IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 May 2023, accepted 22 May 2023, date of publication 30 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281397

RESEARCH ARTICLE

Data-Driven Adaptive Steady-State-Integral-Derivative Controller Using Recursive Least Squares With Performance Conditions

JEONGWOO LEE[1] AND KWANGSEOK OH[2], (Member, IEEE)
[1]Research and Development Center, Bundang-gu, Seongnam-si, Gyeonggi-do 13488, Republic of Korea
[2]School of ICT, Robotics & Mechanical Engineering, Hankyong National University, Anseong-si, Gyeonggi-do 17579, Republic of Korea

Corresponding author: Kwangseok Oh (oks@hknu.ac.kr)

ABSTRACT This paper presents a data-driven adaptive steady state-integral-derivative (SS-ID) control algorithm that uses gradient descent and recursive least squares (RLS) with a forgetting factor. A simplified first-order differential equation of the control system was designed and its parameters were estimated in real-time using the RLS algorithm. The steady-state control input for target-state tracking was derived based on the estimated parameters and steady-state performance conditions. The gradient of the integrated control error to the gain was estimated based on the least-squares method, using the saved past error and gain data in a finite sliding window to determine the control input. The integral gain was adapted based on the gradient descent method, using the estimated gradient, integrated error, and adaptation rate. Simplified control error dynamics were designed, and their parameter was estimated using the RLS algorithm. The derivative control gain can be adapted in real time using the estimated parameters from the simplified control error dynamics and time constant-based performance conditions. The proposed controller was designed in the MATLAB/Simulink environment. A performance evaluation was conducted under various scenarios using a DC motor simulation model and an actual test platform equipped with an optical encoder.

INDEX TERMS Data-driven adaptive control, steady state-integral-derivative control, gradient descent, recursive least squares, forgetting factor, performance condition.

I. INTRODUCTION

Modern robot systems, including humanoid robots, drones, and vehicles, have used various actuators such as electric and hydraulic motors to control translational and rotational motion. Robot systems have been improved by integrating actuators to make human life safer, more efficient, and more convenient. However, robot systems have become increasingly complex, and their nonlinearity has increased accordingly because several types of actuators and subsystems are used. Suppose that a robot system is complex and its nonlinearity is relatively high. In such a case, it is not easy to derive the mathematical model of the system. Various uncertainties exist between the mathematical model and the existing system, such as unmodelled dynamics and parameter uncertainties. If an inaccurate mathematical model (the system is unpredictable by existing uncertainties) is used, system behavior cannot be predicted, and current behavior cannot be precisely represented. The comparatively massive uncertainty can harm control performance because of inaccuracy, and it may be necessary to design an adaptation algorithm to improve performance. In order to overcome these limitations, various studies on control technologies such as optimal, adaptive, and model-free control algorithms have been conducted by research institutes and prominent universities.

Liu et al. [1] proposed a robust fractional-order proportional-integral-derivative controller design for an autonomous underwater vehicle's yaw control system using a three-dimensional stability region analysis method. Wang et al. [2] presented an improved fuzzy PID control method considering hydrogen fuel cell (HFC) voltage-output characteristics. They realized coordinated control of the

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

HFC and power cell by developing an electrochemical and dynamic model for HFC vehicles. Phu and Choi [3] developed a modified Riccati-like equation-based controller with an interval type 2 fuzzy model to ensure robustness to parameter uncertainties and support the calculation progress. Guerrero et al. [4] designed a nonlinear PID controller based on a set of saturation functions for trajectory tracking on underwater vehicles to consider the drawback that the previous methodology needs to be more robust to encompass large and persistent parameter uncertainties. The main advantages of the proposed controller are that it preserves the advantages of a robust control algorithm and allows easy tuning of the control parameters in real applications. Mandava and Vundavilli [5] developed a gain modification algorithm for a torque-based PID controller using a neural network trained by nature-inspired optimization algorithms. The effectiveness of the developed algorithm was tested in computer simulations and on a real bipedal robot. Previous have studies mentioned the aforementioned adaptation laws and used fuzzy algorithms for tracking control algorithm design; however, an unreasonable determination of the adaptation rate or law can harm the target tracking control performance. Therefore, studies on the proportional-integral-derivative (PID) control of various systems have been conducted to ensure robust control stability and improve performance by integrating other control algorithms, such as the sliding-mode control algorithm.

Noordin et al. [6] proposed an adaptive proportional integral derivative control scheme based on second-order sliding-mode control to tune the parameter gains of a PID controller. Moreover, a fuzzy compensator was used to reduce the chattering of the adaptive controller. In addition, Noordin et al. [7] proposed an auto-tuning adaptive proportional-integral-derivative control system for the attitude and position stabilization of quadrotor unmanned aerial vehicles under parameter uncertainties and external disturbances by employing sliding-mode control as the adaptive mechanism. Guo and Ahn [8] investigated an adaptive fault-tolerant pseudo-proportional-integral-derivative sliding-mode control scheme for a high-speed train subject to actuator faults, asymmetric nonlinear actuator saturation, and integral quadratic constraints. They presented simulation results based on a real train dynamic model to demonstrate the main contributions of their study and the effectiveness and feasibility of the proposed schemes. Zhong et al. [9] proposed a trajectory-tracking control method for redundant manipulators using a fuzzy logic system and adaptive control-based chattering attenuation. They integrated the continuous approximation law to eliminate the real-time chattering during the control process without affecting the system's robustness. In addition, they designed a fuzzy adaptive control law to estimate the dynamic system's error and the disturbance's upper bound. They demonstrated the proposed method using experimental data from the Baxter robot, which showed better performance than the previous method. To solve the problem of leader tracking control, Lui et al. [10] proposed a novel distributed PID-like control strategy with a Lyapunov-based adaptation mechanism for control parameters. Yang et al. [11] proposed an adaptive sliding-mode PID control method for underwater manipulators based on the Legendre polynomial function approximation technique. They designed a sliding-mode PID controller to accelerate the system response and reduce joint lag using the adaptive law. Zhang et al. [12] designed a hybrid control strategy for particle swarm sliding-mode fuzzy PID control to weaken the chattering of the sliding-mode control using particle swarm optimization. To solve the problem of tuning PID parameters, Kobaku et al. [13] proposed a design method that uses quantitative feedback theory in conjunction with particle swarm optimization to perform automatic loop shaping. Mahmoodabadi and Nejadkourki [14] proposed an optimal fuzzy adaptive robust proportional-integral-derivative controller for a quarter-car model with an active suspension system based on integral sliding surfaces defined by control errors. Kavyashree et al. [15] presented an observer- and state-estimator-based anti-windup robust proportional-integral-derivative controller for a damped outrigger structure using a magnetorheological damper to mitigate the seismic response. They designed a full-order Kalman observer to estimate the states of a damped outrigger system based on the feedback of the system output. The previous studies proposed rules for the adaptation of control parameters using robust control methods or optimization strategies to ensure control stability. However, excessive smoothing of the robust control input, such as the discrete sliding-mode control input, can cause a loss of stability depending on the individual circumstances. Those circumstances include unreasonable uncertainty estimation and improper parameter determination used for control input calculation. Therefore, various studies using data-driven and learning methods have been conducted to improve system control performance.

