IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 March 2024, accepted 5 April 2024, date of publication 11 April 2024, date of current version 15 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387460

APPLIED RESEARCH

Design of Electric Power Steering System Identification and Control for Autonomous Vehicles Based on Artificial Neural Network

RODI HARTONO[1,3,*], HYUN ROK CHA[2,*], (Member, IEEE), AND KYOO JAE SHIN[1], (Member, IEEE)

[1]Artificial Intelligence Convergence Department, Busan University of Foreign Studies, Busan 46234, Republic of Korea
[2]Automotive Materials and Components Research and Development Group, Korea Institute of Industrial Technology, Gwangju 61012, Republic of Korea
[3]Engineering and Computer Science Department, Universitas Komputer Indonesia, Bandung 40132, Indonesia

Corresponding author: Kyoo Jae Shin (kyoojae@bufs.ac.kr)

This work was supported by Korea Evaluation Institute of Industrial Technology as ''Development of an Industrial Skateboard With Longitudinal and Transverse Expansion Operations'' under Grant KEIT 20022022.

*Hyun Rok Cha and Rodi Hartono contributed equally to this work.

ABSTRACT Electric power steering (EPS) poses significant control challenges in autonomous vehicles due to their inherent complexity and non-linearity. This study explores the application of artificial neural network (ANN) to address these limitations. Two approaches are proposed: 1) an ANN-based identifier utilizing the backpropagation (BP) algorithm to learn the system's non-linear dynamics, and 2) an ANN-based controller leveraging the Levenberg-Marquardt (LM) algorithm to improve control performance. Our findings demonstrate the efficacy of the proposed ANN-based BP algorithm in EPS system identification achieving over 99.6% accuracy in predicting EPS system dynamics compared to the traditional approach. Additionally, the LM-learned ANN-based controller aiming a faster response and precise reference tracking compared to the traditional controller method. These advancements underscore the potential of employing ANN methodologies to optimize EPS performance in autonomous vehicles.

INDEX TERMS Artificial neural network, autonomous vehicles, backpropagation, electric power steering, Levenberg-Marquardt, proportional integral derivative controller, system identification, transfer function estimator.

I. INTRODUCTION

Electric Power Steering (EPS) is a vehicle steering system that employs an electric motor to provide torque assistance for steering. EPS systems are increasingly common in both traditional power-assisted steering (PAS) vehicles, which enhances the driver's steering effort, and in autonomous steering control units for self-driving cars. In PAS-equipped vehicles, the system interprets the driver's steering inputs to adjust torque and instantaneously align with the desired steering angle. Furthermore, in autonomous vehicles, EPS systems autonomously control steering by processing data from various sensors i.e., cameras, radar, and lidar [1].

There are two primary steering control methods: hydraulic steering control and electrical steering control. EPS operates through electrical steering control, and it offers superior reliability, safety, efficiency, reduced emissions, and ease of maintenance when compared to hydraulic steering control [2]. The electrical motor in EPS systems typically employs a brushless direct current motor (BLDCM) or

The associate editor coordinating the review of this manuscript and approving it for publication was Engang Tian.

permanent magnet synchronous motor (PMSM) as the actuator, which transmits power to the vehicle wheels through a gearbox and mechanical steering rack [2], [3], [4].

Maintaining optimal performance and stability of the vehicle's EPS requires applying precise control inputs to the system [5], [6], [7]. This control can be implemented using a variety of methods, such as proportional integral derivative (PID) control. This is a classical control method that has been used for many years. PID controllers are simple to understand and implement, and they are effective in controlling a wide variety of systems. PID controller is typically a linear controller, which means it operates effectively within a linear control system [8], [9], [10], [11]. Due to the nonlinearity of the EPS system, researchers have tried using machine learning and AI to understand and predict this complicated, non-linear behavior. One of these methods is using an artificial neural network (ANN). ANN is a powerful machine learning technique that can be used for a variety of tasks, including control and identification of EPS systems. ANN is inspired by the structure and function of the human brain, and they consist of a network of interconnected nodes, called neurons. Each neuron is able to process information and send signals to other neurons in the network. ANN is an adaptive and data-driven technology that can effectively determine relationships between input elements and process outputs. Activation functions in ANN empower them to capture both linear and nonlinear patterns in data, without requiring any prior assumptions. [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31].

Unlocking the full potential of ANN as an identifier and a controller, this study proposes two lines of research. The first study is to design an EPS identifier by using ANN based backpropagation (BP) algorithm to analyze the effectiveness of ANN based BP algorithm in EPS system identification and the second study is to use ANN based LM algorithmto improve the EPS system controller for autonomous vehicles.

This paper is organized as follows. Section II discussed the analysis of the original EPS system and the concept of ANN-based BP and LM. Section III discussed the identification of the EPS system using an ANN-based BP algorithm. Section IV presents the proposed ANN-based LM algorithm for EPS control. Section V presents the experimental results and discussion of system identification and EPS control using ANN. Section VI presents the conclusion of this research.

II. ANALYSIS OF THE ORIGINAL EPS SYSTEM

A. CONFIGURATION OF EPS

EPS systems are crucial for improving the steering control of autonomous vehicles. In contrast to conventional hydraulic power steering systems, EPS systems in autonomous vehicles employ electric motors to deliver accurate steering control. The main components of an EPS system in an autonomous vehicle are:

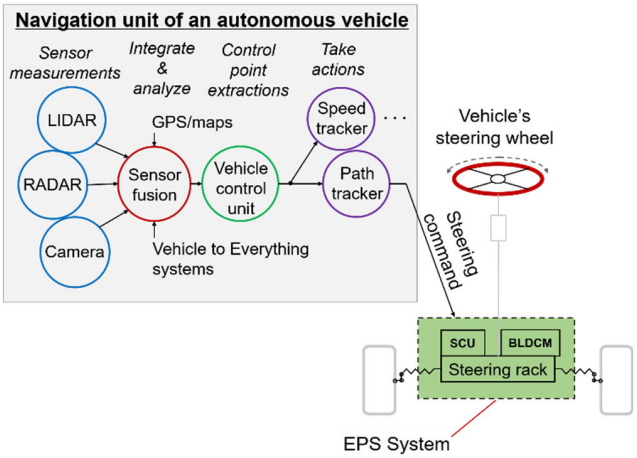

FIGURE 1. EPS system control and mechanism in autonomous vehicles.

- Electric motor: At the core of EPS systems is the electric motor, responsible for generating torque that facilitates steering control.
- Steering control unit (SCU): The steering control unit serves as the brain of the EPS system. It receives the angle command from the navigation unit. The navigation unit measures the vehicle's speed, acceleration, and heading. This information is used by the vehicle control unit to calculate required steering actions and send them to SCU.
- Steering rack: The steering is responsible for converting the torque from the electric motor to the vehicle's wheels.

Refers to Fig. 1, the vehicle control unit integrates and analyzes the sensor data, enabling the system to comprehend the vehicle's surroundings. Based on the analyzed data, the computational unit takes actions or determines the optimal steering adjustments required to navigate the vehicle. The vehicle control unit generates precise steering commands, which are transmitted to the SCU to calculate the required torque for the electric motor to execute the necessary steering actions.

For this study, a steering rack of a sedan car was used. We can see the EPS mechanical actuator in Fig. 2. This steering rack is a mechanical rack and pinion gear type that can be lengthened and shortened by 85mm. This gives a steering wheel a working angle of +46 to −46 degrees from its centerline. A positive sign indicates the steering direction in a clockwise direction and vice versa.

This mechanical rack and pinion are connected to a driven gear that has a ratio of 2.3:1 with the driving gear and a ratio of 26:1 between the final vehicle's wheel with the driving gear. The driving gear and the driven gear are connected by an HTD belt. Table 1 shows the details of vehicle steering rack information used in this research.

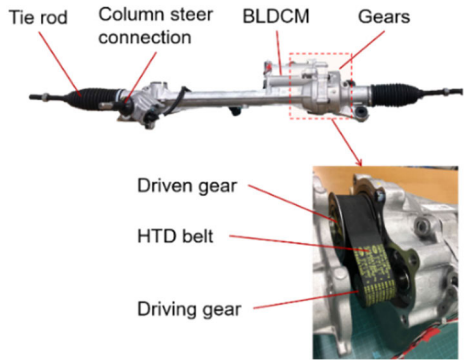

FIGURE 2. The EPS system actuator used in this study.

