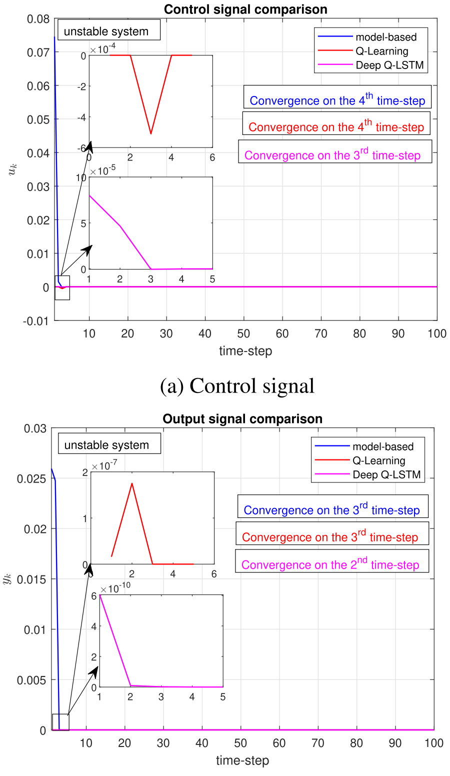
(b) Output signal

FIGURE 6. **Performance comparison in which the blue, red, and magenta graphs represent the control and output signal trajectories for the 2nd case study obtained from model-based, Q-Learning, and DRQN based on LSTM, respectively for (a) control signal trajectories (b) output signal trajectories.**

3) 3RD CASE STUDY: AN UNSTABLE PLANT

Consider an unstable DT system is formulated in Eq. (51) [17]. The open loop eigenvalues for this system is unstable, but controllable and observable. The eigenvalues of the open-loop system are 0.7 and 1.1.

$$x_{k+1} = \begin{bmatrix} 1.8 & -0.77 \\ 1 & 0 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k$$
$$y_k = \begin{bmatrix} 1 & -0.5 \end{bmatrix} x_k \tag{51}$$



(a) Control signal

(b) Output signal

FIGURE 7. **Performance comparison where the blue, red, and magenta graph are represent the control and output signal trajectories for the 3rd case study that obtain from model-based, Q-Learning, and DRQN based on LSTM, respectively for (a)control signal trajectories (b)output signal trajectories.**

The control signal trajectories in Fig. 7a points out that the fastest convergence time is achieved when using the DRQN based on LSTM illustrated in the magenta graph. Additionally, the maximum peak of the control signal generated from the DRQN based on LSTM is also the smallest (see Table 2). The output signal shown in Fig. 7b achieves the same result. The implementation of the DRQN based on LSTM also produces the resulting output signal that holds the fastest convergence time (see the magenta graph). In the third case study, comparisons can be made with previous research [17]. In this research, the control method design is based on system state information, so it is assumed that all states can be observed, which is certainly difficult to find in practice. Meanwhile, in our research, we have implemented the use of measured output signals from the system to generate optimal control signals in the third iteration. Meanwhile, in the [17], the optimal solution was obtained in the fourth iteration.

TABLE 2. **The Maximum peak comparison based on the convergence time is denoted as Conv. Time, the maximum of control signal is denoted as $M_p(u_k)$, and the Case study 1:cart-pole; Case study 2:batch distillation; Case study 3:unstable plant, and the Method 1: Model-based, Method 2:Q-Learning, Method 3: DRQN based on LSTM.**

Case	Method	Conv.Time		$M_p(u_k)$
		u_k	y_k	
1	1	10	7	1.19E-02
	2	5	9	1.16E-06
	3	2	7	3.02E-05
2	1	13	12	8.00E-02
	2	4	6	1.29E-03
	3	3	5	1.78E-05
3	1	4	3	7.45E-02
	2	4	3	2.43E-03
	3	3	2	1.65E-06