Jeyaraj and Nadar [16] designed a deep learning-based data-driven PID controller for unmodeled dynamics compensation of complex industrial processes. The authors investigated specific modeling of data-driven proportional-integral-derivative for complex industrial processes with consideration of online PID parameter tuning. To evaluate the performance, they employed the proposed deep learning-based PID controller in a twin-tank control system. Yu et al. [17] presented a new design scheme for PID controllers based on adaptive updating rules and data-driven techniques. They provided a rigorous Lyapunov-based proof of stability to ensure the convergence of the tracking errors when the initial states belong to a compact set. Makarem et al. [18] proposed data-driven techniques for the iterative feedback tuning of proportional-integral-derivative controller parameters and compared different motor driving techniques. Li and Yu [19] developed a novel deep-reinforcement-learning algorithm called the two-stage training strategy large-scale twin delayed deep determination policy gradient. Zhu et al. [20]

proposed a data-driven approach for the online tuning of minimum-variance PID controllers for a linear system subject to stochastic conditions. Rahmani and Redkar [21] developed a fractional-order PID controller to control a dynamic model linearized by the Koopman operator based on dynamic mode decomposition. They showed that the proposed controller performed better than other controllers in terms of low tracking errors and low control effort. Shuprajhaa et al. [22] developed an adaptive PID controller for controlling open-loop unstable processes based on generic data-driven modified proximal policy-optimization reinforcement learning. The main feature of the proposed control scheme is that it can eliminate the need for process modeling, as well as prerequisite knowledge of system dynamics and control parameter tuning. Wang et al. [23] proposed a linear model and an uncertain nonlinear dynamic system that described the unknown time-varying dynamics of nonlinearity and couplings among controlled variables. The authors developed a feed-forward and one-step optimal control algorithms to obtain two signal compensators to eliminate the negative impact of frequent unexpected variations in the uncertainties. Saad et al. [24] presented the implementation of PID controller tuning using two heuristic techniques which are differential evolution and genetic algorithm. The PID control parameters were applied for higher order system and the performance was evaluated based on mean square error and integral absolute error. Zamani et al. [25] presented and studied application of fractional order PID controller to an automatic voltage regulator with particle swarm optimization and novel cost function. Hsu and Lee [26] proposed an adaptive PID controller and fuzzy compensator. The proposed controller can automatically online tune the control gains based on the gradient descent method and the fuzzy compensator without requiring preliminary offline learning.

This study proposes a data-driven adaptive steady state-integral-derivative control algorithm that uses recursive least squares with forgetting factors. The steady-state control input is derived using simplified error dynamics with real-time parameter estimation. An adaptation algorithm was constructed based on the designed performance conditions for the integral and derivative control inputs. To evaluate the performance of the proposed control algorithm, the proposed control algorithm was evaluated based on simulations and an actual test platform using a DC motor system. The main contributions of this study are summarized as follows.

Problem statement: System state and environmental condition changes can have a negative impact on the performance of PID control with constant gains.

Contribution 1: A novel adaptive feedback control method called the steady-state-integral-derivative (SS-ID) controller using only control error data is proposed in this study.

Contribution 2: A self-tuning algorithm for control inputs, such as steady-state, integral, and derivative inputs, was designed for target value tracking based on RLS with a forgetting factor.

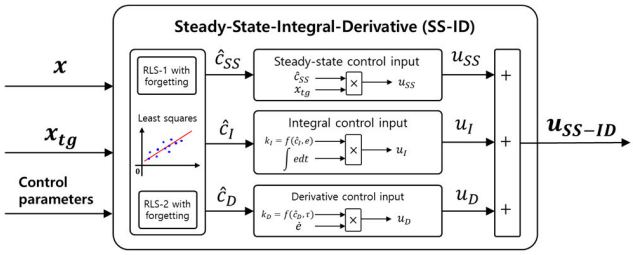

FIGURE 1. Concept of data-driven adaptive SS-ID control algorithm.

The remainder of this paper is organized as follows. Section II describes the concept of the proposed data-driven adaptive SSID controller. Section III describes the mathematical design procedure for the data-driven adaptive SS-ID control algorithm. Section IV describes the controller simulation and actual test-platform-based performance evaluation results. Finally, concluding remarks are provided along with the scope for future work in Section V.

II. CONCEPT OF DATA-DRIVEN ADAPTIVE STEADY-STATE-INTEGRAL-DERIVATIVE CONTROLLER

Fig. 1 illustrates the main concept of the proposed data-driven adaptive steady state-integral-derivative controller.

The control algorithm proposed in this study uses the system state, target state, and certain control parameters to derive the SS-ID control input. The parameter estimation block was designed to estimate parameters, simplify the system dynamics, and control the error dynamics using RLS with forgetting. Moreover, a parameter representing the partial derivative of the defined error with respect to the integral gain has been estimated using the least-squares method. The total control input of the proposed control consists of three control inputs: steady-state, integral, and derivative control. The proposed control algorithm has the advantage of not needing any system parameters and dynamics for target value tracking. Related previous works are based on model-free adaptive control [3], [4], [5], [6], [7], [8], [9], [10], [11], data-driven control, and artificial intelligence control [16], [17], [18], [19], [20], [21]. In the previous related works, generally, the conventional proportional-integral-derivative control scheme was adopted, and the gain adaptation law was proposed using various methods. This study proposes a novel control scheme of a steady-state-integral-derivative controller using recursive least squares with a forgetting factor. Therefore, the proportional gain used in the conventional PID control scheme is not used in the proposed approach. However, because the proposed controller is based on simplified first-order differential equations, determining appropriate initial parameters is necessary for reasonably stable performance in the early evaluation stage. The following section describes the proposed data-driven adaptive SS-ID controller.

III. DATA-DRIVEN ADAPTIVE SS-ID CONTROLLER

Fig. 2 shows a detailed block diagram of the data-driven adaptive steady state-integral-derivative controller. The control

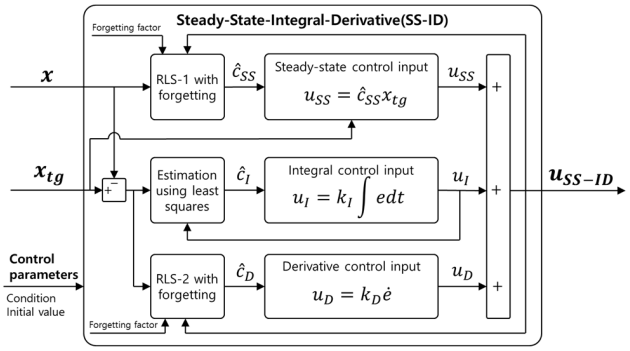

FIGURE 2. Detailed block diagram of data-driven adaptive SS-ID control algorithm proposed in this study.

block consists of three sub-blocks: an adaptive steady-state control input derivation block, adaptive integral control input derivation block, and adaptive derivative control input derivation block with three estimation sub-blocks.

In the parameter estimation block using RLS-1, the parameter $\hat{c}_{SS}$ is estimated based on a simplified first-order differential equation using the system state x; based on the estimated parameter $\hat{c}_{SS}$ and state x, the steady-state control input u_{SS} is computed. This was designed such that the control error (the difference between state x and target state x_{tg}, $e = x_{tg} - x$) is used to estimate parameters $\hat{c}_I$ and $\hat{c}_D$. Finally, the integral and derivative control inputs u_I and u_D are computed using the estimated parameters $\hat{c}_I$ and $\hat{c}_D$. Equations (1) and (2) represent the total control input designed in this study and the definition of each control input, respectively.

$$u_{SS-ID} = u_{SS} + u_I + u_D \tag{1}$$

$$u_{SS} + u_I + u_D = \hat{c}_{SS} x_{tg} + k_I \int edt + k_D \dot{e} \tag{2}$$

where k_I and k_D are the integral and derivative feedback gains, respectively. The steady-state control input designed in this study is computed by multiplying the estimated parameter and target state. The following subsections describe the mathematical formulation of the steady-state, integral, and derivative control inputs for target state tracking.