TABLE 1. Vehicle steering rack information.

Component	Remark
Motor type and power	BLDCM, 600W
Motor poles count	8 poles
Motor max. speed	2,000 RPM
Steering wheel angle maximum	±46°
Driving gear to the final wheel ratio	26:1
Motor gear type	Helical
Drive motor to steering-rack gear connection	Pulley

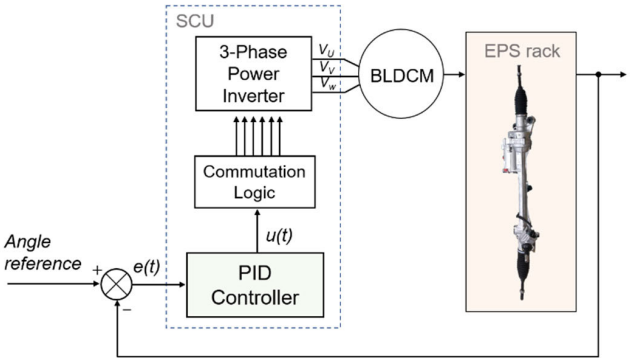

FIGURE 3. EPS scheme of SCU with PID controller.

B. REPRESENTATION OF EPS BEHAVIOUR WITH PID CONTROL IMPLEMENTATION FOR ANN DATASET

To develop an accurate and robust ANN controller for an EPS system control, a comprehensive dataset of input-output pairs is first gathered. The dataset can be gathered from a simulation or observation. For the EPS ANN controller dataset in this research, we used an observation method using a PID controller that makes the EPS follow the angle reference with fast response time and minimum steady-state error. In Fig. 3 we can see the EPS PID controller scheme to provide a controlled EPS.

In Fig. 3, the variable *e(t)* represents the error signal of the difference between the angle reference and the actual measured position of the EPS as read by the angular position sensor at time t. Once the error *e(t)* signal is acquired, the PID controller calculates its derivative and integral concerning time. These calculations allow the controller to track how the

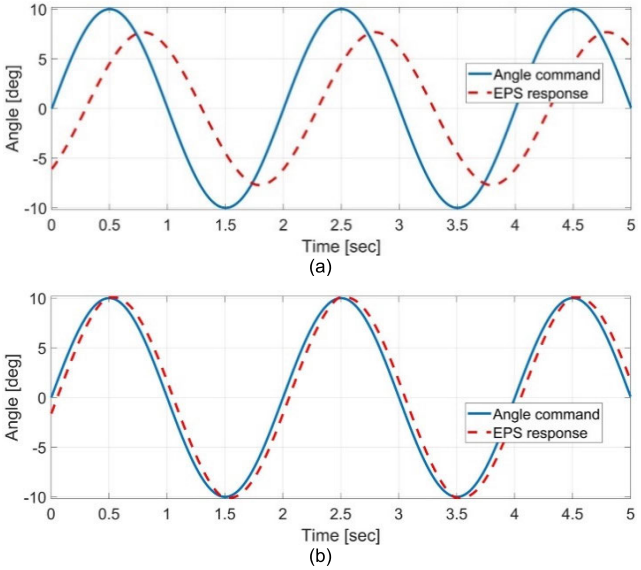

FIGURE 4. Output responses by PID parameter tuning: (a) EPS response with a lower constant of $K_1 = 0.04$, $K_{2=}$ 0.0095, and $K_3 = 0.0000025$, (b) EPS response with optimal constants of $K_{1=}$ 0.113, $K_2 = 2.016$, and $K_3 = 0.0000025$.

angular position error signal is changing over time and how to use this information to adjust the control signal *u(t)* to the EPS movement. The PID controller produces an output that is determined in the time domain based on the feedback error using the following equation:

$$u(t) = K_1 e(t) + K_2 \int e(t)dt + K_3 \frac{de}{dt}, \tag{1}$$

where K_1 is the proportional gain, K_2 is the integral gain and K_3 is the derivative gain.

Testing of the angular position control system is carried out by first determining the parameter values K_1, K_2, and K_3 which are in accordance with the desired system response design. Fig. 4 shows the EPS response results in two sets different of parameter value sets of K_1, K_2, and K_3. The input angle reference is a sine wave with a frequency of 0.5 Hz and amplitude of $\pm 10°$.

To facilitate the EPS ANN system identification and control datasets, we execute the chosen commutation code and its associated sequence algorithm on a microcontroller. This microcontroller is linked to an upper computer through serial communication, enabling the recording of EPS responses based on input signals. MATLAB/Simulink, integrated with the NXP model-based design toolbox (MBDT), is utilized for code generation. Motor speed control is achieved through a PWM technique, with a maximum input voltage of 12VDC provided to rotate the motor, selected to meet the vehicle power environment. The actual motor angle position, serving as feedback, is measured using a magnetic position sensor (AS5247U) through the ABI interface. The experimental setup is described in Fig. 5.

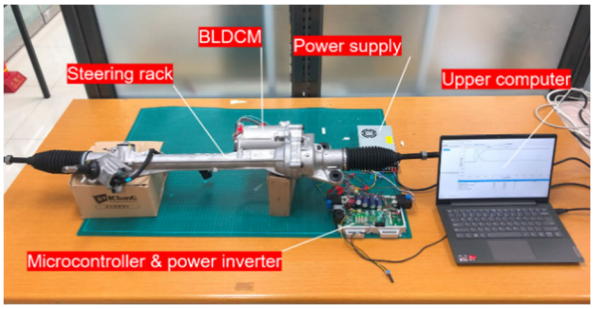

FIGURE 5. Experimental setup.

In Fig. 4(b), the EPS system accurately follows the specified tracking line. Conversely, Fig. 4(a) illustrates the subpar performance of the EPS system, characterized by slow response times and significant steady-state errors. By conducting a cross-correlation analysis between the two sine waves depicted in Fig. 4(b), we determine a phase difference and time delay of 0.0425 seconds. For this research, we choose the parameter configuration depicted in Fig. 4(b) as the dataset for designing the ANN angular controller.

C. ANN BASED BP AND LM CONCEPT

ANN is a fundamental component of machine learning algorithms designed to mimic the functioning of the human brain. ANN learns from data by iteratively adjusting their parameters (weights and biases) based on observed examples to minimize the discrepancy between their predictions and the actual target values. This learning process, often referred to as training, involves feeding input data through the network, computing predictions, comparing them to the actual targets, and updating the network's parameters accordingly. Through this iterative process, ANN gradually improves their ability to make accurate predictions on unseen data.

When designing ANN, it's essential to consider various factors such as the network architecture, activation functions, and optimization algorithms. The network architecture determines how input data is transformed through interconnected layers of neurons to produce output predictions. Activation functions introduce non-linearities to the network, enabling it to capture complex patterns and relationships within the data. Optimization algorithms, such as the BP and LM algorithm play a crucial role in adjusting the network's parameters during the training process to minimize prediction errors effectively.

In this research, we were using ANN-based BP to design EPS identifiers and ANN-based LM to improve the EPS system controller. The ANN-based BP is a multi-layered ANN model. The primary principle involves forward propagating the results, which generates an error that is then minimized. To achieve optimal performance, the network relies on Equation (2) to propagate errors backward, meticulously adjusting its internal weights for enhanced predictions

$$W_{a+1} = W_a - \eta \frac{\partial E}{\partial W_a}. \tag{2}$$

Equation (2) defines the weight update rule for the ANN during training. It uses the difference between predictions and true values across whole data points to adjust the current weight W_a by a factor of the learning rate η. This iterative process, known as BP, guides the ANN towards a model that minimizes the mean absolute error (MAE), fulfilling the desired accuracy requirements.

The LM algorithm is an optimization method commonly used in training ANN. It combines aspects of both gradient descent and Gauss-Newton methods to efficiently find the minimum of a cost function, which represents the error between the network's predictions and the actual target values.