4) 4TH CASE STUDY: SATELLITE ATTITUDE

The focus of this study is on a compact spacecraft featuring three reaction wheels. Two prevalent methods for characterizing the spacecraft's orientation (e.g. Euler angles and quaternion) can be converted between each other. The mathematical depiction of the spacecraft's orientation is achieved through kinematic equations that connect angular position to angular velocity, along with dynamic equations that elucidate the change in angular velocity or, equivalently, angular momentum over time. The details modeling of satellite altitude motion was developed in [39]. The linear DT satellite dynamics are formulated in Eq. (52) where the input and output signals is Multiple Input Multiple Output (MIMO) case.

$$x_{k+1} = Ax_k + Bu_k^i, \quad \forall i = 1, \dots, 3$$
$$y_k^j = Cx_k, \quad \forall j = 1, \dots, 6 \tag{52}$$

The state, input, and output matrices are defined in where $A \in \mathbb{R}^{6\times6}$, $B \in \mathbb{R}^{6\times3}$, and $C \in \mathbb{R}^{6\times6}$ were represented in Eq. (53)-(55), as shown at the bottom of the next page.

Table 3 show for the 4th case study, the fastest convergence time of the control signals were obtain from DRQN based LSTM method. Empirically, the least maximum peak of the control signals were obtain from DRQN based LSTM method.

B. PERFORMANCE ANALYSIS

In this section we try to analyze the performance based on the empirical results. In this research, we use the average cost associated with the policy $u_k = Ky_k$ which represented in Eq. (56) [15].

$$\Omega(K) = \lim_{\tau\to\infty} \frac{1}{\tau} \sum_{t=1}^{\tau} r(y_t, u_t) \tag{56}$$

The Eq. (57) is used to obtain the average cost associated with the optimal controller gain K^* for deterministic discrete-time system.

$$\Omega(K^*) = \mathrm{Tr}((K^*)^T B^T PBK^*) + \mathrm{Tr}(P) - \mathrm{Tr}((A + BK^*)^T P(A + BK^*)) \tag{57}$$

TABLE 3. The maximum peak comparison for the 4th case study based on the convergence time is denoted as Conv. Time, the maximum of control signal is denoted as $M_p(u_k)$ for the 4th case study: satellite attitude control, and the Method 1: Model-based, Method 2:Q-Learning, Method 3: DRQN based on LSTM.

Method	Conv. Time		$M_p(u_k)$
	u_k	y_k	
1	$\begin{bmatrix}u_k^1\\u_k^2\\u_k^3\end{bmatrix}=\begin{bmatrix}5\\1\\1\end{bmatrix}$	$\begin{bmatrix}y_k^1\\y_k^2\\y_k^3\\y_k^4\\y_k^5\\y_k^6\end{bmatrix}=\begin{bmatrix}5\\5\\5\\4\\4\\4\end{bmatrix}$	$\begin{bmatrix}M_p(u_k^1)\\M_p(u_k^2)\\M_p(u_k^3)\end{bmatrix}=\begin{bmatrix}2\\1.32E-05\\2.57E-05\end{bmatrix}$
2	$\begin{bmatrix}u_k^1\\u_k^2\\u_k^3\end{bmatrix}=\begin{bmatrix}5\\1\\1\end{bmatrix}$	$\begin{bmatrix}y_k^1\\y_k^2\\y_k^3\\y_k^4\\y_k^5\\y_k^6\end{bmatrix}=\begin{bmatrix}9\\8\\8\\8\\7\\7\end{bmatrix}$	$\begin{bmatrix}M_p(u_k^1)\\M_p(u_k^2)\\M_p(u_k^3)\end{bmatrix}=\begin{bmatrix}8.64E-01\\2.2090\\9.06E-01\end{bmatrix}$
3	$\begin{bmatrix}u_k^1\\u_k^2\\u_k^3\end{bmatrix}=\begin{bmatrix}4\\1\\1\end{bmatrix}$	$\begin{bmatrix}y_k^1\\y_k^2\\y_k^3\\y_k^4\\y_k^5\\y_k^6\end{bmatrix}=\begin{bmatrix}6\\6\\7\\5\\5\\6\end{bmatrix}$	$\begin{bmatrix}M_p(u_k^1)\\M_p(u_k^2)\\M_p(u_k^3)\end{bmatrix}=\begin{bmatrix}6.68E-09\\1.41E-19\\1.65E-08\end{bmatrix}$

The performance analysis for the three methods are obtained using the relative error $\frac{|\Omega(K^*)-\Omega(K)|}{\Omega(K^*)}$ where the control gain obtained from the conventional Q-Learning or the DRQN based on LSTM are denoted as K [15]. In this study, we assess the relative error numerically, and the result is not significantly different from the control signal obtained from the conventional Q-Learning or the DRQN based on LSTM. The relative errors for the first until fourth case study are 99.89E-02, 99.91E-02, 99.96E-02, and 99.99E-02, respectively. The result show us that empirically, the model-free RL algorithm have been success to substitute the model-based algorithm to obtain the optimal policy.