A. STEADY-STATE CONTROL INPUT

In this study, a steady-state condition-based control input was designed using the target state and system parameters. Generally, the larger the target state applied to the control system, the larger the control input required for target state tracking. Therefore, based on this relationship, the control input can be designed according to the target state and system parameters. To compute the steady-state control input, the following simplified first-order differential equation was designed and used in this study.

$$\dot{x} + c_{SS} x = u_{SS} + \left(\frac{1}{1 - k_D}\right) u_I \tag{3}$$

The parameter c_{SS} is time-varying, and its dimension is 1×1. Therefore, The k_D in Equation (3) should not equal one to prevent divergence of the input value of the simplified first-order differential equation. In this study, the parameters c_{SS} were estimated using RLS with a forgetting factor because the past dynamic data can be considered for parameter estimation by using a forgetting factor. In steady-state conditions, the steady-state control input can be computed by the vector production of the estimated parameters c_{SS} and target state x_{tg} with the designed adaptation rule for integral control input. Equations (4)–(8) are the RLS equations for state estimation with covariance updates.

$$y = \varphi\theta \tag{4}$$

$$y = u_{SS} + \left(\frac{1}{1 - k_D}\right) u_I - \dot{x}, \varphi = x, \theta = c_{SS} \tag{5}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k \left(y_k - \varphi_k \hat{\theta}_{k-1}\right) \tag{6}$$

$$L_k = P_{k-1}\varphi_k \left(\lambda + \varphi_k^2 P_{k-1}\right)^{-1} \tag{7}$$

$$P_k = \left(I - L_k \varphi_k\right) P_{k-1}/\lambda \tag{8}$$

where $L \in R^{1\times1}$ and $P \in R^{1\times1}$ are gain and covariance for RLS estimation, respectively. Based on the assumptions that the derivative of the state x with respect to time t is negligible and that the control inputs u_I and u_D are zero, the steady-state control input can be computed using the estimated parameter $\hat{c}_{SS}$ and target state x_{tg}. Equation (9) shows the steady-state control input designed in this study.

$$u_{SS} = f(\hat{c}_{SS}, x_{tg}) = \hat{c}_{SS} x_{tg} \tag{9}$$

Using the computed steady-state control input, the magnitude of the control input can be reasonably determined using the target state and estimated system parameters. The following subsection describes the mathematical formulation of the integral control input derivation.

B. INTEGRAL CONTROL INPUT

The integral control input in this study was designed using only the current and saved past control error data and adaptation parameters using the gradient descent method. The gradient descent method requires a partial derivative of the adaptation parameter to the control error for gradient descent. To reduce the current and integrated control errors complexly, (10) was used to derive a partial derivative of the integral feedback gain to the integrated control error ($w_I e + \int edt$).

$$\frac{d}{dt}\left(w_I e + \int edt\right) = c_I \frac{d}{dt}(k_I) \tag{10}$$

where w_I is the weighting factor for the design of the integrated control error for the adaptation of the integral feedback gain. This relationship function is newly designed in this study. The parameter c_I is estimated using the least-squares method according to the current and saved past data of the integrated error and integral feedback gain. Fig. 3 shows the estimation concept of the parameter c_I using the least-squares method based on the designed weighted error and adapted integral feedback gain.

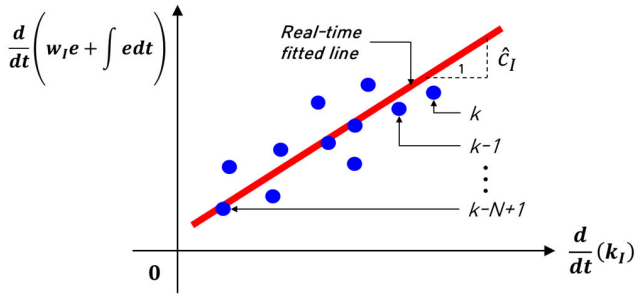

FIGURE 3. Estimation concept of the parameter c_I based on the least-squares method.

The points on the derivatives of the integrated control error and integral feedback gain plane were used to estimate the first-order polynomial equation ($y = ax + b$), illustrated as the straight red line in Fig. 3. In the current step, the a and b are estimated simultaneously by the least-squares method to consider changing characteristic of the past data points in the plane. The number of points in the plane is N and the value is defined as 1,000. The slope a of the estimated first-order polynomial equation was used as the estimated parameter $\hat{c}_I$ and for the integral feedback gain adaptation. Equation (11) describes the gradient descent method using the estimated parameter $\hat{c}_I$.

$$\frac{d}{dt}(k_I) = -\gamma \left(w_I e + \int edt\right)\hat{c}_I \tag{11}$$

where γ is the adaptation gain. Equation (11) can be rewritten by integrating both sides.

$$k_I = -\gamma \int \left(w_I e + \int edt\right)\hat{c}_I dt \tag{12}$$

Based on the adapted integral gain and integrated control error, the integral control input is computed as follows.

$$u_I = k_I \int edt = -\gamma \int \left(w_I e + \int edt\right)\hat{c}_I dt \int edt \tag{13}$$

The following subsection describes the design procedure for the derivative control input for target state tracking.

C. DERIVATIVE CONTROL INPUT

This subsection presents the derivative control input design method for target state tracking based on the designed first-order differential equation of error dynamics under the designed performance conditions. The following is the first-order differential equation designed in this study for derivative gain adaptation under the designed performance conditions.

$$\dot{e} + c_D e = u_I + u_D, \tag{14}$$

Based on the partial control input $u_I + u_D$ and the control error rate, parameter c_D in Equation (14) was estimated using RLS with a forgetting factor. The same procedures as in (4)–(8) were used to estimate c_D. Equation (15) defines the response, regressor, and estimate.

$$y = u_I + u_D - \dot{e}, \varphi = e, \theta = c_D \tag{15}$$

Using the definition of the derivative control input $u_D = k_D\dot{e}$, the designed first-order differential equation can be rewritten using the estimated parameter $\hat{c}_D$ as follows.

$$\dot{e} + \left(\frac{\hat{c}_D}{1-k_D}\right)e = \left(\frac{1}{1-k_D}\right)u_I \tag{16}$$

This study designed performance conditions for reasonable target state tracking using the desired time constant τ_D. The k_D in Equation (16) should not equal one to prevent divergence of the input value of the simplified first-order differential equation. Equation (17) expresses the performance condition defined in this study using the desired time constant.

$$\frac{\hat{c}_D}{1-k_D} = \frac{1}{\tau_D} \tag{17}$$

Based on the defined performance condition, the derivative gain k_D for target state tracking is computed using the estimated parameter and the desired time constant.

$$k_D = 1 - \hat{c}_D\tau_D \tag{18}$$

With the derived steady state control input, integral control input, and derivative control input, the total control input for target state tracking can be computed as follows.

$$u_{SS-ID} = \hat{c}_{SS}x_{tg} - \left(\gamma \int \left(w_I e + \int edt\right)\hat{c}_I dt\right)\int edt + (1 - \hat{c}_D\tau_D)\dot{e} \tag{19}$$

The following subsection describes the Lyapunov theorem-based stability analysis of the proposed steady state-integral derivative control algorithm.

D. STABILITY ANALYSIS

The Lyapunov direct method was used in this study for stability analysis of the proposed control algorithm. The following energy-like Lyapunov candidate was used to check the control stability and derive the conditions.

$$J = \frac{1}{2}e^2 \tag{20}$$

Because the derivative of function of the Lyapunov candidate with respect to time t should always be negative for asymptotic stability, the derivative of the function J with respect to time t was derived and used for stability analysis. Using (16), the derivative of J with respect to time t can be derived as follows [10], [17].

$$\dot{J} = e\dot{e} = e\left(-\frac{1}{\tau_D}e + \left(\frac{1}{1-k_D}\right)u_I\right) \tag{21}$$

The integral control input in (21) can be replaced by the integral control input derived from (3). Under the steady-state condition, the integral control input can be derived using the