Mathematically, the LM algorithm involves updating the parameters of the ANN, denoted as δ. The update rule can be expressed as:

$$\delta_{k+1} = \delta_k - \left(J^T J + \lambda I\right)^{-1} J^T e, \tag{3}$$

where δ_k represents the parameters of the ANN at iteration k, J is the Jacobian matrix, representing the first-order derivatives of the network's outputs with respect to its parameters, e is the error vector, representing the difference between the network's predictions and the actual target values, and λ represents the damping factor, controlling the step size of the update to adjusts dynamically based on the progress of the optimization process.

III. IDENTIFICATION OF EPS SYSTEM USING ANN-BASED BP ALGORITHM

In the domain of system identification, an ANN is employed to approximate the inherent dynamics or connection between input and output variables within a system, utilizing observed data. It acts as a versatile and adaptable model capable of capturing non-linearities and complex interactions within the system's behavior. The input variables are fed into the neurons of the ANN, where they endure processing across multiple interconnected layers. Each neuron employs an activation function to transform its input and transmit the outcome to the subsequent layer. Throughout the training phase, the network's parameters, such as the weights and biases of the neurons, are fine-tuned to enhance performance and reduce the disparity between predicted and observed outputs.

The learning process involves iteratively adjusting the weights of the network's connections. An iteration is specified as a comprehensive round of calculations, including both forward and backward passes. The goal is to minimize the error criterion $e_k(z)$ by reducing a cost function. When all the data, presented collectively with a size of N is processed, the final weight updates occur. The normalized sum of squared error for the networks, with respect to the size

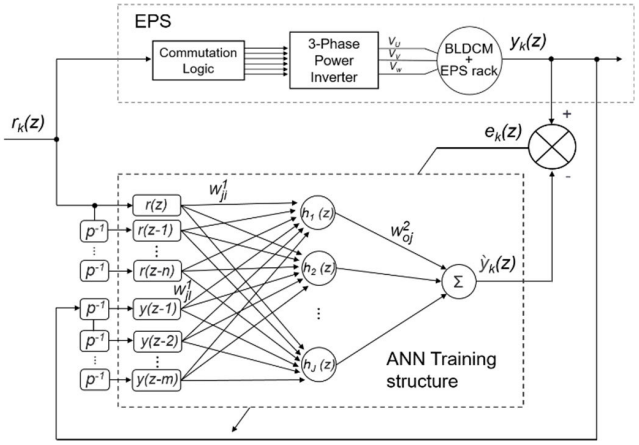

FIGURE 6. Proposed ANN structure for EPS identification.

N, is then expressed as part of this iterative learning process. We thus have

$$E_k(z) = \frac{1}{N}\sum_{k=1}^{N} e_k^2(z). \tag{4}$$

The proposed architecture of the ANN for system identification in this study is shown in Fig. 6. The network's output is a function of both past inputs and outputs. In Fig. 6, multiple inputs, denoted as $r(z)$, $r(z-1)$, $\ldots$, $r(z-n)$ and $y(z-1)$, $y(z-2)$, $\ldots$, $y(z-m)$ are depicted. Here $r(z)$, $r(z-1)$, $\ldots$, $r(z-n)$ serves as the primary input for the main neurons, and $y(z-1)$, $y(z-2)$, $\ldots$, $y(z-m)$ represents inputs from the feedback output plant. The desired output for the ANN identifier output, denoted as $y_k(z)$ is expressed as $\hat{y}_k(z)$. In the notation W_{ji}^1, W_{jl}^1 and W_{oj}^2,the indices i, j, and o represent different weights within the network. The signal flow within the network progresses from left to right. In particular, neuron j corresponds to a neuron within the hidden layer, whereas neuron o represents the singular neuron in the output layer.

The feedforward process in the network initiates by multiplying the input of each neuron i and l with their respective weights, denoted as W_{ji}^1 and W_{jl}^1. Subsequently, the products are transmitted to each neuron j situated in the hidden layer. Within each neuron j, the computed values from all inputs go through summation, where each input is multiplied by its corresponding weight. The summation outcome in each neuron j is then processed through an activation function, represented as f. The output of f from each neuron j is further multiplied by its corresponding weight W_{oj}^2. The summation of all resulting products ultimately yields the network's output $\hat{y}k(z)$. The output $\hat{y}k(z)$ can be written as

$$\hat{y}k(z) = \left(\sum_{j=1}^{J} W_{oj}^2 h_j(z)\right), \tag{5}$$

where

$$h_j(z) = f\left(\sum_{i=0}^{n} W_{ji}^1 r(z-i) + \sum_{l=1}^{m} W_{jl}^1 y(z-l)\right), \tag{6}$$

In a typical ANN setup, every neuron is linked to weights and biases. However, in this study, biases are not employed for data modeling, as the data inherently centers around zero. The exclusion of biases offers the advantage of diminishing the number of parameters in the model, resulting in a simpler network with fewer computations.

The BP algorithm is employed to iteratively adjust the weights and biases of the ANN. This iterative process continues until the model attains the desired level of performance on the training data. Mathematically, the sum of squared error $E_k(z)$ is defined by measuring the discrepancy between $y_k(z)$ and $\hat{y}k(z)$. Subsequently, the partial derivatives of $E_k(z)$ with respect to W_{ji}^1, W_{jl}^1, and W_{oj}^2 are:

$$\frac{\partial E_k(z)}{\partial W_{oj}^2(z)} = \frac{\partial E_k(z)}{\partial \hat{y}_k(z)}\frac{\partial \hat{y}_k(z)}{\partial W_{oj}^2(z)}, \tag{7}$$

$$\frac{\partial E_k(z)}{\partial W_{ji}^1(z)} = \frac{\partial E_k(z)}{\partial \hat{y}_k(z)}\frac{\partial \hat{y}_k(z)}{\partial h_j(z)}\frac{\partial h_j(z)}{\partial W_{ji}^1(z)}, \tag{8}$$

$$\frac{\partial E_k(z)}{\partial W_{jl}^1(z)} = \frac{\partial E_k(z)}{\partial \hat{y}_k(z)}\frac{\partial \hat{y}_k(z)}{\partial h_j(z)}\frac{\partial h_j(z)}{\partial W_{jl}^1(z)}, \tag{9}$$

and the procedure for updating the weights is outlined as:

$$W_{oj}^2(z+1) = w_{oj}^2(z) - \eta\frac{\partial E_k(z)}{\partial W_{oj}^2(z)}, \tag{10}$$

$$W_{ji}^1(z+1) = w_{ji}^1(z) - \eta\frac{\partial E_k(z)}{\partial W_{ji}^1(z)}, \tag{11}$$

$$W_{jl}^1(z+1) = w_{jl}^1(z) - \eta\frac{\partial E_k(z)}{\partial W_{jl}^1(z)}, \tag{12}$$

where, η represents the learning rate.

To investigate diverse possibilities and determine the optimal ANN structure for EPS system identification, several tests were conducted. Step signals were employed for both data learning and validation, while sine, square, and triangular wave signals served as the test datasets. These datasets were extracted from EPS, as depicted in Fig. 5. The upper computer recorded all data, involving input reference signal data and corresponding EPS response output for each input. We systematically varied parameters such as the number and types of samples, the order of input variables (nth and mth order), and the number of neurons in the hidden layer. The number of hidden layers was restricted to one, and the tanh activation function was utilized. Details of the test configurations and results can be found in Table 2, while Algorithm 1 explains the training process for EPS system identification in this research.