C. LEARNING STABILITY RESULT OF LSTM NETWORKS

In this test, the learning results are as shown in Table 4 for the 1st until 4th case studies. This test is conducted to review the value of the norm infinity or the supremum function of recurrence weights at the input gate, output gate, forget gate and cell gate complied with the Eq. (42)-(44). Empirically, it can be observed that the value of the recurrent weights of learning results based on LSTM networks can meet the requirements of LSTM stability. The computational process of the supremum function in this study uses Algorithm 2 as in Appendix.

The convergence test of the LSTM network in this study summarized in Table 4. In the second and third rows, namely the supremum function of the repeated weights at the input gate and output gate is less than $1-||f||_\infty$. Then, in the fourth row, the highest function of the repeated weights on the cell gate is also less than $1/4$ from $1-||f||_\infty$. In the last line, the value of the supremum function of the recurrent weight of the forget gate is also smaller than the quadratic value of $1-||f||_\infty$. These results apply to all case studies.

TABLE 4. The supremum function of recurrent weights in input gate, output gate, cell gate, and forget gate denoted as R_i, R_o, R_c, and R_f, respectively.

Weights	Case Study			
	1	2	3	4
$1-\|\|f\|\|_\infty$	52.21	50.95	51.32	51.15
$\|\|R_i\|\|_\infty$	6.80	4.51	6.91	5.31
$\|\|R_o\|\|_\infty$	5.94	6.32	7.69	9.48
$\|\|R_c\|\|_\infty$	6.45	4.82	2.72	6.16
$\|\|R_f\|\|_\infty$	9.13	2.12	6.07	10.31

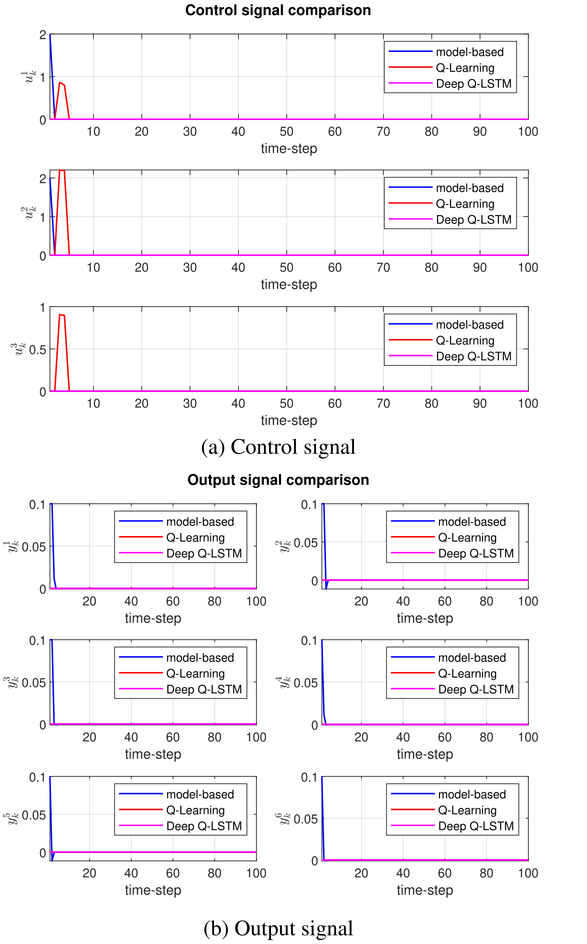
D. Q-FUNCTION COMPARISON

The trajectories of Q-function for the the 1st, 2nd, 3rd, and 4th case study are shown in Fig. 9, 10, 11, and 12, respectively. The process, subsequently, computes the Q-function trajectory using Eq. (7). The blue graph represent the Q-function that is obtained model-based method and denoted as