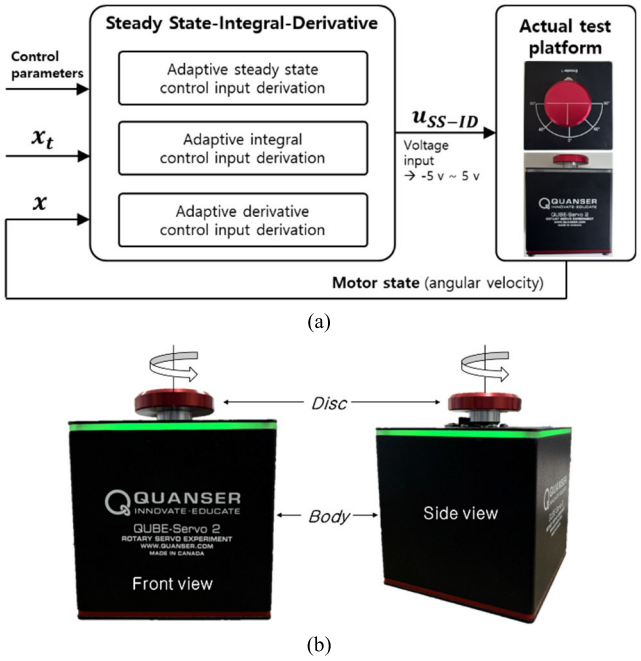

FIGURE 4. (a) Schematic of the model used for performance evaluation. (b) Actual test platform: DC motor system with optical encoder.

following equation with the defined steady-state control input in (9).

$$u_I = -(1 - k_D)\,\hat{c}_{SS} e \tag{22}$$

Using the derived integral control input in (22), the derivative of the Lyapunov candidate function with respect to time t in (21) can be rewritten as follows.

$$\dot{J} = -\left(\frac{1}{\tau_D} + \hat{c}_{SS}\right) e^2 \tag{23}$$

To ensure the stability of the control algorithm, the derivative of the Lyapunov candidate function with respect to time t should be less than or equal to zero. Therefore, one condition that stabilizes the control system can be derived as follows.

$$\hat{c}_{SS} \geq -\frac{1}{\tau_D} \tag{24}$$

Based on the stability condition, estimation parameters such as the covariance, initial value, and forgetting factor that can satisfy the inequality condition in (24) were determined for control stability. Equation (23) can be rewritten as follows using the stability condition in (24).

$$\dot{J} = -\left(\frac{1}{\tau_D} + \hat{c}_{SS}\right) e^2 < 0, e \neq 0 \tag{25}$$

The next section presents the performance evaluation results and analysis based on the simulation and test platform.

IV. PERFORMANCE EVALUATION

Fig. 4 shows a schematic of the model used for the performance evaluation of the proposed control algorithm.

TABLE 1. Used system parameters.

Parameter description	Value [unit]
Terminal resistance	8.4 [Ω]
Torque constant	0.042 [Nm/A]
Motor back-EMF constant	0.042 [V/(rad/s)]
Rotor inductance	1.16 [mH]
Inertia	4.6×10^{-6} [kgm^2]

TABLE 2. Used PID gains for performance evaluation.

Division	Fixed case	Adaptation case
Proportional gain	0.1	Online adaptation
Integral gain	0.1	Online adaptation
Derivative gain	0.05	Online adaptation

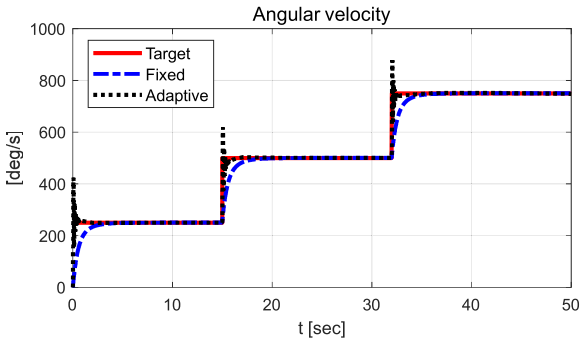

FIGURE 5. Simulation-based results (step target velocity): angular velocity - target, fixed, and adaptive.

An actual DC motor test platform (QUBE-Servo 2) equipped with an optical rotary encoder to measure the angular displacement was used to evaluate the performance of the proposed control algorithm. A transfer function was designed using Simulink to estimate the angular velocity of the motor. The control algorithm was designed in the MATLAB/Simulink environment, and the actual test platform was connected to MATLAB/Simulink for real-time measurement and control for the performance evaluation. The system parameters of the DC motor provided by the company are listed in Table 1, and it was used for the simulation-based performance evaluation. Table 2 lists the applied control gains for each performance evaluation case. The control gains for fixed case were determined to represent the case that the system or load applied to the system varies with time. Evaluation times of 50 s and 20 s were applied to the simulation and experiment, respectively.

A. SIMULATION-BASED EVALUATION

- Step target velocity (simulation)

Figs. 5–13 show the simulation-based performance evaluation results when the step target velocity was applied.

As shown in Figs. 4 and 5, the tracking result in the adaptive case showed better tracking performance than that in the fixed case. The result in the adaptive case showed that there was

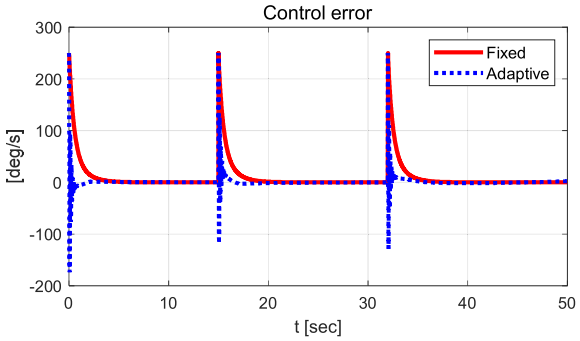

FIGURE 6. Simulation-based results (step target velocity): control error - fixed and adaptive.

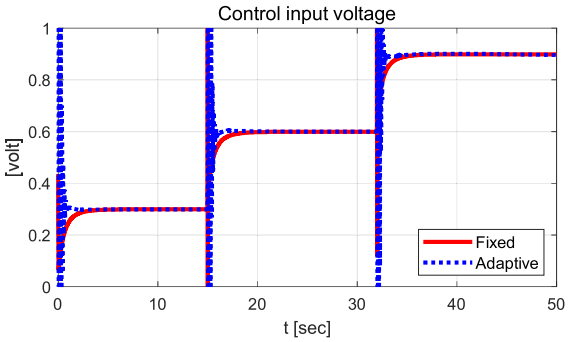

FIGURE 7. Simulation-based results (step target velocity): control input voltage - fixed and adaptive.

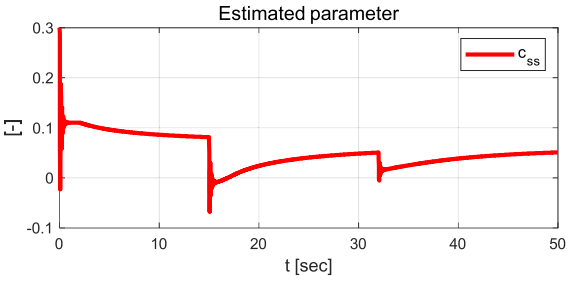

FIGURE 8. Simulation-based results (step target velocity): estimated parameter - c_{ss}.

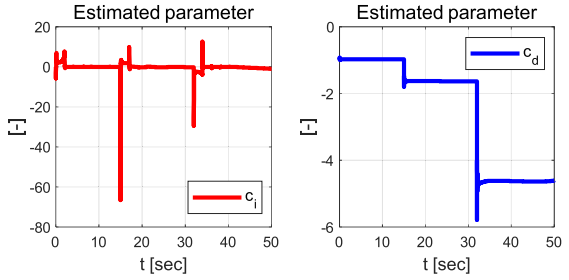

FIGURE 9. Simulation-based results (step target velocity): estimated parameter - c_i (left) and c_d (right).

an overshoot, but the control errors in the transient region were relatively small. Fig. 7 shows the applied control inputs for each case. Figs. 8 and 9 show the estimated parameters

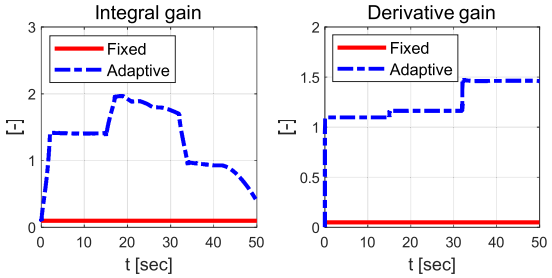

FIGURE 10. Simulation-based results (step target velocity): feedback gains – integral (left) and derivative (right).