IV. THE PROPOSED ANN-BASED LM ALGORITHM FOR EPS CONTROL

The ANN control system is a control system that the ANN algorithm to control a system. ANN is a type of machine

Algorithm 1 The Training Process of the System Identification

Initialize: weights W_{ji}^1, W_{jl}^1, W_{oj}^2
Select: η
Input: pattern-k, enter training dataset of $r(z), r(z-1), \ldots, r(z-n), y(z-1), y(z-2), \ldots, y(z-m)$
Output: $\grave{yk}(z)$
for $i \leftarrow 1$ to maximum_iteration **do**
  Calculation for neuron outputs:
  for z steps **do**
    Feedforward Calculation of first layer:
    feedforward first layer calculation $h_j(z)$
    utilize the tanh activation function
    Feedforward Calculation of second layer:
    feedforward second layer calculation $\grave{y}k(z)$
    Square Error calculation: E_k
    Backward Calculation of:
    Calculate the derivatives: $\frac{\partial Ek(z)}{\partial W_{oj}^2(z)}$, $\frac{\partial Ek(z)}{\partial W_{ji}^1(z)}$, $\frac{\partial Ek(z)}{\partial W_{jl}^1(z)}$
    Update the weights: $W_{oj}^2(z+1)$, $W_{ji}^1(z+1)$, $W_{jl}^1(z+1)$
  end
end

TABLE 2. Performance evaluation of predicted outputs of the EPS ANN system identification under various configurations.

No.	Learning iterations	System order	Neurons in hidden layer	Test data: Sine wave Best-fit at η 0.1 (%)	Sine wave Best-fit at η 0.01 (%)	Square wave Best-fit at η 0.1 (%)	Square wave Best-fit at η 0.01 (%)	Triangular wave Best-fit at η 0.1 (%)	Triangular wave Best-fit at η 0.01 (%)
1			3	94.9	97.8	96.0	98.1	95.4	98.0
2		2	5	98.7	97.4	99.7	97.7	98.9	97.8
3			9	64.1	97.9	74.4	99.1	65.9	98.1
4			3	95.5	94.7	96.3	95.4	15.3	77.1
5	10	3	5	86.0	98.4	89.6	99.1	67.5	-615.4
6			9	98.4	98.9	98.8	98.5	14.5	-233.6
7			3	58.3	90.5	68.0	89.7	67.4	-63.6
8		5	5	96.5	92.3	96.2	93.0	84.3	55.4
9			9	83.6	96.1	96.2	95.3	31.0	-219.1
10			3	99.4	99.3	99.6	99.8	99.5	99.4
11		**2**	**5**	**99.7**	**99.7**	**99.9**	**99.9**	**99.8**	**99.8**
12			9	99.3	99.5	99.4	99.9	99.4	99.7
13			3	99.2	99.3	99.5	99.8	78.1	-164.8
14	100	3	5	99.6	99.5	99.9	99.8	78.1	-107.3
15			9	99.6	99.5	99.7	99.9	-135.9	45.5
16			3	98.9	97.9	99.3	98.2	57.3	46.2
17		5	5	98.3	99.0	99.3	99.3	-8.6	0.8
18			9	96.9	99.6	96.8	99.6	7.7	-666.6

learning algorithm that can learn from data and make predictions. In designing the ANN for training the ANN controller, this study gathers data from the EPS system with a given sine wave signal at a frequency of 0.5 Hz as an angle input reference. The output response results with its corresponding input signal as depicted in Fig. 4(b). All data are shown in Fig. 4 recorded with a time sampling of 400 Hz. We used only one full sine wave of the data with its corresponding compensated control signals, totaling 800 data points. The

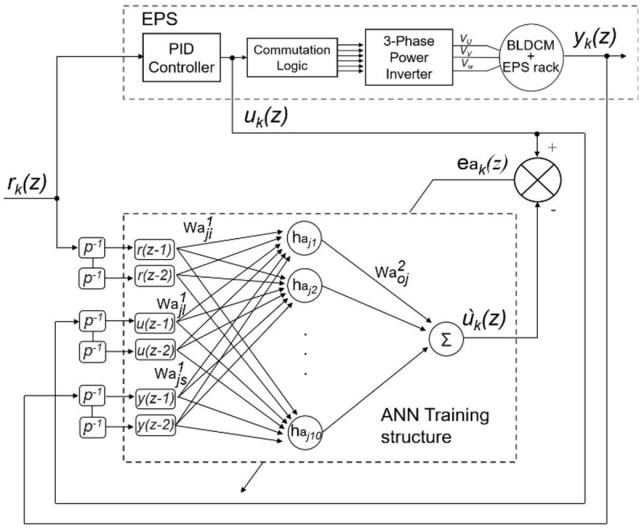

FIGURE 7. Proposed ANN Controller Training Model for EPS.

dataset was divided into training, validation, and testing sets with a ratio of 70%, 15%, and 15%, respectively. The training set is used to train the ANN, the validation set is used to tune the hyperparameter, and the testing set is used to evaluate the model's parameters. This model is then used as an EPS ANN controller in this study.

As depicted in Fig. 7, the ANN controller ANN_a training topology is 6-10-1. Which consists of 6 neurons in the input layer, 10 neurons in the hidden layer, and 1 neuron in the output layer. The input vectors of ANN_a are $[r(z\text{-}1),\ r(z\text{-}2),\ u(z\text{-}1),\ u(z\text{-}2),\ y(z\text{-}1),\ y(z\text{-}2)]^T$ and the output of ANN_a is $\grave{u_k}(z)$.

We thus have

$$\grave{u}_k(z) = \left(\sum_{j=1}^{10} \text{Wa}_{oj}^2 \text{ha}_j(z) \right), \tag{13}$$

where

$$\text{h}_{\text{a}_j}(z) = g_k \left(\sum_{i=1}^{2} \text{W}_{\text{a}_{ji}}^l r(z-i) + \sum_{l=1}^{2} \text{W}_{\text{a}_{jl}}^l u(z-l) + \sum_{s=1}^{2} \text{W}_{\text{a}_{js}}^l y(z-s) \right) \tag{14}$$

The $\text{W}_{\text{a}ji}^l$ and $\text{W}_{\text{a}jl}^l$ are the weight vectors in between input and hidden layer and $\text{W}_{a_{oj}}{}^2$ are the weight vector in between hidden and output layer. Fig. 7 is the proposed ANN angular controller training architecture scheme for the pre-trained model before being used as an EPS ANN controller.

In this research, The LM algorithm is used to adjust the weights of the networks such that the error eak between the desired output $u_k(z)$ and the estimated output $\grave{u}_k(z)$ approaches a very small value. Utilizing the LM algorithm, the error performance function Ea_k is expressed in the form

of a sum of squares error as

$$\mathrm{Ea}_k(z) = \frac{1}{2}\left(u_k(z) - \grave{u}_k(z)\right)^2 = \frac{1}{2}\mathrm{ea}_k^2(z). \tag{15}$$

The current gradient function $\nabla \mathrm{Ea}_k(z)$ as

$$\nabla \mathrm{Ea}_k(z) = \frac{\partial \mathrm{Ea}_k(z)}{\partial W(z)} = \mathrm{e}_{\mathrm{a}_k}(z)\frac{\partial \mathrm{e}_k(z)}{\partial W(z)} = J_c^T(z)\mathrm{e}_k(z), \tag{16}$$

where J_c is the Jacobian matrix. The Jacobian matrix J_c contains the first-order partial derivatives of the error function concerning the threshold and weight values. When the error function is minimized, the elements of the Jacobian matrix can be ignored, thus we have

$$\nabla^2 \mathrm{Ea}_k(z) = J_c^T(z)J_c(z). \tag{17}$$

The Hessian matrix is the second derivative of the error function with respect to the weight values. It contains information about how the error function changes as the weight values are changed. The Hessian matrix is used in the LM algorithm to compute the update step for the weight values $\mathrm{Wa}_{a_{kN}}(z+1)$. If the Hessian matrix is invertible, then the LM algorithm can be used to compute the exact update step that will minimize the error function. However, the Hessian matrix is not always invertible, especially when the weight values are close to the minimum of the error function. To use the LM algorithm even when the Hessian matrix is not invertible, a coefficient λ is introduced. This coefficient controls how much the LM algorithm relies on the Hessian matrix. When λ is large, the LM algorithm relies more on the Hessian matrix and behaves like the Gauss-Newton method. When λ is small, the LM algorithm relies less on the Hessian matrix and behaves like the gradient descent method [32], we thus have

$$\mathrm{Wa}_{kN}(z+1) = \mathrm{Wa}_{a_{kN}}(z) - \left[J_c^T(z)J_c(z) + \lambda I\right]^{-1} J_c^T(z)\mathrm{e}_k(z), \tag{18}$$

where $\mathrm{Wa}_{kN}(z)$ represents the parameters of the ANN at iteration z, J_c is the Jacobian matrix, representing the firstorder derivatives of the network's outputs with respect to its parameter, and $\mathrm{e}_k(z)$ is the error vector, representing the difference between the network's predictions and the actual target values.

