Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Data Driven Linear Quadratic Gaussian Control Design

Adi Novitarini Putri, Carmadi Machbub (Member, IEEE), Dimitri Mahayana (Member, IEEE), and Egi Hidayat (Member, IEEE)
Control System and Computer Research Group, School of Electrical Engineering and Informatics, Institut Teknologi Bandung
e-mail: adinovitarini@gmail.com, carmadi@itb.ac.id, dimitri@itb.ac.id, egi.hidayat@itb.ac.id

Corresponding author: Adi Novitarini Putri (e-mail: adinovitarini@gmail.com).

This work is partially funded by School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB)

ABSTRACT The implementation of the Linear Quadratic Gaussian (LQG) scheme is often considered problematic as it requires a dynamic model of the system as a whole. The challenges come from state variables without a physical representation and the interference factors that affect the reading process. This paper presents and assesses a combination of methods to adapt the LQG scheme to a discrete-time linear system. The method KalmanNet constructed by the Long-Short Term Memory architecture (LSTM) is employed to replace the role of Kalman Filter (KF). The Value Iteration (VI) algorithm supersedes the role of the Linear Quadratic Regulator (LQR) controller in solving quadratic regulation issues. The assessment of the proposed algorithm on a cart-pole system and batch distillation column with a disturbance factor in uncorrelated Gaussian white noise is carried out in a simulated way under a discrete-time linear system. The result indicates that the solving of regulation problems through the conventional LQG method is not conclusive as the output response oscillation is still in progress. The combination of the KalmanNet and VI algorithm, as aforementioned, provides better results as it proves to solve the regulation problem as well as to compel the system output to converge.

INDEX TERMS optimal control, LQG, LSTM, state estimation, reinforcement learning

I. INTRODUCTION

Optimal control refers to a scientific application that is developed to find the optimal control strategy of a system through an optimization of its objective function [1] [2]. The traditional optimal control involves a plant model to generate the Algebraic Riccati Equation (ARE) [3]. The performance of the controller is heavily dependent on the model [4]. A paradigm shift has begun to emerge due to several weaknesses in the model-based control approach when viewed from an optimal control perspective.

The data driven scheme is classified into two types; model-based and model-free. The former involves directly searching the controller parameters based on cost values without any attempts at the model dynamics. While the latter involves data measurement to approximate the underlying dynamics [5]. Practically, the system model is often unknown, thereby it is necessary to identify the system using the input and output signal measurement data of the system. This scheme is referred to as a two-step control procedure for its design of new controllers as this enables an execution after the system identification stage [6] [7].

Linear Quadratic Gaussian (LQG) refers to a method that applies the principle of separation between state estimates and optimal controllers [1] [2]. The principle of separation states that the solution to the LQG problem is to utilize an observer based controller, which consists of Kalman Filter (KF) and Linear Quadratic Regulator (LQR) solutions. LQG combines the role of the KF and LQR as estimators and controllers [1] [8] [9]. The combination of these methods is able to handle the problem of regulation of linear systems with disturbance factors with statistical properties in the form of Gaussian. The traditional LQG method is known to possess a flaw of the system dynamics being linear and known. In addition, the system disturbance factor and measurements are stochastic, with their statistical characteristics also known in the form of a Gaussian distribution.

The development of machine learning methods progressed quite rapidly. One of the common ones is the Reinforcement Learning (RL). The terminology of RL is often referred to as adaptive optimal control [1] [10]. Several approaches to the RL method can be performed in order to produce an optimal strategy. The main dichotomy of the RL method is model-based one and the one without a model. The model-based RL method calculates the performance index by utilizing information from the environment [11]. The RL methods commonly employed to solve this problem are Policy Iteration (PI) and Value Iteration (VI). The adoption of the VI algorithm to solve linear quadratic problems based on the Bellman Equation has been widely conducted [12] [13] [14] [15]. These studies assume that all state information can be obtained. Past studies conducted concerning the use of the RL method in optimal control were mostly carried out under deterministic systems [16] [17].

The application of the VI algorithm has been performed to solve tracking problems in plants with unknown models [13]. An Artificial Neural Network (ANN) is utilized for the identification of the system to approximate the model. The network output is presumably a state vector but all states are assumed to be observable. The implementation proves to be challenging due to the limitations of sensors in a control process. In addition, the VI algorithm is also applied in systems that utilize the role of input-output plant data to approximate the control parameters [14]. They uses the Least-Square (LS) method from measurement data sets to solve the Bellman Equation online. The combination between KalmanNet and a conventional LQR controller was also proposed by [18]. The LS method from measurement data sets is employed to solve regulatory issues by utilizing plant input and output measurement data. These data are processed by KalmanNet to generate the estimated state $\hat{x}_k$. In comparison, the controller designed in [18] remains non-causal and can only be performed offline as it uses a conventional LQR solution.

This research proposes an algorithm specifically for a data-driven environment, not a fully measurable state, and a partially known dynamics model. Major contribution of this study are listed below :

- We adapt the LQG scheme for discrete-time linear system with Gaussian distribution of the disturbances characteristic using KalmanNet and VI algorithm
- We replace the state estimation scheme in LQG using KalmanNet. Meanwhwile, KalmanNet is a data-driven optimal filtering based on Recurrent Neural Networks (RNN) architecture
- We use the VI algorithm for controller design to solve the regulation problem. The VI algorithm is a model-based RL method that could solve the non-causality problem that arise in LQR solution

The remaining section of this paper covers the development of the proposed algorithm. Section II comprises the definition of the problem. Section III discusses the new data-driven LQG method as our proposed solution. The application of the proposed solution to the design data driven LQG control for cart-pole system and batch distillation column are included in Section IV, which also covers the simulations and evaluations of several test schemes. This arrangement is aimed at empirically ensuring that our proposed algorithm provides the most optimal results. Lastly, Section V contains the conclusions of this research. The notations presented in this research are listed in **Table 1**.