$$A=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\6.21E-07&2.01E-08&7.73E-08&-2.62E-04&5.74E-02&5.72E-02\\7.38E-08&8.69E-07&-1.71E-07&-5.82E-02&2.63E-04&-5.79E-02\\1.19E-07&-2.22E-08&1.03E-06&-4.38E-02&4.38E-02&-1.45E-06\end{bmatrix} \tag{53}$$

$$B=\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\5.74E-03&1.49E-05&1.12E-05\\1.49E-05&5.81E-03&1.14E-05\\1.12E-5&1.14E-05&4.37E-03\end{bmatrix} \tag{54}$$

$$C=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix} \tag{55}$$



(a) Control signal

(b) Output signal

FIGURE 8. Performance comparison where the blue, red, and magenta graph are represent the control and output signal trajectories for the 4th case study that obtain from model-based, Q-Learning, and DRQN based on LSTM, respectively. (a)control signal trajectories (b)output signal trajectories.

TABLE 5. Performance comparison based on the convergence time denoted as Conv. Time and the norm of Q-function denoted as $||Q(y_k, u_k)||$ where the Case study 1:cart-pole; Case study 2:batch distillation; Case study 3:unstable plant; Case study 4:satellite attitude, and the Method 1: Model-based, Method 2:Q-Learning, Method 3: DRQN based on LSTM.

Case Study	Method	Conv. Time	$\|\|Q(y_k, u_k)\|\|$
1	1	4	4.72E-02
	2	6	2.03E-04
	3	1	2.11E-08
2	1	5	2.20E-02
	2	4	4.08E-01
	3	3	3.15E-06
3	1	4	4.94E-02
	2	4	5.93E-05
	3	3	3.79E-09
4	1	4	6.00E-03
	2	4	2.27E-07
	3	3	1.59E-13

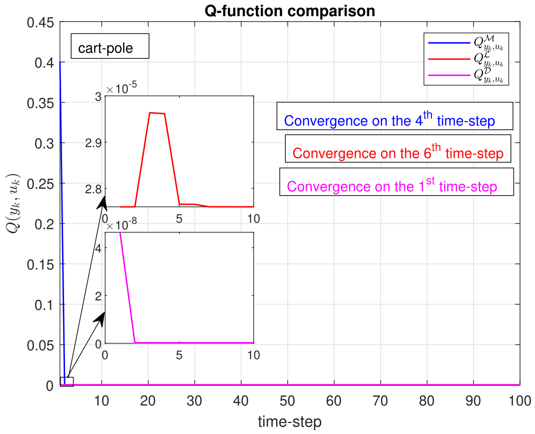


FIGURE 9. Q-Function trajectory of 1*st* case study.

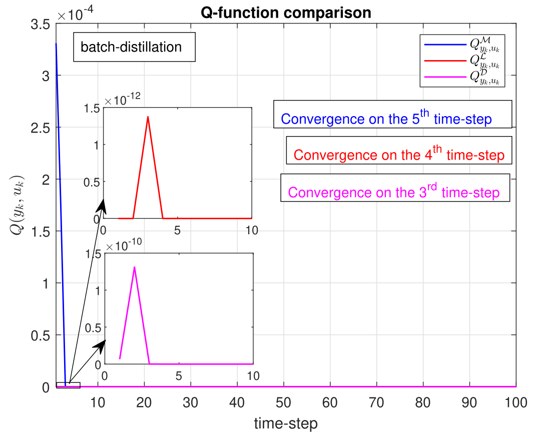


FIGURE 10. Q-Function trajectory of 2*nd* case study.

$Q^{\mathcal{M}}(y_k, u_k)$. These trajectories are executed by implementing the control signal u_k achieved from model-based method into the system and computing the output signal y_k. While the red graph represents the Q-function trajectory is achieved from Q-Learning method and denoted as $Q^{\mathcal{L}}(y_k^{\mathcal{L}}, u_k)$. The red graph in Fig. 9-12 are carried out after implementing the control signal obtained from Q-Learning based on the measured data which is formulated in Eq. (24). The magenta graph in Fig. 9-12 represents the Q-function trajectory which is achieved from the DRQN based on LSTM (see Algorithm 1) and denoted as $Q^{\mathcal{D}}(y_k^{\mathcal{D}}, u_k)$.