Once the ANN is trained, it is then used to predict the system's output for a given input. This information can then be used to generate a control signal that will cause the system to achieve its desired state. In Fig. 8, we can see the proposed EPS ANN controller system block diagram. The ANN takes inputs from the angle command $r_k(z)$, the measured angle of $y_k(z)$ with time delays p^{-1} of two order system, and the control signal $u_k(z)$ as feedback also to the input. The ANN processes these input data and produces the control signal output $u_k(z)$, serving as a voltage reference for the BLDCM switching device.

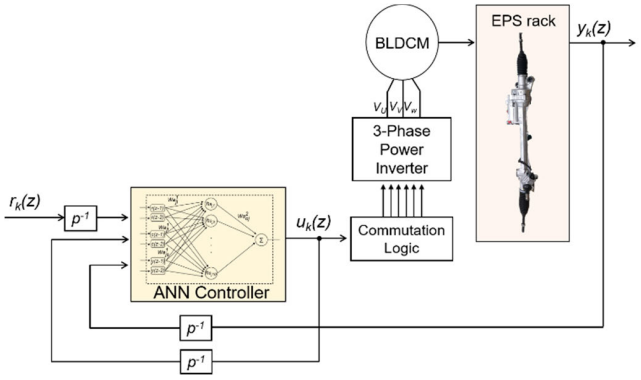

FIGURE 8. Overall EPS block diagram with Proposed ANN controller.

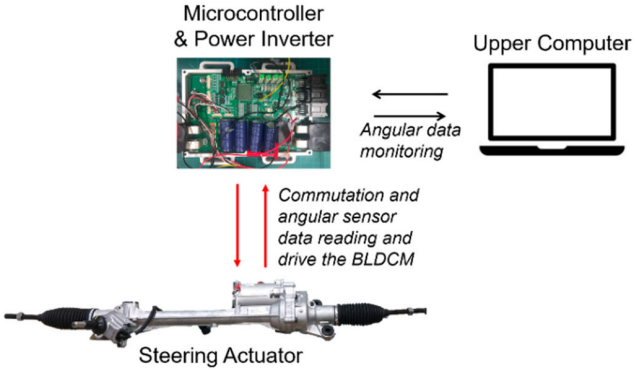

FIGURE 9. EPS testbed setup for Proposed ANN controller reviews.

V. EXPERIMENTAL RESULT AND DISCUSSION

A. H/W CONFIGURATION FOR EXPERIMENTAL SETUP

In this comprehensive experimental study, we investigated the effectiveness of EPS system identification using an ANN and the performance of EPS using an ANN controller. In the investigation of the ANN works in EPS system identification, we used five observed data taken from the EPS. For EPS system identification comparison, we use a linear transfer function estimator (TFE). For EPS control, the experiment was designed to assess the steering system's response in distinct scenarios of angular control.

The angular control scenario focused on the system's reaction to varying steering wheel angles using sine wave input as an angular position reference. Sine wave angle reference input, for this purpose, is used to simulate the act of the steering vehicle in real steering activity. We examined the controllers' capability to accurately translate the angle command input into wheel movements. Both EPS system identification and controller code were programmed in a microcontroller and the data of angle reference input and EPS response output were recorded by the upper computer via UART line. Fig. 9 shows the testbed setup for this purpose.

B. RESULTS OF EPS MODELLING

A common technique employed in system identification is the TFE. This method is utilized to characterize and estimate the dynamics of a linear time-invariant (LTI) system, where the connection between inputs and outputs remains constant over time and is unaffected by variations in the system's operating conditions. The transfer function acts as a mathematical representation of how the system's input and output are related in the frequency domain. It is defined as the ratio of the system's output to its input in the Laplace domain (s-domain). The general expression for the transfer function of an LTI system is:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{19}$$

where $H(s)$ denotes the transfer function, $Y(s)$ represents the Laplace transform of the system's output $y(t)$, $U(s)$ corresponds to the Laplace transform of the system's input $u(t)$, and 's' characterizes the complex frequency variable.

LTI systems comprise a subset of dynamic systems, but many real-world systems exhibit nonlinear behavior due to factors such as the nonlinear friction of the motor in our case, acting as an EPS actuator, and external uncertainties. In such scenarios, the Transfer TFE may struggle to accurately capture the system's dynamics, resulting in deficient model performance and limited predictive capabilities. The estimation results of the TFE model are depicted in Fig. 10.

Examining Fig. 10 reveals that this TFE yielded best-fit input-output models of 87.6%, 92.1%, and 87.3% for sine wave, square wave, and triangular wave test datasets, respectively. These outcomes highlight the effectiveness of ANN in system identification when compared to the TFE algorithm.

C. RESULTS OF EPS SYSTEM IDENTIFICATION USING ANN

When choosing an ANN architecture, it is crucial to carefully consider factors like task complexity, data characteristics, overfitting, computational resources, training time, and generalization capabilities. All the options detailed in Table 2 feature a rapid training process, but architecture number 11 stands out as an optimal choice. This selection strikes a balance between model complexity and the specific requirements of the problem. With a fast-training time, it can achieve excellent generalization performance. The results of the best-fit test model estimation in this study are presented in Fig. 11.

The optimal model for system identification in this research, as illustrated in Fig. 11, is obtained based on the test results using ANN with the design scenario specified as number 11. The network topology is set to 5-5-1, and a learning rate of 0.1 is applied. The error converges to the desired goal of 0.0167 after 100 iterations. Notably, at both learning rates of 0.1 and 0.01, design configuration number 11 achieves an optimal fit, expressing a match of over 99.6% between the model's predictions and the actual measured data across all testing dataset.

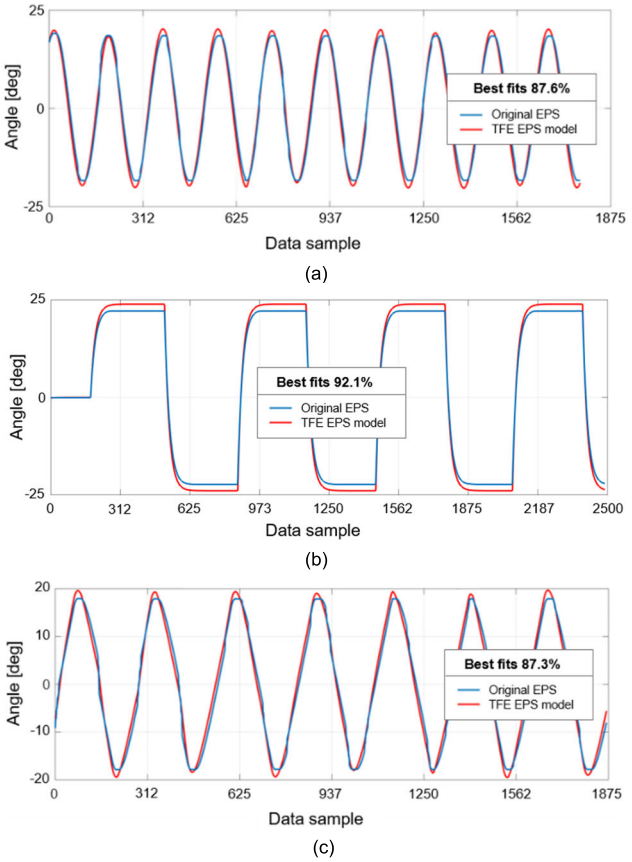

FIGURE 10. The time response outcomes for original EPS and the simplified 2nd order TFE model (a) Response to the sine wave, (b) Response to the square wave, (c) Response to the triangular wave.