TABLE 1: Notation and abbreviations in this research

Symbol	**Description**
The subscript k	Time step
Z^{-1}	Delay operator
$(.)^\top$	Matrix transpose operation
w_k,v_k	Process noise and measurement noise
R_{ww}, R_{vv}	Process noise and measurement noise covariance matrices
$\hat{x}_k^-$,Σ_k^-	Prior estimated state and error covariance (before including the measurement y_k)
$\hat{y}_k^-$,Ξ_k^-	Prior estimated observed output and error covariance (before including the measurement y_k)
$\hat{x}_k$,Σ_k	Posterior estimated state and error covariance (after including the measurement y_k)
$\hat{y}_k$,Ξ_k	Posterior estimated observed output and error covariance (after including the measurement y_k)
x_k,u_k,y_k	State, control, and output vectors
n,m,p	State, control signal, and output dimensions
N	Final time
n_h	Number of hidden unit
Q,R	Weigh cost function on state trajectory and control signal
H_k,λ_k	Hamilton function and Lagrange multiplier
γ	Discount factor

II. PROBLEM FORMULATION

In the implementation of the control system, there is a limitation of the number of sensors used, consequently not all state information from the plant can be obtained. In addition, the data measurement process also often contains noise. As a result, implementing an optimal control scheme proves to be difficult. In order to solve the regulator problem, a system affected by disturbance factors, a scheme is required to be implemented to deal with this situation. This section covers the complete model information requirements on KF as a state estimation method and the non-causality that appears in conventional LQR solutions. These problems can affect the LQG controller design process. In this research, we classify three method combinations (see Problem 1-3 in Section II.C) to adapt the LQG controller scheme to deal with the issue above.

A. KF

The dynamics of a discrete-time linear system can be expressed as in Eq. (1) with $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$ are vectors of state variables, control signals, and measured output signals, respectively. Meanwhile, w_k and v_k are uncorrelated white Gaussian noise with the covariance matrix denoted as R_{ww} and R_{vv}, respectively.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, w_k \sim \mathcal{N}(0, R_{ww}) \\ y_k &= Cx_k + v_k, v_k \sim \mathcal{N}(0, R_{vv}) \end{aligned} \tag{1}$$

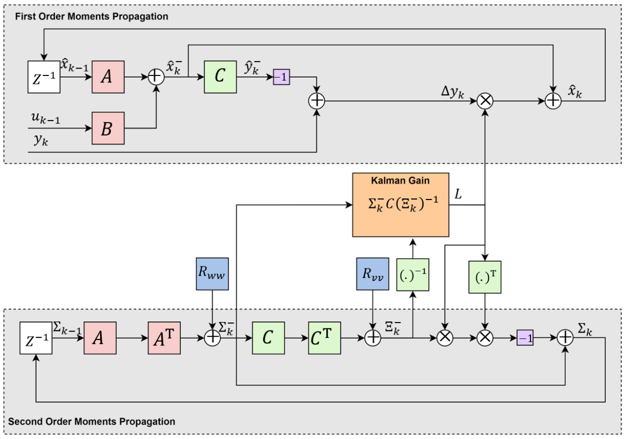

FIGURE 1: Kalman filter conventional

It is assumed that the system is observable. We can estimate the state $\hat{x}_k$ of the state x_k based on observation of the measured output y_k. KF is a recursive linear MSE filter that is also MSE optimal for the SS model in Eq. (1) [19]. From Table 1, it can be concluded the abbreviations of KF. The KF design is achieved through a two-step procedure : *prediction* and *update* which illustrated in Fig. 1.

1) Prediction step : This step involves computing a prior first and second order moment moments from the state trajectory using Eq. (2) and (3) respectively.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2}$$

$$\Sigma_k^- = A\Sigma_{k-1}A^\top + R_{ww} \tag{3}$$

In this step, we compute a prior first and second order moments from observed output using Eq. (4) and (5).

$$\hat{y}_k^- = C\hat{x}_k^- \tag{4}$$

$$\Xi_k^- = C\Sigma_k^- C^\top + R_{vv} \tag{5}$$

2) Update step : This step processes a posterior first order moments using Eq. (6) where the Δy_k is defined in Eq. (7).

$$\hat{x}_k = \hat{x}_k^- + L_k\Delta y_k \tag{6}$$

$$\Delta y_k = y_k - C\hat{x}_k^- \tag{7}$$

The propagation of posterior second order moments is computed with Eq.(8). Meanwhile the Eq. (9) is used to compute the Kalman gain.

$$\Sigma_k = \Sigma_k^- - L_k\Xi_k^- L_k^\top \tag{8}$$

$$L_k = \Sigma_k^- C^\top \left(\Xi_k^-\right)^{-1} \tag{9}$$

It is practically a demanding task to design and implement the optimal estimator since the system dynamics and noise statistics are unknown [20].

B. LQR

The second problem arises in the LQR problem. LRQ requires the solution of a Riccati equation given as a function of the plant's state-space model [4]. Additionally, the solution of the HJB equation for the LQR problem is non-causal. Furthermore, the VI method can be employed to solve the HJB equation for optimal control problems online [20].

In [1] [8], the performance index is the quadratic function as formulated in Eq. (10).

$$J_k = \frac{1}{2}\left[x_N^\top Q_N x_N + \sum_{i=k}^{N-1} x_i^T Q x_i + u_i^T R u_i\right] \tag{10}$$

$x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are respectively system state and control input. The cost-weighting matrices Q_N, Q, R are symmetric positive semi-definite matrices. The objective of the regulator problem is to find an u_k policy capable of minimizing the performance index in Eq. (10) during the system trajectory(1).

We begin with the Hamiltonian function (see Eq. (11).

$$H_k = \frac{1}{2}\left(x_k^\top Q x_k + u_k^\top R u_k\right) + \lambda_{k+1}^\top\left(Ax_k + Bu_k\right) \tag{11}$$

Hamiltonian function refers to an approach adopted for analyzing the optimization of the performance index [1]. The λ is a Lagrange multiplier chosen to solve the constraint optimization problem [1] [2]. The necessary condition for a minimum point of performance index that also fulfills the constraint system is represented in Eq. (12)-(14). The Eq. (12) and Eq. (13) are known as the state and co-state equation. While the Eq. (14) is a stationary condition [1].

$$\frac{\partial H_k}{\partial \lambda_{k+1}} = Ax_k + Bu_k \tag{12}$$

$$\frac{\partial H_k}{\partial x_k} = Qx_k + A^\top \lambda_{k+1} \tag{13}$$

$$0 = \frac{\partial H_k}{\partial u_k} = Ru_k + B^\top \lambda_{k+1} \tag{14}$$

From the stationary condition in Eq. (14), the control signal is obtained through Eq. (15)

$$u_k = -R^{-1}B^\top \lambda_{k+1} \tag{15}$$

Assume that a linear relation for the co-state and state equation in Eq. (16) where P_k is an intermediate sequence of $n \times n$ matrices.