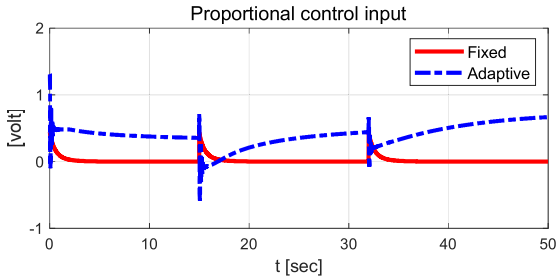

FIGURE 11. Simulation-based results (step target velocity): proportional control input - fixed and adaptive.

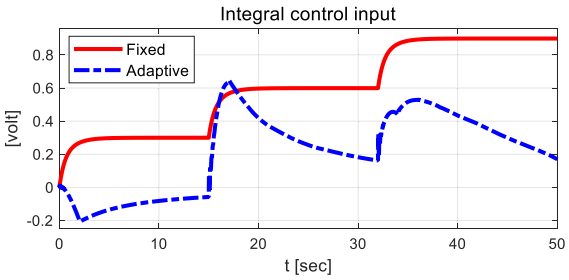

FIGURE 12. Simulation-based results (step target velocity): integral control input - fixed and adaptive.

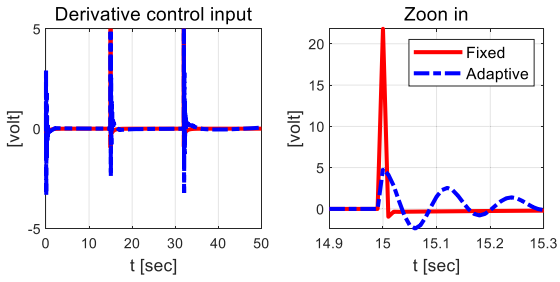

FIGURE 13. Simulation-based results (step target velocity): derivative control input - fixed and adaptive.

used to calculate the proportional control input and integral and derivative gains. Fig. 10 shows the computed integral and derivative gains obtained using the estimated parameters. Figs. 11-13 shows the proportional, integral, and derivative control inputs, respectively.

- Sinusoidal target velocity (simulation)

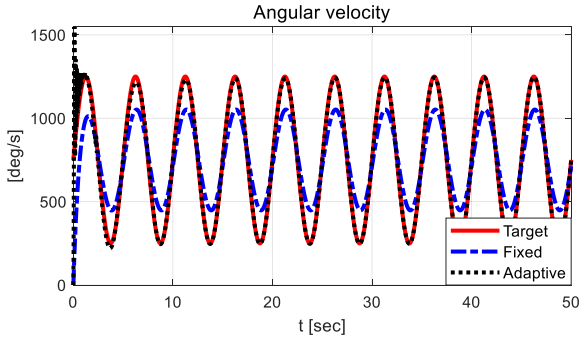

FIGURE 14. Simulation-based results (sinusoidal target velocity): angular velocity - target, fixed, adaptive.

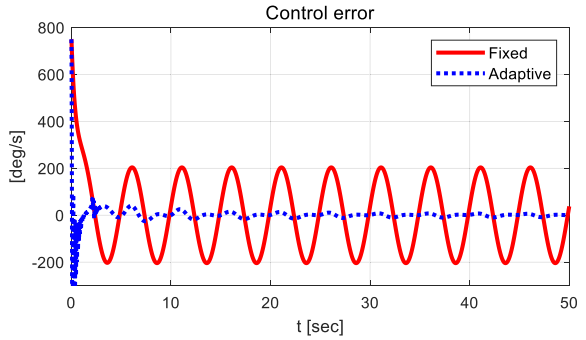

FIGURE 15. Simulation-based results (sinusoidal target velocity): control error - fixed, adaptive.

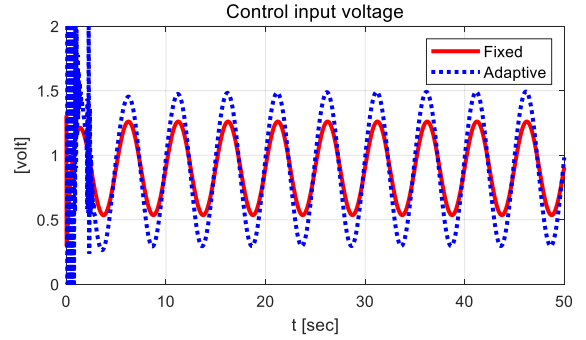

FIGURE 16. Simulation-based results (sinusoidal target velocity): control input voltage - fixed, adaptive.

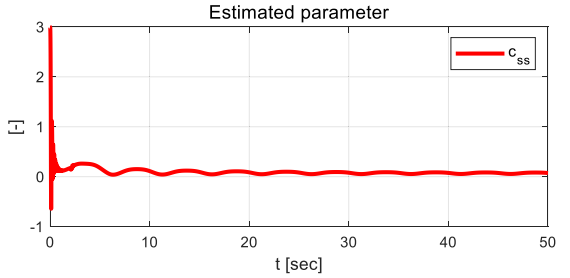

FIGURE 17. Simulation-based results (sinusoidal target velocity): estimated parameter - c_{ss}.

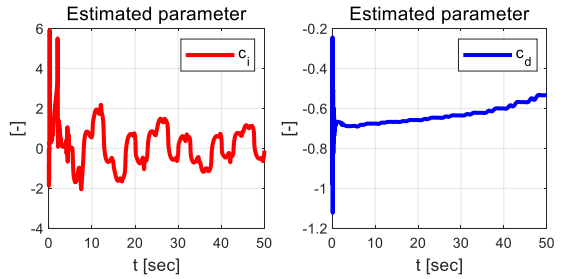

FIGURE 18. Simulation-based results (sinusoidal target velocity): estimated parameter - c_i (left) and c_d (right).

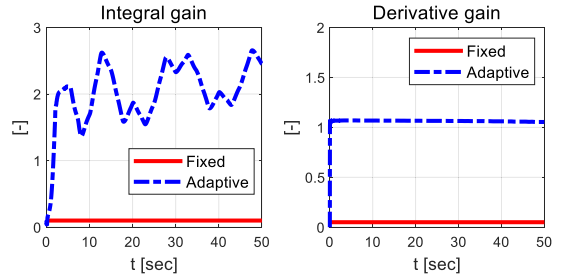

FIGURE 19. Simulation-based results (sinusoidal target velocity): feedback gains – integral (left), derivative (right).

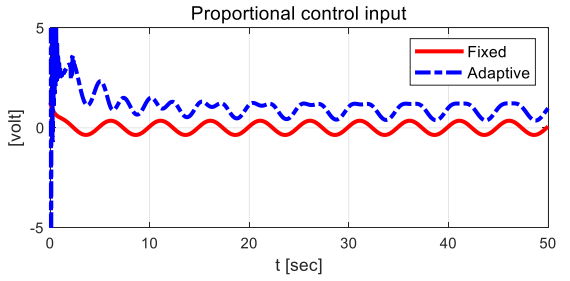

FIGURE 20. Simulation-based results (sinusoidal target velocity): proportional control input - fixed, adaptive.

Figs. 14–22 show the simulation-based performance evaluation results when a sinusoidal target velocity was applied as the target state.

As shown in Figs. 14 and 15, the tracking result in the adaptive case shows better performance than that in the fixed case. The result for the fixed case showed that there was a relatively large tracking error (maximum ∼200 deg/s). Fig. 16 shows the applied control inputs for each case. Figs. 17 and 18 show the estimated parameters for calculating the proportional control input and the integral and derivative gains. Fig. 19 shows the computed integral and derivative gains online using the estimated parameters. Figs. 20-22 show the proportional, integral, and derivative control inputs. The magnitude of the integral control input in the adaptive case was gradually reduced by the designed adaptation rule. Table 3 summarizes the tracking control error in the evaluation results.