D. TIME RESPONSE RESULTS OF DESIGNED EPS ANN CONTROLLER

In this research, we design two types of EPS controllers: PID and ANN controllers to see the performance of each controller in controlling the EPS system. The 10%-90% rise-time type, settling time response, steady-state analysis, and overshoot analysis was performed to analyze the performance of the EPS with respect to the step input reference.

The results of the EPS system responses using the proposed ANN controller with a given step input command of $\pm 10°$ and with the given sine wave at frequencies of 0.25 Hz, 0.5 Hz, and 1 Hz in the $\pm 10°$ and $\pm 5°$ of reference angle with the compensated signal in this study are shown in Figs. 12 to 15, respectively. The system with some given sine wave frequencies of 0.25 Hz, 0.5 Hz, and 1 Hz as input references to know the performance of the EPS PID and EPS controller. These selected frequencies are based on the normal range of steering frequency of a car [33].

The experimental results reveal a notable difference in compensation signals when the EPS target angle varies between $\pm 10°$ and $\pm 5°$. This suggests a strong correlation between the compensation signal and the specific target angle required for EPS performance in autonomous vehicles.

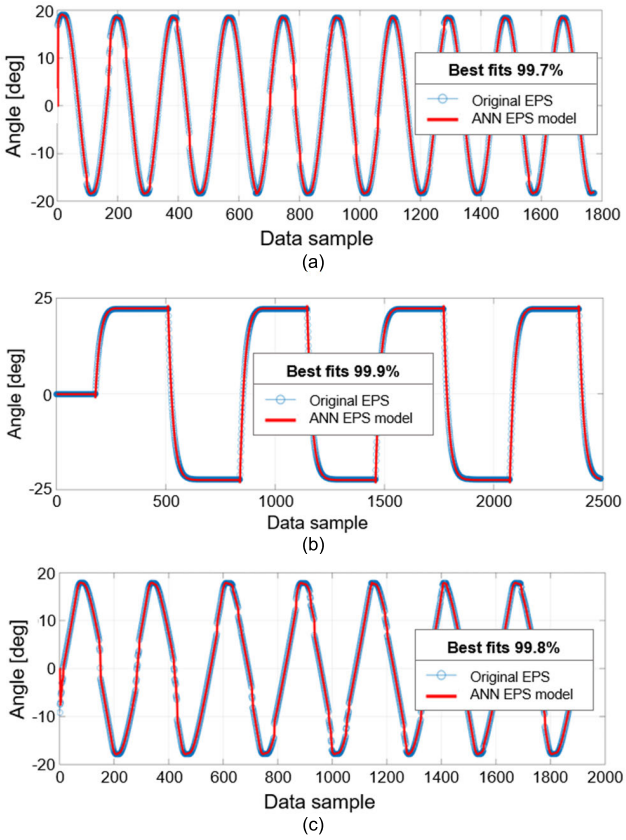

FIGURE 11. The time response outcomes for both the original EPS and the identified ANN model with network topology of 5-5-1, 100 learning iterations, and LR = 0.1. The responses to: (a) Sine wave, (b) Square wave, (c) Triangular wave.

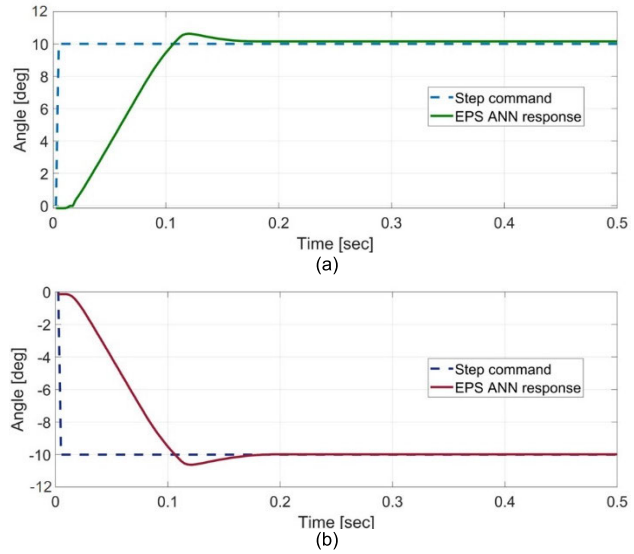

FIGURE 12. EPS response on step input of ±10°: (a) EPS response using ANN controller in clockwise rotation, (b) EPS response using ANN controller in counterclockwise rotation.

In essence, varying target angles in autonomous vehicles influence the dynamics of the steering mechanism and

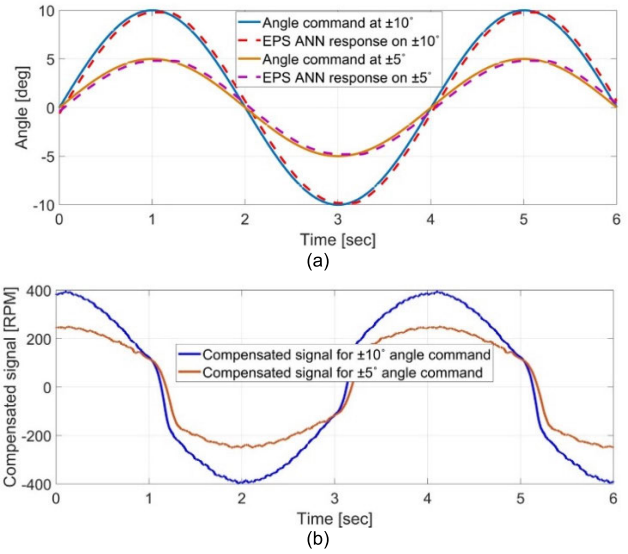

FIGURE 13. Time response results of EPS using ANN controller with sine wave input at a frequency of 0.25 Hz: (a) Responses on EPS angle target of ±10° and ±5°, (b) The compensated signals.

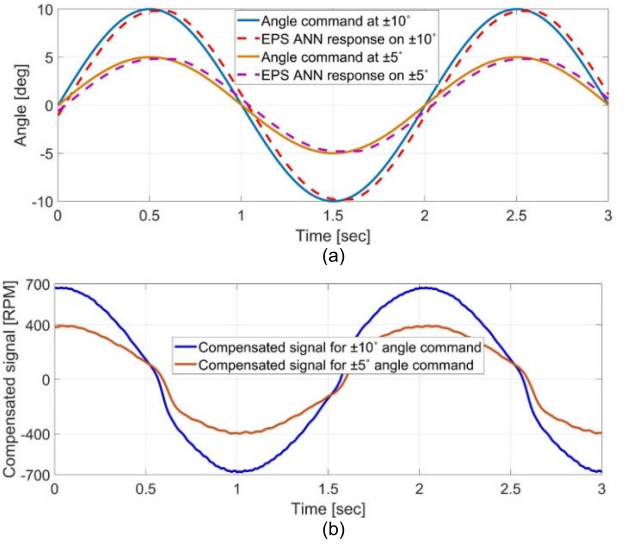

FIGURE 14. Time response results of EPS using ANN controller with sine wave input at a frequency of 0.5 Hz: (a) Responses on EPS angle target of ±10° and ±5°, (b) The compensated signals.

the vehicle's response characteristics. Larger target angles, necessitate more pronounced adjustments to the steering system to ensure the vehicle follows the desired trajectory.

E. RESULTS COMPARISON AND ANALYSIS OF ORIGINAL EPS WITH PID CONTROLLER AND PROPOSED ANN CONTROLLER

This section shows the comparison of EPS system response using both PID and ANN controllers. The comparison of PID and ANN controller EPS output performance corresponds to the provided step input of ±10° and input sine wave with angle peak target of ±10° using frequencies of 0.25 Hz,

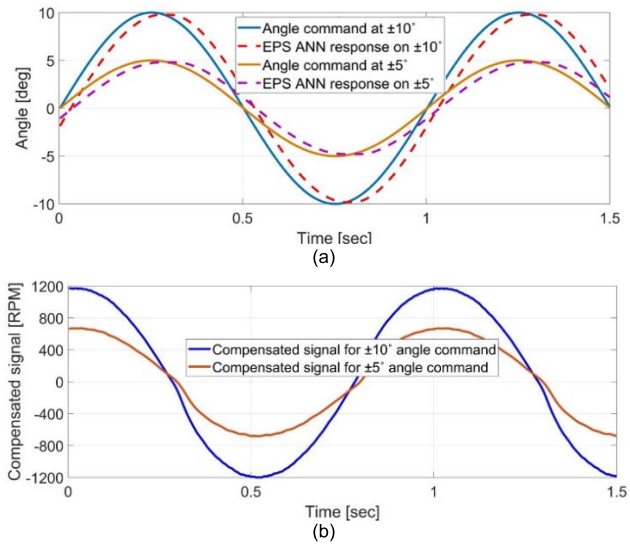

FIGURE 15. Time response results of EPS using ANN controller with sine wave input at a frequency of 1 Hz: (a) Responses on EPS angle target of ±10° and ±5°, (b) The compensated signals.