$$\lambda_k = P_k x_k \tag{16}$$

Then proceed to substitute the linear relation in Eq. (16) to co-state equation in Eq. (12) became Eq. (17).

$$P_k x_k = Qx_k + A^\top P_{k+1} x_{k+1} \tag{17}$$

The backward recursion for P_k using matrix inversion lemma is obtained through Eq. (18) [1].

$$P_k = A^\top\left[P_{k+1} + BR^{-1}B^\top\right]^{-1}A + Q \tag{18}$$

Substitute the Eq. (18) to (16), thus the control signal in Eq. (15) became Eq. (19).

$$u_k = -R^{-1}B^\top \left(A^\top \left[P_{k+1} + BR^{-1}B^\top \right]^{-1} A + Q \right) x_k \quad (19)$$

The control gain is defined in Eq. (20)

$$K_k = R^{-1}B^\top P_k \quad (20)$$

It is conclude that the Eq. (18) is the Riccati equation solution that is computed and stored in the computer memory before the control is applied to the plant [1] [20]. Consequently, this conventional method is unfeasible to be implemented online.

C. LQG

The performance index with weight matrix Q and R are positive semi-definite and positive definite, respectively, as in Eq. (21). This research focuses on optimizing the cost function Eq. (21) which is the $\mathbb{E}\{.\}$ represent the expected value.

$$J(x_k, u_k) = \frac{1}{2}\mathbb{E}\left\{ x_N^\top Q_N x_N + \sum_{k=1}^{N-1} x_k^\top Q x_k + u_k^\top R u_k \right\} \quad (21)$$

In this research, the close-loop dynamics is jointly described by Eq. (22)

$$\zeta_{k+1} = \Lambda_k \zeta_k + \Gamma_k \mu_k \quad (22)$$

In which $\zeta = [x^\top \tilde{x}^\top]^\top \in \mathbb{R}^q$ consists of the state and estimation error. Meanwhile, the $\mu = [w^\top v^\top]^\top$ is the white noise, and $\Lambda \in \mathbb{R}^{q\times q}$ is the close-loop matrix is given as in Eq. (23).

$$\begin{bmatrix} x_{k+1} \\ \tilde{x}_{k+1} \end{bmatrix} = \begin{bmatrix} A - BK_k & BK_k \\ 0 & A - L_k C \end{bmatrix} \begin{bmatrix} x_k \\ \tilde{x}_k \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -L_k \end{bmatrix} \begin{bmatrix} w_k \\ v_k \end{bmatrix} \quad (23)$$

The calculation of the K_k and L_k matrices are formulated as in Eq. (20) and (9).We proceed to test our combined method with four types of combinations. The first, second and third combinations are defined in Problem 1, 2, and 3, correspondingly. Problems 1 to 3 are similar to the conventional LQG problems, the difference lies in the methods used in designing the observer and controller.

Problem 1: Consider the dynamical system in Eq. (1). Design the observer gain L_k and controller gain K_k so as to minimize the performance index in Eq. (21). The observer gain is obtained from the KF method. Consequently, it is required to process the tuning of the covariance matrix of measurement noise R_{vv}. At the same time, the controller gain K_k is obtained through the VI algorithm.

Problem 2: Consider the dynamical system in Eq. (1). Design the observer gain L_k and controller gain K_k so as to minimize the performance index in Eq. (21). The observer gain is obtained through KalmanNet. In consequence, it is required to process the tuning of the LSTM parameters, namely optimization method, number of hidden states, and the activation function. The controller gain K_k is obtained through the LQR method.

Problem 3: Consider the dynamical system in Eq. (1). Design the observer gain L_k and controller gain K_k so as to minimize the performance index in Eq. (21). The observer gain is obtained through the KalmanNet. Consequently, it is required to process the tuning of the LSTM parameters, namely optimization method, number of hidden states, and the activation function. The controller gain K_k is obtained through the VI algorithm.

In Problem 1, the conventional KF method is employed to produce an estimated state. This study assumes that the value of the measurement noise covariance matrix is greater than that of the process noise. The tuning of the covariance matrix parameter R_{vv} consequently has a more excellent value than that of R_{ww}. Next in the controller design, the implemented program in the Algorithm 2 is operated to calculate the controller gain value online (not backwards-in-time) similar to a conventional LQR solution.

The solution to Problem 2 and Problem 3 involves the role of KalmanNet to generate an estimated state of $\hat{x}_k$. KalmanNet adapts the conventional KF method based on ANN. ANN is utilized to predict the Kalman gain value based on input and output plant measurement data only without the requirement of information about the statistical characteristics of measurement of process noise. However, the tuning scheme is devised when using KalmanNet to compare the hyper parameters in ANN. Hyper parameter tuning is used in this study to vary the optimizer type, mini-batch size, and activation function to further compare system performances.

III. ADAPTATION LQG METHOD

The combination of methods proposed in this study adapt a finite-time LQG scheme to solve regulatory problems. The adaptation scheme is operated to utilize the role of ANN for replacing the function of the KF as a state estimation method. The RL method returns the conventional optimal controller designed using the LQR method. Additionally, stability analysis based on the evolution of the eigenvalues of a closed-loop system is performed to ensure the system's stability when controlled through a model-based RL method. The proposed algorithm is as shown in Fig. 2, as in this study there are two subsystems, namely the estimator and the controller. A more detailed discussion of the estimator proposed in this study is presented in Section III.A. In contrast, the discussion regarding the controller is presented in Section III.B.