The RMS values from the evaluation results when the proposed control algorithm was used were smaller than those in the fixed case.

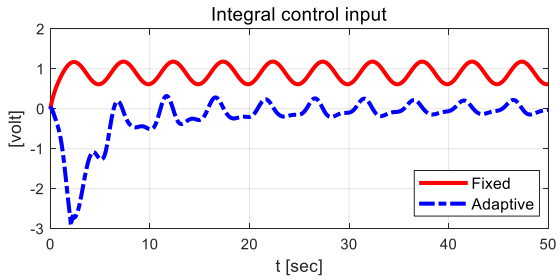

FIGURE 21. Simulation-based results (sinusoidal target velocity): integral control input - fixed, adaptive.

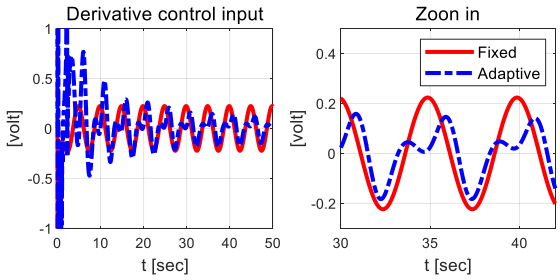

FIGURE 22. Results (sinusoidal target velocity): derivative control input - fixed, adaptive.

TABLE 3. Tracking error summary of evaluation results (simulation).

Division	Tracking control error (RMS value, deg/s)	
	Step target velocity (Figure 6)	Sinusoidal target velocity (Figure 15)
Fixed case	34.1036	158.1544
Adaptive case	12.8996	37.7100

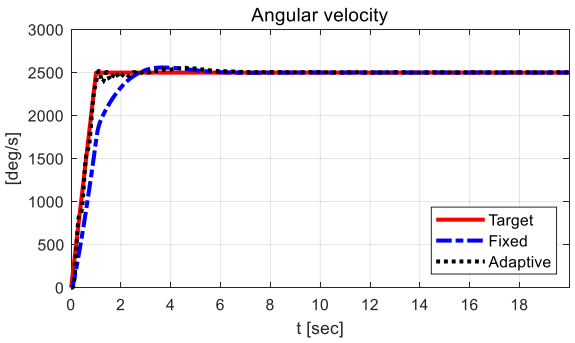

FIGURE 23. Actual test-platform-based results (step target velocity): angular velocity - target, fixed, adaptive.

B. ACTUAL TEST-PLATFORM-BASED EVALUATION

- Step target velocity (using actual test platform)

Figs. 23–32 show the actual test-platform-based evaluation results when a sinusoidal target velocity was applied.

As shown in Figs. 23 and 24, the proposed control algorithm (in the adaptive case) allowed the DC motor to track the step target velocity with a relatively small control error. However, the tracking result in the fixed case showed a relatively

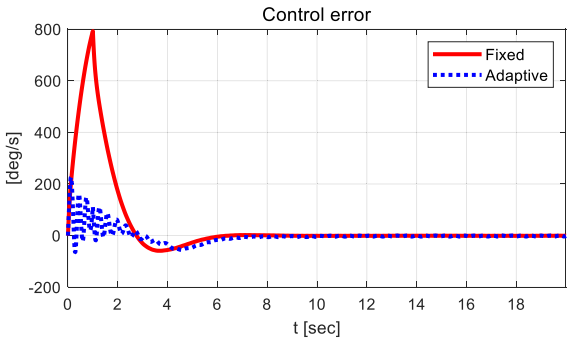

FIGURE 24. Actual test-platform-based results (step target velocity): control error - fixed, adaptive.

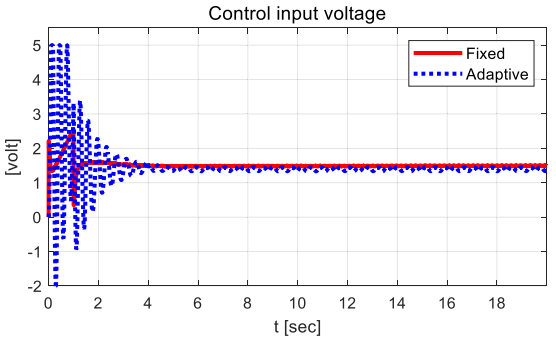

FIGURE 25. Actual test-platform-based results (step target velocity): control input voltage - fixed, adaptive.

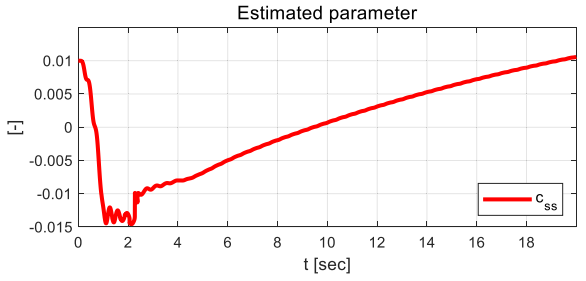

FIGURE 26. Actual test-platform-based results (step target velocity): estimated parameter - c_{ss}.

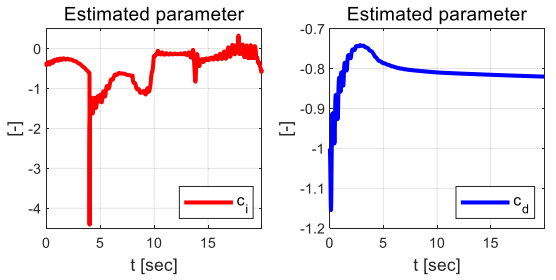

FIGURE 27. Actual test-platform-based results (step target velocity): estimated parameter - c_i (left) and c_d (right).

large tracking error (maximum ∼900 deg/s) in the transient region. The applied control inputs are shown in Fig. 25; there was a relatively large oscillation in the adaptive case early in

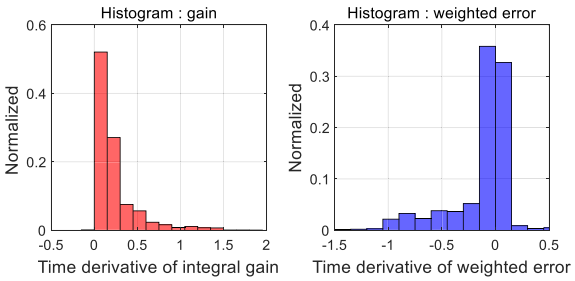

FIGURE 28. Actual test-platform-based results (step target velocity): normalized histogram – gain (left) and weighted error (right).

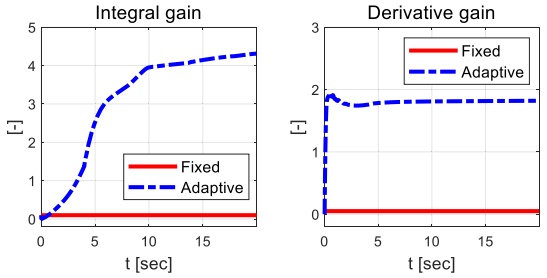

FIGURE 29. Actual test-platform-based results (step target velocity): feedback gains – integral (left), derivative (right).

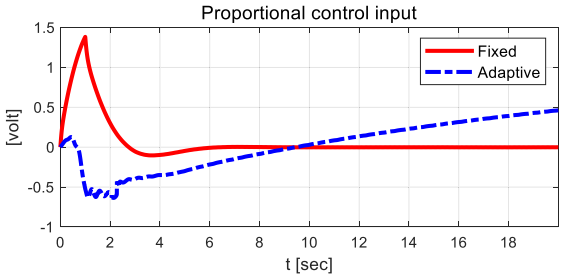

FIGURE 30. Actual test-platform-based results (step target velocity): proportional control input – fixed, adaptive.