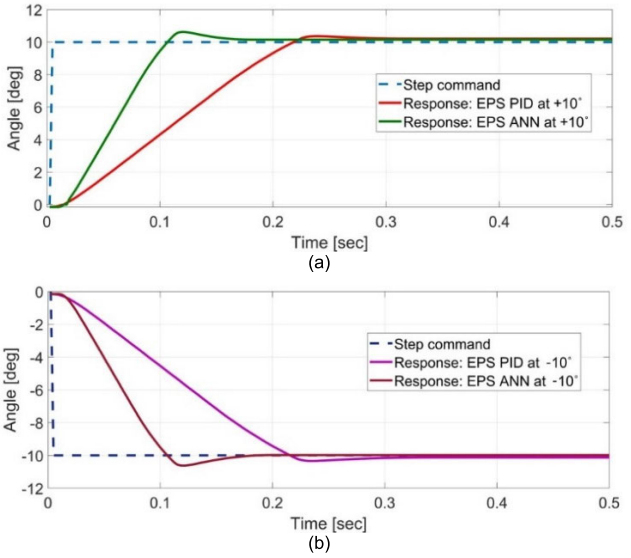

FIGURE 16. EPS response results on step input of ±10°: (a) EPS response using PID and ANN controllers in clockwise rotation, (b) EPS response using PID and ANN controllers in counterclockwise rotation.

TABLE 3. Results of EPS PID in time response analysis on input commands of ±10°.

Input step command	Rise time [sec]	Settling time [sec]	Steady state error [%]	Overshoot [%]
+10°	0.155	0.307	2.1	3.6
-10°	0.155	0.310	1.2	3.3

0.5 Hz, and 1 Hz. These comparison graphs can be seen in Fig. 16 and Fig. 17, respectively. Next, Table 3 and Table 4 show the results detail of EPS PID and EPS ANN controller response time with a given step input commands of ±10°.

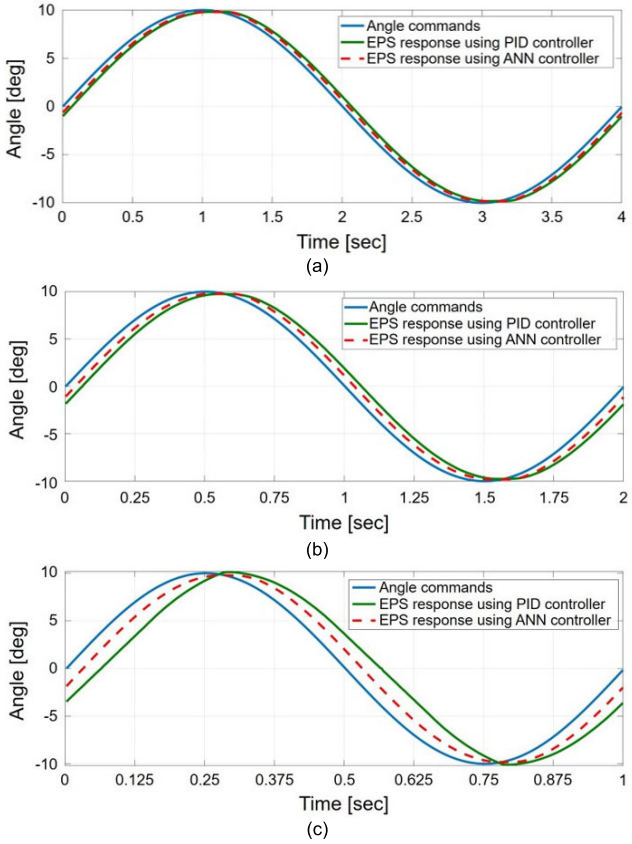

FIGURE 17. EPS PID and EPS ANN controllers output performance comparison corresponding to the provided input sine wave with angle target peak to peak of ±10°: (a) At a frequency of 0.25 Hz, (b) At a frequency of 0.5 Hz, (c) At a frequency of 1 Hz.

TABLE 4. Results of EPS ANN in time response analysis on input commands of ±10°.

Input step command	Rise time [sec]	Settling time [sec]	Steady state error [%]	Overshoot [%]
+10°	0.07	0.177	1.5	6.3
-10°	0.07	0.182	0.17	6.25

As we can see from the results in Table 3 and Table 4, the ANN controller has a faster response time compared with the PID controller. ANN controller has a 10-90% rise-time response of 0.07 seconds and settling time of 0.177 seconds in order to reach the +10° angle target and has a 10-90% rise-time response of 0.07 seconds and settling time of 0.182 seconds in order to reach -10° angle target. The signs (+) and (-) indicate the turning rotation of the right and left of the EPS system, respectively.

As illustrated in Fig. 17, a comparative study between PID and ANN controllers for an EPS system was conducted. Both controllers were implemented on identical testbed configurations detailed in Fig. 9. The results revealed that the ANN controller achieved a significantly faster response time compared to the PID controller, despite independent data

recording. This advantage stems from the inherent ability of ANN controllers to learn and adapt from data.

VI. CONCLUSION

This study investigated the application of ANN to enhance the identification and control of EPS systems for autonomous vehicles. Two approaches were developed:

- ANN-based identifier: This method, utilizing the BP algorithm, achieved a significant improvement in accuracy compared to the TFE method. Experiments demonstrated an achievement of over than 99.6% accuracy in predicting EPS system dynamics across various input signals.
- ANN-based controller: This approach, leveraging the LM algorithm, surpassed the PID controller. The ANN controller achieved a rise time of 0.07 seconds and settling times of 0.177 seconds (for a $+10°$ target) and 0.182 seconds (for a $-10°$ target), demonstrating faster response and improved reference tracking. Additionally, the controller consistently outperformed the traditional method when tested with varying sine wave inputs.

REFERENCES

[1] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016, doi: 10.1109/TIV.2016.2578706.

[2] H. Akhondi, J. Milimonfared, and K. Malekian, "Performance evaluation of electric power steering with IPM motor and drive system," in *Proc. 13th Int. Power Electron. Motion Control Conf.*, Poznan, Poland, Sep. 2008, pp. 2071–2075.

[3] A.-M. Nicorici, M. Ruba, C. S. Martis, L. Szabó, and Z. Máthé, "Comparative analysis of permanent magnet synchronous machines designed for electric power steering applications," in *Proc. 11th Int. Conf. Electr. Power Drive Syst. (ICEPDS)*, Oct. 2020, pp. 1–6, doi: 10.1109/ICEPDS47235.2020.9249074.

[4] Z. Máthé, A.-M. Nicorici, and L. Szabó, "Electrical machines used in electric power steering applications," in *Proc. 8th Int. Conf. Modern Power Syst. (MPS)*, Cluj, Cluj, Romania, May 2019, pp. 1–8, doi: 10.1109/MPS.2019.8759736.

[5] Z. Qun and H. Juhua, "Modeling and simulation of the electric power steering system," in *Proc. Pacific-Asia Conf. Circuits, Commun. Syst.*, Chengdu, China, May 2009, pp. 236–239, doi: 10.1109/paccs.2009.67.

[6] D. Lee, K.-S. Kim, and S. Kim, "Controller design of an electric power steering system," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 2, pp. 748–755, Mar. 2018, doi: 10.1109/TCST.2017.2679062.