A. KALMANNET

The calculation of the Kalman gain previously discussed in [19] indicates that the calculations are based on a system's model. KalmanNet is a Kalman gain calculation mechanism processed in a hybrid of model-based and data-based, combining the ANN with the conventional KF. The first in implementing KalmanNet is to build the SS model to design a recursive filter that operates as a KF. At this stage, it is assumed that the constants of the state matrix A, the

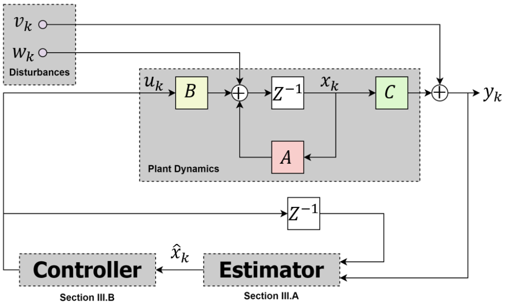

FIGURE 2: Proposed algorithm scheme

input matrix B, and the measurement model C are known, although not accurately. The covariance matrices R_{ww} and R_{vv} need not be known. The problem is to change the statistical process of the state propagation, as shown in Fig. 3, to obtain a Kalman gain (L) via ANN. Some questions arising before designing KalmanNet include (i) What input signal does ANN need to learn the Kalman gain value? (ii) What is the required ANN architecture to supersede the role of the KF? (iii) How can the ANN do the learning from data only? [19].

In the traditional KF, the Kalman gain (L) is not dependent on the current observation data y_k, not is it a function of the estimated state $\hat{x}_k$. The process of calculating the Kalman gain (L) through the KF method is based on the second-order statistical moment Σ_k. Therefore, the implementation of KalmanNet requires using the ANN, which has a memory element [19]. This follows the Long Short-Term Memory (LSTM) to adapt the KalmanNet scheme. This scheme is successfully implemented as a state estimation method on the batch distillation column system [21].

KalmanNet's learning process is carried out offline (in a supervised learning manner) using training data. Then, the model obtained from the training process will be used to calculate the Kalman gain value. In KalmanNet, the previous posterior estimated state $\hat{x}_{k-1}$ and delayed control signal u_{k-1} are used to calculate the prior estimated state $\hat{x}_k^-$ based on the system dynamics model. Meanwhile, the control signal in this research is obtained from LQR and VI algorithms (details in Section III.B). The estimated state $\hat{x}_k^-$ is subsequently used to make predictions for the next prior observed output $\hat{y}_k^-$. The calculation of the observation error denoted as Δy_k is based on the measured output y_k and the previous prior observed output $\hat{y}_k^-$. The process of calculating the posterior estimated state update $\hat{x}_{k+1}$ employs Kalman gain L and Δy_k.

The KalmanNet scheme adopted in this study is slightly different from that in [19]. Fig. 4 denotes the KalmanNet scheme adopted in this research, consisting of three subsystems. The first subsystem combines information from the current observation data y_k and the estimated state $\hat{x}_k$ into an input signal denoted as φ_k. The output of an LSTM layer is the hidden state h_k. The hidden state represents the covariance matrices in the sense of KF [19]. The subsystem is a fully connected layer. This layer is responsible for reconstructing the Kalman gain dimension (L). The KalmanNet scheme proposed in this study is stated in more detail in the **Algorithm 1**.

The structure of a single LSTM represented in [22]. LSTMs are designed to block the long-term dependency problems arising in typical RNN structures. The LSTM primary key is in the cell state. Three gates aimed at deleting or adding information to the next cell are inside a cell. The three gates consists of forget, input, and output gates. Each gate produces an output in the range of values 0 to 1. If the gate outputs is 0, no information is sent to the next cell, and vice versa.

From Fig. 4 we could conclude that the φ_k is input vector at time k for LSTM. Weights in LSTM layer defined below:

- Input weights : W_f, W_i,W_c, $W_o \in \mathbb{R}^{n_h \times (n+p)}$
- 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 ι_k, the formula are respectively described in Eq. (24) and (25). Input data and recurrent from the previous state are added up. A Hadamard product of two vectors is represented by $\odot$. The function $g(.)$ in Eq. (26) and (29) are hyperbolic tangent function.

$$f_k = \sigma\big(W_f \varphi_k + R_f h_{k-1} + b_f\big) \tag{24}$$

$$\iota_k = \sigma\big(W_i \varphi_k + R_i h_{k-1} + b_i\big) \tag{25}$$

$$\tilde{C}_k = g\big(W_c \varphi_k + R_c h_{k-1}\big) \tag{26}$$

Connections between the cell to all gates are added to the architecture to make precise timing easy to learn [22]. Eq. (27) describes the formulation in a cell.

$$c_k = \iota_k \odot \tilde{C}_k + f_k \odot c_{k-1} \tag{27}$$

The output gate denoted as o_k, formulation represented in Eq. (28). Meanwhile the block output was denoted in Eq.(29) and the h_k is representing the output of LSTM network.

$$o_k = \sigma\big(W_o \varphi_k + R_o h_{k-1} \odot c_k + b_o\big) \tag{28}$$

$$h_k = o_k \odot g\big(c_k\big) \tag{29}$$

This scheme for training the LSTM is an extension of the standard back-propagation algorithm known as Back-Propagation Through Time (BPPT) [22].

B. VALUE ITERATION ALGORITHM

The proposed algorithm in this research, specifically for the VI algorithm, is inspired by [13] [14] [15], developing the VI algorithm for Linear Quadratic Tracking (LQT) problem. In this subsection, the VI algorithm is formulated for the LQR