Performance comparison where the blue, red, and magenta graph are represent the Q-Function trajectories that obtain from model-based, Q-Learning, and DRQN based on LSTM, respectively for (a)1st case study (b)2nd case study (c)3rd case study (d)4th case. The next performance criterion from the proposed framework obtained the norm of Q-function in Eq. (7). The norm of Q-function in 1st case study are 4.72E-02. 2.03E-04, and 2.11E-08, respectively. The trajectory of Q-function from 1st case study from model-based, Q-Learning, and DRQN based on LSTM are converge on the 4th, 6th, and 1st time-step. The norm of Q-function in 2nd case study are 2.20E-02. 4.08E-01, and 3.15E-06, respectively. The trajectory of Q-function from 2nd case study from model-based, Q-Learning, and DRQN based on LSTM are converge on the 5th, 4th, and 3rd time-step. The norm of Q-function in 3rd case study are 4.94E-02, 5.93E-05, and 3.79E-09, respectively. The trajectory of Q-function from 3rd case study from model-based, Q-Learning, and DRQN

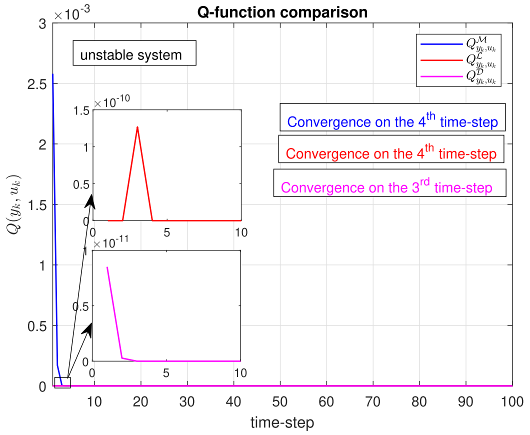


FIGURE 11. Q-Function trajectory of 3^{rd} case study.

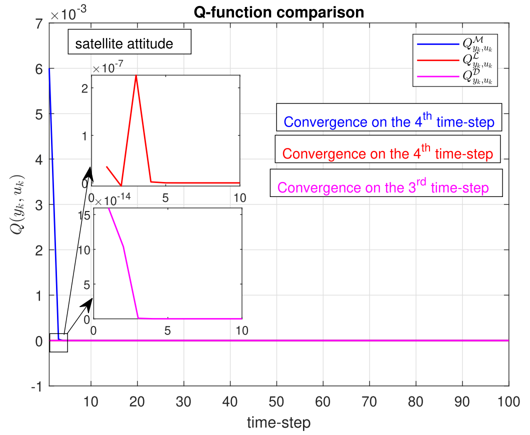


FIGURE 12. Q-Function trajectory of 4^{th} case study.

based on LSTM are converge on the 4th, 4th, and 3rd time-step. The trajectory of Q-function from 4th case study from model-based, Q-Learning, and DRQN based on LSTM are converge on the 4th, 4th, and 3rd time-step. Meanwhile, the norm of Q-function in 4th case study are 6.00E-03, 2.27E-07, and 1.59E-13, respectively.

VI. CONCLUSION

The control signal trajectory generated from DRQN based on LSTM is the smallest than model-based and Q-Learning method. The control signal trajectory generated from the DRQN based on LSTM is the smallest among the model-based and the Q-Learning method. The maximum peak value of the proposed algorithm carries the smallest value compared to other methods (model-based and Q-Learning). For the 1st, 2nd, and 3rd case studies, the maximum peak of the control signals are 3.02E-05, 1.78E-05, 1.65E-06, in corresponding order. The same results also apply to the 4th case study which is MIMO where the smallest value of the maximum peak of the control signals are obtain from DRQN based LSTM method. This result empirically indicates that our proposed algorithm can be applied to find the optimal control gain that minimizes the energy of its control signal.