[7] V. Nguyen, X. Guo, C. Zhang, and X. Tran, "Parameter estimation, robust controller design and performance analysis for an electric power steering system," *Algorithms*, vol. 12, no. 3, p. 57, Mar. 2019, doi: 10.3390/a12030057.

[8] M. T. Emirler, I. M. C. Uygan, B. Aksun Güvenç, and L. Güvenç, "Robust PID steering control in parameter space for highly automated driving," *Int. J. Veh. Technol.*, vol. 2014, pp. 1–8, Feb. 2014, doi: 10.1155/2014/259465.

[9] M. E. Abed, M. Aly, H. H. Ammar, and R. Shalaby, "Steering control for autonomous vehicles using PID control with gradient descent tuning and behavioral cloning," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Giza, Giza, Egypt, Oct. 2020, pp. 583–587, doi: 10.1109/NILES50944.2020.9257946.

[10] L. Wang and J. Ackermann, "Robustly stabilizing PID controllers for car steering systems," in *Proc. Amer. Control Conf. ACC*, 1998, pp. 41–42.

[11] R. Marino, S. Scalzi, G. Orlando, and M. Netto, "A nested PID steering control for lane keeping in vision based autonomous vehicles," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 2885–2890, doi: 10.1109/ACC.2009.5160343.

[12] G. Garimella, J. Funke, C. Wang, and M. Kobilarov, "Neural network modeling for steering control of an autonomous vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 2609–2615, doi: 10.1109/IROS.2017.8206084.

[13] M. Yildirim, M. C. Catalbas, A. Gulten, and H. Kurum, "Modelling and estimation parameters of electronic differential system for an electric vehicle using radial basis neural network," in *Proc. IEEE 16th Int. Conf. Environ. Electr. Eng. (EEEIC)*, Florence, Italy, Jun. 2016, pp. 1–5, doi: 10.1109/EEEIC.2016.7555798.

[14] Y. Li, G. Wu, L. Wu, and S. Chen, "Electric power steering nonlinear problem based on proportional-integral-derivative parameter self-tuning of back propagation neural network," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 234, no. 23, pp. 4725–4736, Dec. 2020, doi: 10.1177/0954406220926549.

[15] L. Fu and P. Li, "The research survey of system identification method," in *Proc. 5th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, vol. 2, Hangzhou, China, Aug. 2013, pp. 397–401, doi: 10.1109/IHMSC.2013.242.

[16] S. Chen and S. A. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *Int. J. Control*, vol. 56, no. 2, pp. 319–346, 1992, doi: 10.1080/00207179208934317.

[17] A. Yazdizadeh and K. Khorasani, "Identification of a class of nonlinear systems using dynamic neural network structures," in *Proc. Int. Conf. Neural Netw.*, Houston, TX, USA, 1997, pp. 194–198, doi: 10.1109/icnn.1997.611663.

[18] H. Han, W. Zhou, J. Qiao, and G. Feng, "A direct self-constructing neural controller design for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1312–1322, Jun. 2015, doi: 10.1109/TNNLS.2015.2401395.

[19] D. Xianglei, Z. Shuguang, H. Lvchang, and H. Rong, "The neural network direct inverse control of four-wheel steering system," in *Proc. 3rd Int. Conf. Measuring Technol. Mechatronics Autom.*, vol. 3, Shanghai, China, Jan. 2011, pp. 865–869, doi: 10.1109/ICMTMA.2011.789.

[20] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990, doi: 10.1109/5.58337.

[21] K. J. Shin, "Design of backpropagation neural network for aging estimation of electric battery," *Sensors Mater.*, vol. 35, no. 4, p. 1385, 2023, doi: 10.18494/sam4181.

[22] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990, doi: 10.1109/72.80202.

[23] T. Yamada and T. Yabuta, "Dynamic system identification using neural networks," *IEEE Trans. Syst. Man, Cybern.*, vol. 23, no. 1, pp. 204–211, Jan. 1993, doi: 10.1109/21.214778.

[24] A. Haddoun, M. E. H. Benbouzid, D. Diallo, R. Abdessemed, J. Ghouili, and K. Srairi, "Comparative analysis of estimation techniques of SFOC induction motor for electric vehicles," in *Proc. 18th Int. Conf. Electr. Mach.*, Vilamoura, Portugal, Sep. 2008, pp. 1–6, doi: 10.1109/icelmach.2008.4800166.

[25] K. Chernyshov, "A non-linear MIMO system identification approach based on the multiple maximal correlation technique," in *Proc. Int. Russian Autom. Conf.*, Sep. 2021, pp. 950–955, doi: 10.1109/RusAutoCon52004.2021.9537504.

[26] Md. M. Quamar and M. Aldhaifallah, "Instrumental variable system identification for time-delayed system with non-integer time-delay," in *Proc. 17th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Monastir, Tunisia, Jul. 2020, pp. 97–102, doi: 10.1109/SSD49366.2020.9364099.

[27] Y. Naung, A. Schagin, H. L. Oo, K. Z. Ye, and Z. M. Khaing, "Implementation of data driven control system of DC motor by using system identification process," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng.*, Jan. 2018, pp. 1801–1804, doi: 10.1109/EICONRUS.2018.8317455.

[28] Z. Hou, H. Gao, and F. Lewis, "Data-driven control and learning systems," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4070–4075, May 2017.

[29] J. Kang and W. Meng, "An improved method of system identification based on neural network," *Comput. Digit. Eng.*, vol. 40, no. 1, pp. 31–33, 2012.

[30] C. Wang, L. K. Xin, and J. F. Pan, "Identification of mechanical structure for servo drive system," in *Proc. 8th Int. Conf. Power Electron. Syst. Appl. (PESA)*, Dec. 2020, pp. 1–4.
[31] Y. Zhao, A. Fatehi, and B. Huang, "A data-driven hybrid ARX and Markov chain modeling approach to process identification with time-varying time delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4226–4236, May 2017, doi: 10.1109/TIE.2016.2597764.
[32] J. J. Moré, *The Levenberg–Marquardt Algorithm: Implementation and Theory in Numerical Analysis*. Cham, Switzerland: Springer, 1978, pp. 105–116.
[33] *Road Vehicles-Lateral Transient Response Test Methods-Open Loop Test Methods*. Accessed: Nov. 27, 2023. [Online]. Available: https://www.conforward.cn/ueditor/php/upload/file/20150901/1441081386459881.pdf

RODI HARTONO received the B.S. degree in electrical engineering from Universitas Komputer Indonesia, Indonesia, in 2010, and the M.S. degree in electrical engineering from Institut Teknologi Bandung, Indonesia, in 2014. He is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence Convergence, Busan University of Foreign Studies, Republic of Korea.

HYUN ROK CHA (Member, IEEE) received the Ph.D. degree in physics from Tokyo Institute of Technology, Tokyo, Japan, in 2009. He was with the Samsung Electronics Research Center, Gwangju, for a period of four years. Since 2004, he has been a Senior Researcher with the Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology, where he is also the current Director of the Mobility Core Components and Materials Center, Seonam Division, and a Professor with the University of Science and Technology. His research interests include E-mobility, electric vehicle (EV) platforms, smart vehicle control hybrid-powered drones, and power electronics.

KYOO JAE SHIN (Member, IEEE) received the B.S. degree in electronics engineering, in 1985, the M.S. degree in electrical engineering from Chonbuk National University, in 1988, and the Ph.D. degree in electrical science from Pusan National University, in 2009. He is currently a Professor in intelligent robot science with Busan University of Foreign Studies, South Korea. He is also the Director of the Future Creative Science Research Institute, BUFS. He has been a Professor with the Navy Technical Education School and the main Director and a Research Associate in dynamic stabilization systems with Busan Defense Weapon Research Institute. He has also researched and developed a fish robot, a submarine robot, an automatic dog spaying robot operating in a glass room, an automatic milking robot using a manipulator, a personal electrical vehicle, a smart accumulated aquarium using a heat pump, a solar tracking systems, a 3D hologram systems, and a gun/turret stabilization systems. His research interests include intelligent robots, image signal processing application systems, smart farms, and aquariums using new energy and IoT technology.