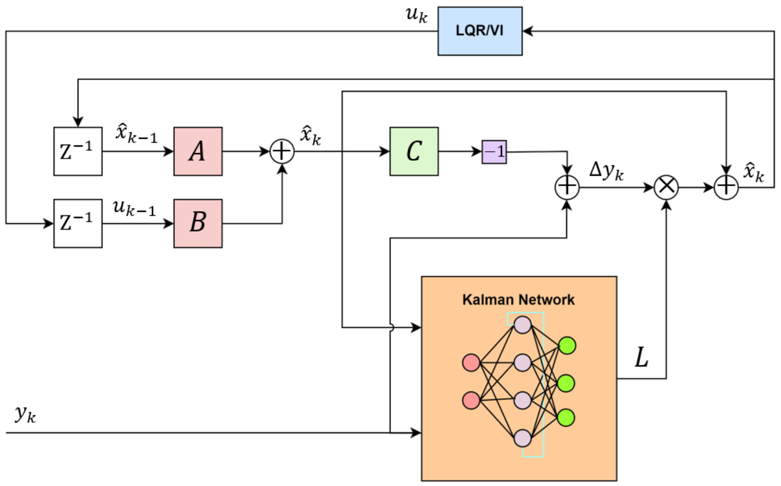

FIGURE 3: KalmanNet block diagram

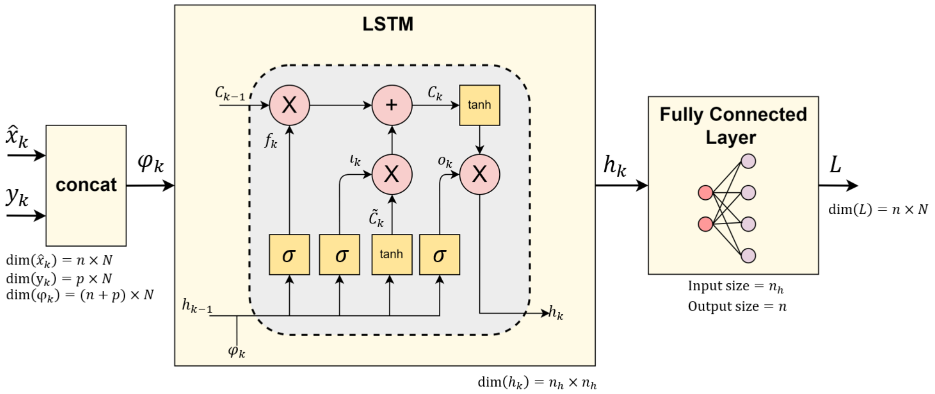

FIGURE 4: KalmanNet-LSTM architecture

problem. The performance index or value function for LQR problem is formulated as in Eq. (30).

$$V(x_k) = \frac{1}{2}\left(x_k^TQx_k + u_k^TRu_k + \sum_{i=k+1}^{\infty}\left[x_i^TQx_i + u_i^TRu_i\right]\right) \tag{30}$$

which also generates the LQR Bellman equation in Eq. (31)

$$V(x_k) = \frac{1}{2}\left(x_k^TQx_k + u_k^TRu_k\right) + V(x_{k+1}) \tag{31}$$

With the assumption that the performance index value along the x_k trajectory is quadratic in order that the performance index can be expressed as Eq. (32) through the Kernel matrix P.

$$V(x_k) = \frac{1}{2}x_k^TPx_k \tag{32}$$

The substitution of Eq. (31) and (32) for Eq. (33) occurs to form the Bellman Equation on the LQR problem. Assuming a constant state feedback control signal $u_k = -K\hat{x}_k$ for some stabilizing gain [14].

$$x_k^TPx_k = x_k^TQx_k + u_k^TRu_k + x_k^T(A-BK)^TP(A-BK)x_k \tag{33}$$

It then proceeds to substitute the Bellman equation with DT LQR, which is called the Lyapunnov Equation [14] in Eq. (32) in which the performance index V_k is dependent on the estimated current state $\hat{x}_k$ and control inputs u_k.

$$(A-BK)^TP(A-BK) - P + Q + K^TRK = 0 \tag{34}$$

The Hamilton function of the DT LQR system is formulated in Eq. (35).

$$\begin{aligned} H(x_k, u_k) =& x_k^TQx_k + u_k^TRu_k \\ &+ (Ax_k + Bu_k)^TP(Ax_k + Bu_k) - x_k^TPx_k \end{aligned} \tag{35}$$

The first derivative of Hamiltonian function leading to the necessary condition of optimality is represented in Eq. (36).

$$\frac{\partial H(x_k, u_k)}{\partial u_k} = (R + B^TPB)u_k + B^TPAx_k = 0 \tag{36}$$

From the Eq. (36), we could compute the control signal such as Eq. (37) is computed by inserting it to Eq. (33) to generate the Eq. (38), which is simply called DT ARE.

$$u_k = -(R + B^TPB)^{-1}B^TPAx_k \tag{37}$$

Algorithm 1 LSTM for KalmanNet

```
Initialization: Some parameters such as :
  (1) Number of epoch
  (2) Parameter of LSTM
  (3) Define the dataset ($\varphi$)
  $\varphi = \begin{bmatrix}\hat{x} & y\end{bmatrix}$, where $\hat{x}$ and $y$ are estimated state and measurement data, respectively.
  (4) $x$ : true hidden state vectors
  (5) $\psi$ : the threshold of MSE value
Output: $L$ : Kalman gain
  Pre-processing data : Split the dataset into training and testing data
  while iteration $\leq$ number of epoch do
      Compute the $h_k$ using Eq.(29)
      Compute the Kalman gain ($L$) by reconstruct the output layer $h_k$ using fully connected layer into the dimension of Kalman gain [19]
  end while
  Test the KalmanNet model using testing dataset
  Compute MSE value
  if MSE value$\leq \psi$ then
      Re-train
  else
      Stop
  end if
```

$$A^T PA - P + Q - A^T PB(B^T PB + R)^{-1} B^T PA = 0 \quad (38)$$

From Eq. (34), it can be concluded that it represents the Bellman optimality equation. It is possible to adopt this equation in implementing the VI algorithm. The VI Algorithm format is simply a Lyapunov recursion that converges to the solution of the Riccati equation [14]. In this study, the offline VI algorithm is employed to solve the LQR problem. For that reason, complete knowledge of the system dynamics (A, B) is highly necessary.

C. IMPLEMENTATION OF DISCOUNT FACTOR

The problem that occurs when implementing the VI algorithm is how to generate a stabilizing control policy [23]. From [15] and [14], it is concluded that a discount factor influences the stability. γ is a discount factor with the value range of $\gamma \in (0, 1)$ which provides the weight of the performance index. The effect of the discount factor is to provide weight to the performance index with a constant that is time-varying decaying [24]. Based on the Bellman Equation for the infinite horizon, discounted LQR problem can be formulated as Eq. (39).

$$V(x_k) = \frac{1}{2}\Bigg(x_k^\top Q x_k + u_k^\top R u_k + \sum_{i=k+1}^{\infty} \gamma^{i-k} \Big(x_i^\top Q x_i + u_i^\top R u_i\Big)\Bigg) \quad (39)$$

Eq. (39) which generates the LQR Bellman equation in Eq. (40).

$$V(x_k) = \frac{1}{2}\Bigg(x_k^\top Q x_k + u_k^\top R u_k + \gamma V\Big(x_{k+1}\Big)\Bigg) \quad (40)$$

Proceed to substitute Eq. 32 with the value function represented in Eq. (40) to obtain Eq. (41).

$$x_k^\top P x_k = x_k^\top Q x_k + u_k^\top R u_k + \gamma x_{k+1}^\top P x_{k+1} \quad (41)$$

The initial idea of Theorem 1 is based on the research [14]], with an addition of modification. The system dynamics we use in this study are $\bar{A}$ and $\bar{B}$, as they function to solve regulatory problems. Whereas in [14], the system dynamics used are $\bar{T}$ and $\bar{B}_1$, the augmentation matrices of state and reference.