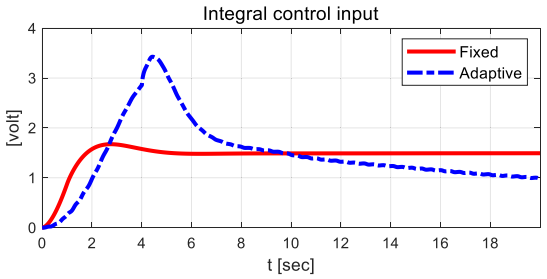

FIGURE 31. Actual test-platform-based results (step target velocity): integral control input – fixed, adaptive.

the evaluation. Figs. 26 and 27 show the estimated parameters for calculating the proportional control input and the integral and derivative gains. Fig. 28 shows the normalized histograms of derivatives of integral gain and weighted error with respect to time. Fig. 29 shows the calculated integral and derivative gains obtained using the estimated real-time parameters. Figs. 30-32 show the proportional, integral, and

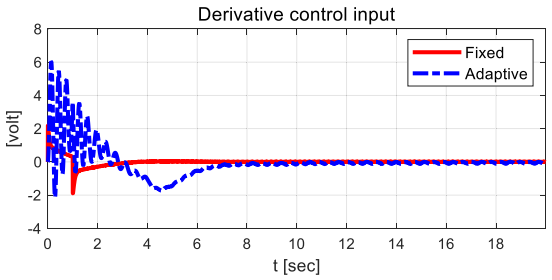

FIGURE 32. Actual test-platform-based results (step target velocity): derivative control input – fixed, adaptive.

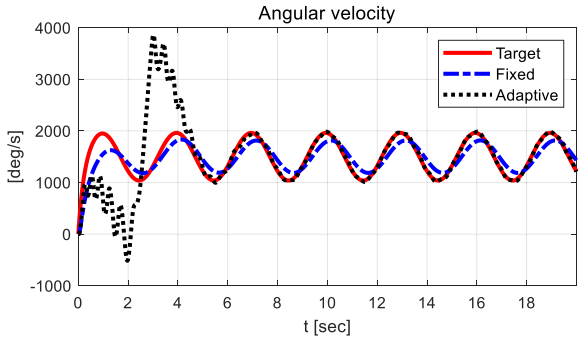

FIGURE 33. Actual test-platform-based results (sinusoidal target velocity): angular velocity - target, fixed, adaptive.

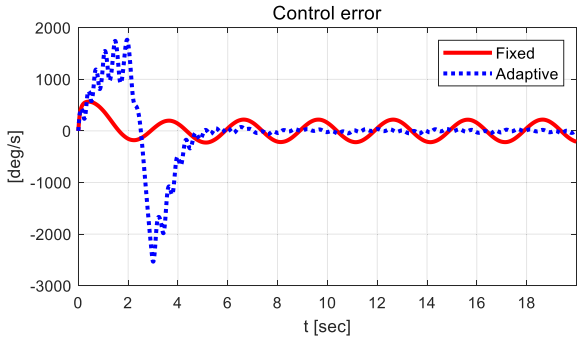

FIGURE 34. Actual test-platform-based results (sinusoidal target velocity): control error - fixed, adaptive.

derivative control inputs. The value of the integral control input in the adaptive case was found to be greater than that in the fixed case.

- Sinusoidal target velocity (using actual test platform)

Figs. 33–42 show the actual test-platform-based evaluation results when a sinusoidal target velocity was applied.

As shown in Figs. 33 and 34, the DC motor could track the sinusoidal target velocity with a relatively small control error after adaptation (after approximately 5 s). However, the tracking result in the fixed case showed a relatively large tracking error (maximum ∼200 deg/s) in the steady-state region. Fig. 35 shows the applied control inputs; as shown, that there was a relatively large oscillation resulting from adaptation in the adaptive case early in the evaluation. Figs. 36 and 37 show the estimated parameters for

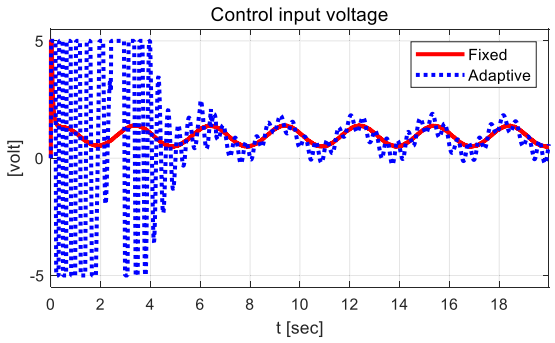

FIGURE 35. Actual test-platform-based results (sinusoidal target velocity): control input voltage - fixed, adaptive.

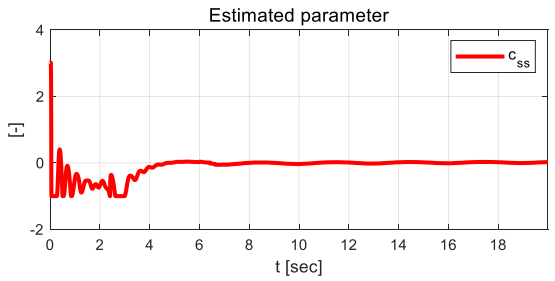

FIGURE 36. Actual test-platform-based results (sinusoidal target velocity): estimated parameter - c_{ss}.

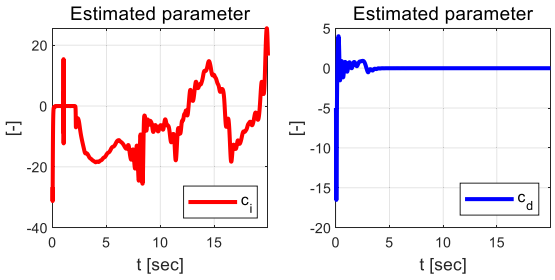

FIGURE 37. Actual test-platform-based results (sinusoidal target velocity): estimated parameter - c_i (left) and c_d (right).

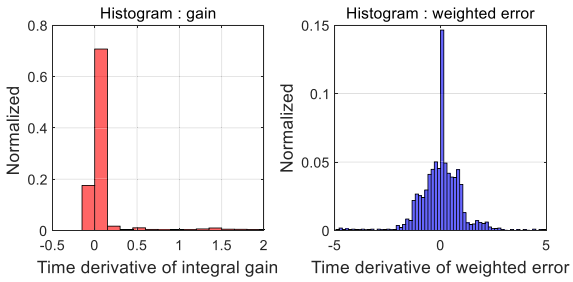

FIGURE 38. Actual test-platform-based results (sinusoidal target velocity): normalized histogram – gain (left) and weighted error (right).

calculating the proportional control input and the integral and derivative gains. Fig. 38 shows the normalized histograms of derivatives of integral gain and weighted error with respect to time. Fig. 39 shows the integral and derivative gains calculated using the estimated parameters. Figs. 40-42 show

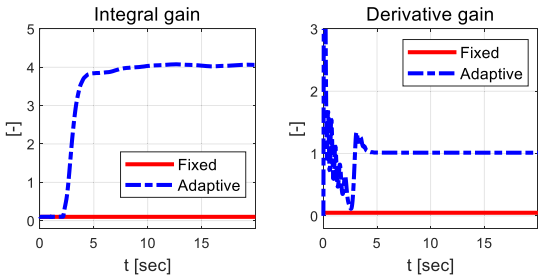

FIGURE 39. Actual test-platform-based results (sinusoidal target velocity): feedback gains - integral (left), derivative (right).

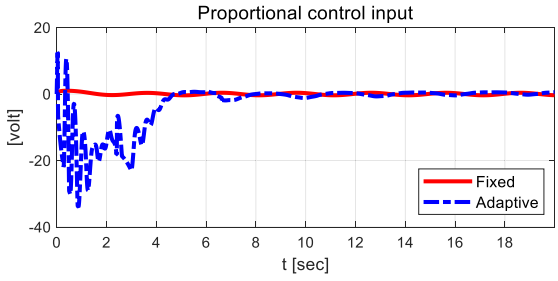

FIGURE 40. Actual test-platform-based results (sinusoidal target velocity): proportional control input - fixed, adaptive.