The fastest convergence time in Q-function trajectories are carried out with the DRQN based on LSTM shown in Fig. 9-12. It is empirically discovered that the same result is proven by the norm values resulting from the Q-function trajectory. The norm of Q-function trajectory for our proposed algorithm on the 1st, 2nd, 3rd, and 4th case studies are 2.11E-08, 3.15E-06, 3.79E-09, and 1.59E-13, respectively.

APPENDIX

ALGORITHM FOR SUPREMUM FUNCTION

Supremum function is used to represent the $||f||_\infty$ that used to obtain the stability of LSTM (see Eq. (42)-(44)). The Algorithm 2 was used to do the numerical computation of the supremum function to define the learning stability in LSTM networks.

```
Algorithm 2 Supremum Function
Initialization: $\mathcal{N}$ ← Set of numbers
    if $\mathcal{N}$ =={} then
        $\mathcal{S}$ = NaN
    else
        $\mathcal{S}$ = max($\mathcal{N}$)
    end if
Output: $\mathcal{S}$ ← supreme function
```

REFERENCES

[1] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.

[2] H. Modares, F. L. Lewis, and Z.-P. Jiang, "Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2401–2410, Nov. 2016.

[3] S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement Q-learning control for the discrete-time linear quadratic regulator problem," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 1311–1316.

[4] J. Huang, *Nonlinear Output Regulation: Theory and Applications* (Advances in Design and Control). Philadelphia, PA, USA: SIAM, 2004. [Online]. Available: https://books.google.co.id/books?id=xLqECMHpGf0C

[5] Z.-P. Jiang, T. Bian, and W. Gao, "Learning-based control: A tutorial and some recent results," *Found. Trends® Syst. Control*, vol. 8, no. 3, pp. 176–284, 2020.

[6] H. J. Cruz Neto and M. A. Trindade, "Control of drill string torsional vibrations using optimal static output feedback," *Control Eng. Pract.*, vol. 130, Jan. 2023, Art. no. 105366. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066122001976

[7] K. Xie, Y. Zheng, W. Lan, and X. Yu, "Adaptive optimal output regulation of unknown linear continuous-time systems by dynamic output feedback and value iteration," *Control Eng. Pract.*, vol. 141, Dec. 2023, Art. no. 105675. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066123002447

[8] R. Self, M. Harlan, and R. Kamalapurkar, "Model-based reinforcement learning for output-feedback optimal control of a class of nonlinear systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 2378–2383.

[9] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2019. [Online]. Available: https://books.google.co.id/books?id=gNcRuQEACAAJ

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[11] H. Dong, Z. Ding, and S. Zhang, *Deep Reinforcement Learning*. Cham, Switzerland: Springer, 2020.

[12] X. Li, L. Xue, and C. Sun, "Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm," *Neurocomputing*, vol. 314, pp. 86–93, Nov. 2018.

[13] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M. B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014.

[14] F. A. Yaghmaie and F. Gustafsson, "Using reinforcement learning for model-free linear quadratic control with process and measurement noises," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 6510–6517.

[15] F. A. Yaghmaie, F. Gustafsson, and L. Ljung, "Linear quadratic control using model-free reinforcement learning," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 737–752, Feb. 2023.

[16] B. Kiumarsi, Frank. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, Dec. 2015.

[17] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1523–1536, May 2019.

[18] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ-policy iteration for discrete-time linear quadratic regulation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 635–649, Feb. 2023.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, and A. Graves, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Jan. 2015.

[20] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, Jun. 2013.

[21] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable MDPs," in *Proc. AAAI Fall Symp. Ser.*, 2015.

[22] A. N. Putri, C. Machbub, D. Mahayana, and E. Hidayat, "Data driven linear quadratic Gaussian control design," *IEEE Access*, vol. 11, pp. 24227–24237, 2023.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[24] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. Hoboken, NJ, USA: Wiley, 1972. [Online]. Available: https://books.google.co.id/books?id=mf0pAQAAMAAJ

[25] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control* (EngineeringPro collection). Hoboken, NJ, USA: Wiley, 2012. [Online]. Available: https://books.google.co.id/books?id=NFEYFmllK9QC

[26] W. Aangenent, D. Kostic, B. de Jager, R. van de Molengraft, and M. Steinbuch, "Data-based optimal control," in *Proc., Amer. Control Conf.*, 2005, pp. 1460–1465.