Theorem 1: The ARE solution of the VI algorithm could be formulated in Eq. (42).

$$Q - P + \bar{A}^\top P \bar{A} - \bar{A}^\top P \bar{B}\Big(R + \bar{B}^\top P \bar{B}\Big)^{-1} \bar{B}^\top P \bar{A} = 0 \quad (42)$$

The assumptions used in Theorem 1 are as follows:
(A, B) can be stabilized (stabilizable) then $(\bar{A}, \bar{B})$, can also be stabilized (stabilizable) where $\bar{A} = \gamma^{1/2} A$ and $\bar{B} = \gamma^{1/2} B$.
The Hamiltonian function for the discounted linear regulator problem and using the assumption in Theorem 1 would be formulated in Eq. (43).

$$H(x_k, u_k) = x_k^\top Q x_k + u_k^\top R u_k + \gamma\Big(\bar{A}x_k + \bar{B}u_k\Big)^\top P \Big(\bar{A}x_k + \bar{B}u_k\Big) - x_k^\top P x_k = 0 \quad (43)$$

The first derivative results from Eq. (43) is formulated in Eq. (44).

$$\frac{\partial H(x_k, u_k)}{\partial u_k} = \Big(R + \gamma \bar{B}^\top P \bar{B}\Big) u_k + \gamma \bar{B}^\top P \bar{A} x_k = 0 \quad (44)$$

The calculation of the control signal u_k is based on the Eq. (44) could be obtained in Eq. (45)

$$u_k = -K x_k \quad (45)$$

where K is formulated in Eq. (46)

$$K = \big(R + \gamma \bar{B}^\top P \bar{B}\big)^{-1} \gamma \bar{B}^\top P \bar{A} \quad (46)$$

From [15] [14], we use the iterative algorithms to solve the discounted LQR problem (see **Algorithm 2**).

The control problem that we examine in this study is a problem with finite time or finite horizon. Therefore, we will not discuss the dynamical characteristics of the system at times outside the finite horizon we define. Before we state the stability definition, recall that we use the notation of $\Phi(k, 0)$ to indicate the evolution operator of Eq. (22). Therefore we adapt Willem's definition of stability [25] to

Algorithm 2 VI for Discounted LQR Solution

```
Initialization: Start with a control policy $K = 0$
while $0 < j < N$ do
    1. Policy Evaluation, solve the computation of $P_{j+1}$ using Eq. (47)
        $$P_{j+1} = Q + K_j^\top R K_j + \gamma(\bar{A} - \bar{B}K_j)^\top P_j (\bar{A} - \bar{B}K_j) \quad (47)$$
    2. Policy Improvement
        $$K_{j+1} = \left(R + \gamma \bar{B}^\top P_{j+1} \bar{B}\right)^{-1} \gamma \bar{B}^\top P_{j+1} \bar{A} \quad (48)$$
end while
```

a similar definition of stability, but in the context of a finite time horizon.

Definition 1: The system Eq. (22) with $\mu \equiv 0$ is called: stable in a finite horizon $k = 0, 1, \ldots, (N-1)$, if there exists a bound $c > 0$ (which may depend on k_0) such that $||\Phi(k, 0)|| \leq c$ holds for all $k = 0, 1, \ldots, (N-1)$.

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

Proposition 2: The system (22) with $\mu \equiv 0$ is stable in a finite horizon $k = 0, 1, \ldots, (N-1)$, if and only if $||\Lambda_k||$ is bounded for all $k = 0, 1, \ldots, (N-1)$ [26].

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

Furthermore, by observing the design process that we carried out, both KF and KalmanNet, LQR and VI, it is impossible to produce unbounded gain filter or gain regulator results, then by using the necessary and sufficient condition given in Proposition 2, we can conclude that all the closed-loop system will be stable in the defined finite horizon, namely $k = 0, 1, \ldots, (N-1)$, in the sense according to Definition 1.

IV. SIMULATION STUDY

This simulation study comprises two linear system case studies namely cart-pole system and batch distillation column. All these simulations are uploaded in [1] using MATLAB. The parameters used in this study are as follow :

- w_k and v_k used in this simulation are uncorrelated white noise Gaussian
- Covariance matrices R_{ww} and R_{vv} respectively equal to 0.01 and 0.1, respectively
- Weight matrices Q and R respectively equal to 0.01 and 0.1, respectively.
- Time index (N) = 100
- The proposed KalmanNet solution uses the LSTM architecture with the ADAM optimizer, 3 hidden units, and the number of mini-batch sizes used is 8
- γ = 0.01.
- Initial condition $x(0) = 0.1$

1) Cart-Pole System

One of the classic control problems was 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 chosen deliberately simple to demonstrate the aims of this research. The dynamics of the cart-pole system represented in Eq. (49) with the parameter value was summarized in Table 2 from the technical detail in [2]. The plant dynamics would be formulated in Eq. (49) where 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.

TABLE 2: 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

$$x_{k+1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m_p}{m_c} g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m_p + m_c}{l m_c} g & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ \frac{1}{m_c} \\ 0 \\ \frac{1}{l m_c} \end{bmatrix} u + w_k$$
$$y_k = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} x_k + v_k \quad (49)$$

Our proposed algorithm for the first case study was already published on [27].

2) Batch Distillation Column

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) [28]. 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

[1] https://github.com/adinovitarini/Adaptation_LQG_method_by_data

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

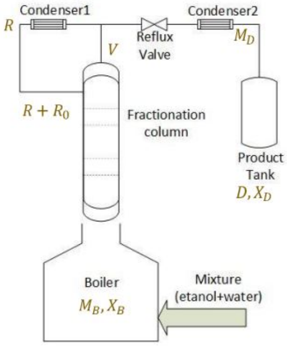

FIGURE 5: Batch distillation column schematic diagram

(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 [21].The state, input, and output matrices is define in Eq.(50).

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{50}$$

The state, input, and output matrices denoted as A, B, and C are defined in Eq. (51).

$$\begin{aligned} A &= \begin{bmatrix} 1.14 & -0.78 & -0.41 & -0.93 & 0 \\ 1.05 & 1.02 & 0.52 & 0.55 & 0 \\ -0.77 & 0.74 & -0.83 & 2.68e-03 & 0 \\ 1.18 & 0.95 & -0.65 & -0.79 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ B &= \begin{bmatrix} -1.37 \\ 0.45 \\ 1.08 \\ -0.38 \\ 1 \end{bmatrix} \\ C &= \begin{bmatrix} 0.73 & -0.78 & 0.97 & -0.34 & 1 \end{bmatrix} \end{aligned} \tag{51}$$

In Section IV.A, we compare the performance of KF and KalmanNet as state estimation method. Section IV.B will be contain the comparison of the convergence of four scenarios that have been done to convince our proposed algorithm.

A. KF VS KALMANNET

Based on the comparison between KalmanNet and KF as shown in Table 3, we find that the MSE value of the estimated state $\hat{x}$ with the original state x generated by KalmanNet is much smaller than KF. In addition, in this study, the control signal u_k obtained from LQR produces a smaller MSE when compared to the VI Algorithm. Therefore, the KalmanNet model used in this study uses control signals from LQR. In addition, there is no need to process equation parameter in KalmanNet as it is required in the KF method. This results has some implications for the use of KalmanNet method, which is more efficient than KF as the state estimation method.

TABLE 3: Performance Comparison of KalmanNet and KF

Control methods	Methods	Case Study	
		1	2
LQR	KalmanNet	1.65E-05	1.07E-04
	KF	1.0409	1.2625
VI	KalmanNet	6.72E-05	1.87E-04
	KF	3.3787	1.5504

TABLE 4: Performance comparison of four scenarios testing

Performance	Methods	Case Study	
		1	2
$\|\|u\|\|$	KF-VI	2.48E-01	1.14E-02
	KN-LQR	8.00E-03	3.40E-03
	KN-VI	1.38E-02	1.83E-02
	LQG	1.60E-02	8.80E-03
J	KF-VI	4.90E-02	1.59E-01
	KN-LQR	2.85E-02	3.55E-02
	KN-VI	2.94E-02	3.55E-02
	LQG	5.86E-02	1.46E-01
CT (time-step)	KF-VI	Still oscillate	Still oscillate
	KN-LQR	8	6
	KN-VI	10	8
	LQG	Still oscillate	Still oscillate
Time Elapsed (sec)	KF-VI	1.53E-02	1.71E-02
	KN-LQR	1.50E-03	1.50E-03
	KN-VI	2.30E-03	2.20E-03
	LQG	1.55E-01	1.47E-01

B. COMPARISON FOUR SCENARIOS

This section examines four scenarios and review the control signals' norm values, the performance indices, Convergence Time (CT) in time-step domain, and Time elapsed in second domain which summarized in Table 4. The use for the second scenario has a control signal norm value as well as a performance index, faster CT, and faster computation time. The control signal (K), obtained using VI, requires a longer computation time when compared to the LQR solution. This is because the initialization of the control signal parameters in the algorithm is distributed randomly. However, it is important to note that in this method, the LQR solutions performed contain non causality.

If we look at the time elapsed, representing computation time, the second scenario has the fastest computation time. However, based on testing we also found that using the VI algorithm also requires computation time that is not too far from the second scenario. The CT test results can be seen in Table 4 found from the control signal trajectory in Fig. 6 and 7. In the test results, we found that the use of the first and fourth scenarios has not been able to make a convergent control signal trajectory. Therefore, as a trade-off, the use of the third method, the KalmanNet-VI, is adequately efficient to adapt the LQG controller scheme.

V. CONCLUSION

Our proposed algorithm empirically shows that it can solve regulatory problems in discrete-time linear systems affected by uncorrelated Gaussian white noise. The results of the tests

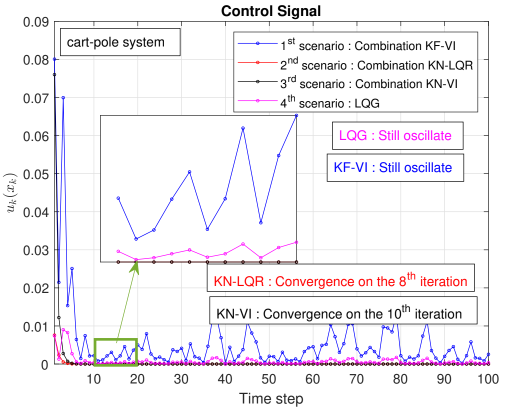

FIGURE 6: Control signal plot for 1^{st} case study

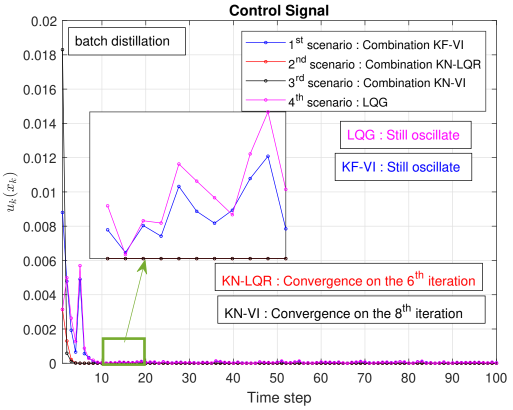

FIGURE 7: Control signal plot for 2^{nd} case study

that have been carried out show that the use of conventional KF has not been able to produce a trajectory of the control signal u_k that converges to a zero value during a certain time-horizon. Meanwhile, the use of KalmanNet is able to produce a trajectory of the control signal u_k that converges to a zero value. This is because both KF and KalmanNet are used to generate the estimated state $\hat{x}_k$ which is used to build the control signal u_k. Meanwhile, the use of the VI algorithm can solve regulatory problems. However, it can make the convergence of the control signal evolution longer than conventional LQR solutions. The VI algorithm has advantageous because the ARE solution is done iteratively. Thus, it does not require backward-in-time calculations like the traditional solution of LQR. In this research, the first and second case study's control signal was able to converge to a value of zero at the 8-th and 6-th time-steps when implementing the 2-nd scheme, respectively. Nevertheless, first and second case study's control signal was able to converge in the 10-th and 8-th time-steps when we implemented the 3-rd scenario, respectively. Future research is to use a combination of these methods in different case studies. Case studies can be in robotics systems or complex systems in industrial processes.