[27] K. Furuta and M. Wongsaisuwan, "Closed-form solutions to discrete-time LQ optimal control and disturbance attenuation," *Syst. Control Lett.*, vol. 20, no. 6, pp. 427–437, Jun. 1993.

[28] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[29] J. Miller and M. Hardt, "Stable recurrent models," 2018, *arXiv:1805.10369*.

[30] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[31] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. CS-25, no. 9, pp. 772–781, Sep. 1978.

[32] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020.

[33] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[34] A. M. Schäfer, "Reinforcement learning with recurrent neural networks," Ph.D. thesis, Osnabrück, Univ., Osnabrück, Germany, 2008. [Online]. Available: https://www.bibtex.com/t/template-phdthesis/

[35] B. Bakker, "Reinforcement learning by backpropagation through an LSTM model/critic," in *Proc. IEEE Int. Symp. Approx. Dyn. Program. Reinforcement Learn.*, Apr. 2007, pp. 127–134.

[36] A. N. Vargas, J. B. R. D. Val, and E. F. Costa, "On stability of linear time-varying stochastic discrete-time systems," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2007, pp. 2423–2427.

[37] A. S. Rohman, P. H. Rusmin, R. Maulidda, E. Hidayat, C. Machbub, and D. Mahayana, "Modelling of the mini batch distillation column," *Int. J. Electr. Eng. Informat.*, vol. 10, no. 2, pp. 350–368, 2018.

[38] A. N. Putri, C. Machbub, and E. M. Idris Hidayat, "Combination of Elman neural network and Kalman network for modeling of batch distillation process," in *Proc. 13th Asian Control Conf. (ASCC)*, May 2022, pp. 926–931.

[39] D. Mahayana, "Synthesis of data-driven LightGBM controller for spacecraft attitude control," *IEEE Access*, vol. 11, pp. 70238–70247, 2023.

ADI NOVITARINI PUTRI received the bachelor's degree in electrical engineering with majoring in control system engineering from Institut Teknologi Sepuluh Nopember (ITS), in 2019, and the master's degree (cum laude) in electrical engineering majoring in control engineering from Institut Teknologi Bandung (ITB), in 2021, where she is currently pursuing the Ph.D. degree with the Control and Computer Systems Research Group, School of Electrical Engineering and Informatics. Her current research interests include data-driven control and reinforcement learning.

EGI HIDAYAT (Member, IEEE) received the bachelor's degree in electrical engineering from Institut Teknologi Bandung (ITB), the Master of Science degree in control and information system from Universitat Duisburg-Essen, and the Ph.D. degree in electrical engineering from Uppsala University. He is currently a Lecturer with the School of Electrical Engineering and Informatics, ITB. His research interests include modeling and system identification, control and learning, and robotics.

DIMITRI MAHAYANA (Member, IEEE) received the bachelor's degree (cum laude) in electrical engineering from the Bandung Institute of Technology, in 1989, the Master of Engineering degree in electrical engineering from Waseda University, Tokyo, Japan, in 1994, and the Ph.D. degree (cum laude) from the Bandung Institute of Technology (ITB), in 1998. He is currently a Lecturer with the School of Electrical Engineering and Informatics, ITB. His research interests include nonlinear dynamical systems, time varying systems, control theory and convergence between control engineering, and data science.

CARMADI MACHBUB (Member, IEEE) received the bachelor's degree in electrical engineering from Institut Teknologi Bandung (ITB), in 1980, the D.E.A. degree in control engineering and industrial informatics, in 1988, and the Ph.D. degree in engineering sciences majoring in control engineering and industrial informatics from Ecole Centrale de Nantes, Université de Nantes, in 1991. He is currently a Professor with the School of Electrical Engineering and Informatics, ITB, and the Rector of Institut Teknologi Sains Bandung, Bekasi, Indonesia. His current research interests include control, machine perception, and intelligent systems.