REFERENCES

[1] F. Lewis, D. Vrabie, and V. Syrmos, Optimal Control, ser. EngineeringPro collection. Wiley, 2012. [Online]. Available: https://books.google.co.id/books?id=NFEYFmllK9QC

[2] R. Stengel, Optimal Control and Estimation, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: https://books.google.co.id/books?id=JqDDAgAAQBAJ

[3] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," Information Sciences, vol. 595, pp. 364–377, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522002031

[4] G. R. G. da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-driven lqr control design," IEEE control systems letters, vol. 3, no. 1, pp. 180–185, 2018.

[5] H. Mohammadi, M. Soltanolkotabi, and M. R. Jovanović, "On the lack of gradient domination for linear quadratic gaussian problems with incomplete state information," in 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, 2021, pp. 1120–1124.

[6] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," IEEE Transactions on Automatic Control, vol. 65, no. 11, pp. 4753–4768, 2020.

[7] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," IEEE Transactions on Automatic Control, vol. 65, no. 3, pp. 909–924, 2019.

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

[9] S. Skogestad, Multivariable Feedback Control: Analysis and Design. Wiley India, 2014. [Online]. Available: https://books.google.co.id/books?id=CF3gvQEACAAJ

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

[11] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. The MIT Press, 2018.

[12] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ-policy iteration for discrete-time linear quadratic regulation," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[13] 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, 2018.

[14] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," IEEE Transactions on Cybernetics, vol. 45, no. 12, pp. 2770–2779, 2015.

[15] 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, 2014.

[16] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming for stochastic systems with state and control dependent noise," IEEE Transactions on Automatic control, vol. 61, no. 12, pp. 4170–4175, 2016.

[17] F. A. Yaghmaie and F. Gustafsson, "Using reinforcement learning for model-free linear quadratic control with process and measurement noises," in 2019 IEEE 58th Conference on Decision and Control (CDC). IEEE, 2019, pp. 6510–6517.

[18] S. G. Casspi, O. Husser, G. Revach, and N. Shlezinger, "Lqgnet: Hybrid model-based and data-driven linear quadratic stochastic control," 2022. [Online]. Available: https://arxiv.org/abs/2210.12803

[19] G. Revach, N. Shlezinger, R. J. G. van Sloun, and Y. C. Eldar, "Kalmannet: Data-driven kalman filtering," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3905–3909.

[20] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 1, pp. 14–25, 2011.

[21] A. N. Putri, C. Machbub, and E. Hidayat, "Combination of elman neural network and kalman network for modeling of batch distillation process," in 2022 13th Asian Control Conference (ASCC), 2022, pp. 926–931.

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

[23] M. Ha, D. Wang, and D. Liu, “Generalized value iteration for discounted optimal control with stability analysis,” Systems & Control Letters, vol. 147, p. 104847, 2021.

[24] M. Granzotto, R. Postoyan, L. Buşoniu, D. Nešić, and J. Daafouz, “Finite-horizon discounted optimal control: stability and performance,” IEEE Transactions on Automatic Control, vol. 66, no. 2, pp. 550–565, 2020.

[25] J. L. Willems, Stability Theory of Dynamical Systems. Wiley Interscience Division, 1970.

[26] A. N. Vargas, J. B. R. do Val, and E. F. Costa, “On stability of linear time-varying stochastic discrete-time systems,” in 2007 European Control Conference (ECC), 2007, pp. 2423–2427.

[27] A. N. Putri, C. Machbub, D. Mahayana, and E. Hidayat, “Linear quadratic gaussian using kalman network and reinforcement learning for discrete-time system,” in 2022 12th International Conference on System Engineering and Technology (ICSET), 2022, pp. 54–60.

[28] A. S. Rohman, P. H. Rusmin, R. Maulidda, E. Hidayat, C. Machbub, and D. Mahayana, “Modelling of the mini batch distillation column,” International Journal on Electrical Engineering and Informatics, vol. 10, no. 2, pp. 350–368, 2018.

ADI NOVITARINI PUTRI received her bachelor degree in electrical engineering with majoring in Control System Engineering from Institut Teknologi Sepuluh Nopember (ITS) in 2019 and master degree in electrical engineering majoring in Control Engineering with cum-laude predicate from Institut Teknologi Bandung (ITB) in 2021. She is now PhD student in Control and Computer Systems Research Group, School of Electrical Engineering and Informatics, ITB. Her current research interests are in data-driven control and reinforcement learning.

CARMADI MACHBUB received his bachelor degree in electrical engineering from Institut Teknologi Bandung (ITB) in 1980, DEA in Control Engineering and Industrial Informatics in 1988, and Doctoral degree in Engineering Sciences majoring in Control Engineering and Industrial Informatics from Université de Nantes/Ecole Centrale de Nantes in 1991. He is now professor and Head of Control and Computer Systems Research Division, School of Electrical Engineering and Informatics, ITB. His current research interests are in control, machine perception, and intelligent systems.

DIMITRI MAHAYANA graduated from Bandung Institute of Technology in 1989 with a bachelor’s degree in electrical engineering with cum-laude/honor predicate. In 1994, he finished his Master of Engineering in Electrical Engineering as well in Waseda University (Tokyo, Japan), with straight-A mark. In 1998, he got his doctoral degree in ITB with cum-laude predicate. Some of his research interests are nonlinear dynamical system, time varying system, control theory and convergence between control engineering and data science. He is currently a lecturer in School of Electrical Engineering and Informatics, Bandung Institute of Technology (ITB).

EGI HIDAYAT received his bachelor degree in electrical engineering from Institut Teknologi Bandung (ITB), Master of Science in Control and Information System from Universitat Duisburg-Essen, and Ph.D. degree in Electrical Engineering from Uppsala University. Currently, he is a lecturer at the School of Electrical Engineering and Informatics, ITB. His research interests include modeling & system identification, control & learning, and robotics.