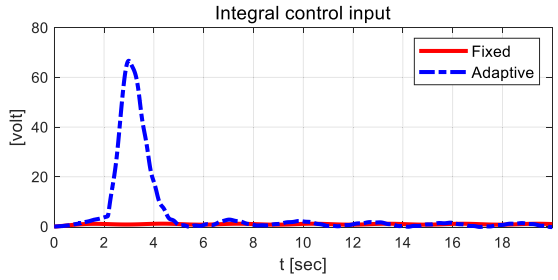

FIGURE 41. Actual test-platform-based results (sinusoidal target velocity): integral control input - fixed, adaptive.

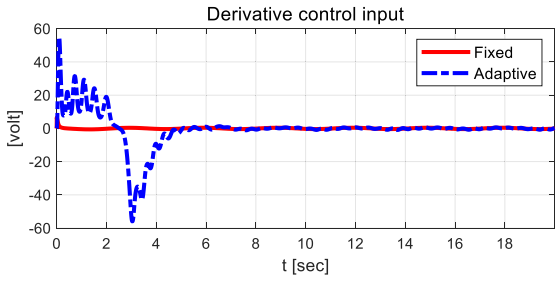

FIGURE 42. Actual test-platform-based results (sinusoidal target velocity): derivative control input - fixed, adaptive.

the proportional, integral, and derivative control inputs. The value of integral control input in the adaptive case was more than that in the fixed case, similar to the case of the step target velocity-based evaluation result. Table 4 summarizes the tracking control error of the evaluation results after an evaluation time of 10 s.

TABLE 4. Tracking error summary of evaluation results (actual test).

Division	Tracking control error (RMS value, deg/s)	
	Step target (Figure 24)	Sinusoidal target (after 5 s) (Figure 34)
Fixed case	156.0650	154.5177
Adaptive case	30.8712	25.3317

The RMS values from the step target velocity-based evaluation results in the adaptive case were smaller than those in the fixed case. However, for the sinusoidal target velocity-based evaluation results, the RMS value in the adaptive case was larger than that in the fixed case because of the adaptation process early in the evaluation.

V. CONCLUSION

This study proposed and evaluated a new data-driven adaptive SS-ID control algorithm for target state tracking using a simulation technique and an actual test platform. The steady-state proportional control input was derived using simplified dynamics based on the estimated coefficient. With the computed integral and derivative gains, the data-driven adaptation rule and RLS-based estimation methods were used with the performance conditions. The control inputs were designed to be derived using the computed gains in real time. Based on the steady-state, integral, and derivative control inputs, the total SS-ID control input was computed and used as a control input for target state tracking. A performance evaluation was conducted based on the simulation technique and an actual test platform using a DC motor for evaluation. The results indicate that the DC motor could effectively track the target velocity using the proposed data-driven adaptive SS-ID control algorithm. However, the proposed adaptive control algorithm has certain limitations. First, the assumptions for the steady-state control input are for the limited condition that the derivative of the state x with respect to time t, the control inputs u_I and u_D are near zero. Therefore, improvement of steady-state control method is considered as a future work to cope with the condition that the derivative of the state x with respect to time t, the control inputs u_I and u_D are not zero. Second, the determination of appropriate initial parameters is necessary for reasonably stable performance in the early stage of evaluation. Third, the convergence time for robust performance has not been ensured. Therefore, the improvement of the controller's adaptation rule to achieve more robust performance and evaluation under various scenarios are considered as future work, and it is also planned that the finite-time convergence condition will be applied to the proposed control algorithm based on the sliding-mode approach. Nevertheless, the proposed adaptive controller does not require any system information or a mathematical model, and it is expected that the data-driven adaptive control algorithm can be used as a target tracking control algorithm for various complex systems that have relatively high nonlinearity.

REFERENCES

[1] L. Liu, L. Zhang, G. Pan, and S. Zhang, "Robust yaw control of autonomous underwater vehicle based on fractional-order PID controller," *Ocean Eng.*, vol. 257, Aug. 2022, Art. no. 111493.

[2] Z. Wang, G. Yi, and S. Zhang, "An improved fuzzy PID control method considering hydrogen fuel cell voltage-output characteristics for a hydrogen vehicle power system," *Energies*, vol. 14, no. 19, p. 6140, Sep. 2021.

[3] D. X. Phu and S. B. Choi, "A new adaptive fuzzy PID controller based on Riccati-like equation with application to vibration control of vehicle seat suspension," *Appl. Sci.*, vol. 9, no. 21, p. 4540, Oct. 2019.

[4] J. Guerrero, J. Torres, V. Creuze, A. Chemori, and E. Campos, "Saturation based nonlinear PID control for underwater vehicles: Design, stability analysis and experiments," *Mechatronics*, vol. 61, pp. 96–105, Aug. 2019.

[5] R. K. Mandava and P. R. Vundavilli, "An adaptive PID control algorithm for the two-legged robot walking on a slope," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3407–3421, Apr. 2020.

[6] A. Noordin, M. A. Mohd Basri, and Z. Mohamed, "Position and attitude tracking of MAV quadrotor using SMC-based adaptive PID controller," *Drones*, vol. 6, no. 9, p. 263, Sep. 2022.

[7] A. Noordin, M. A. M. Basri, Z. Mohamed, and I. M. Lazim, "Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization," *Arabian J. Sci. Eng.*, vol. 46, no. 2, pp. 963–981, Feb. 2021.

[8] X. Guo and C. K. Ahn, "Adaptive fault-tolerant pseudo-PID sliding-mode control for high-speed train with integral quadratic constraints and actuator saturation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7421–7431, Dec. 2021.

[9] G. Zhong, C. Wang, and W. Dou, "Fuzzy adaptive PID fast terminal sliding mode controller for a redundant manipulator," *Mech. Syst. Signal Process.*, vol. 159, Oct. 2021, Art. no. 107577.

[10] D. G. Lui, A. Petrillo, and S. Santini, "Leader tracking control for heterogeneous uncertain nonlinear multi-agent systems via a distributed robust adaptive PID strategy," *Nonlinear Dyn.*, vol. 108, no. 1, pp. 363–378, Mar. 2022.

[11] C. Yang, F. Yao, M. Zhang, Z. Zhang, Z. Wu, and P. Dan, "Adaptive sliding mode PID control for underwater manipulator based on Legendre polynomial function approximation and its experimental evaluation," *Appl. Sci.*, vol. 10, no. 5, p. 1728, Mar. 2020.

[12] C. Zhang, X. Wu, and J. Xu, "Particle swarm sliding mode-fuzzy PID control based on maglev system," *IEEE Access*, vol. 9, pp. 96337–96344, 2021.

[13] T. Kobaku, R. Poola, and V. Agarwal, "Design of robust PID controller using PSO-based automated QFT for nonminimum phase boost converter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 12, pp. 4854–4858, Dec. 2022.

[14] M. J. Mahmoodabadi and N. Nejadkourki, "Optimal fuzzy adaptive robust PID control for an active suspension system," *Austral. J. Mech. Eng.*, vol. 20, no. 3, pp. 681–691, May 2022.

[15] B. G. Kavyashree, S. Patil, and V. S. Rao, "Observer-based anti-windup robust PID controller for performance enhancement of damped outrigger structure," *Innov. Infrastruct. Solutions*, vol. 7, no. 3, p. 205, Jun. 2022.

[16] P. R. Jeyaraj and E. R. S. Nadar, "Real-time data-driven PID controller for multivariable process employing deep neural network," *Asian J. Control*, vol. 24, no. 6, pp. 3240–3251, Nov. 2022.

[17] H. Yu, Z. Guan, T. Chen, and T. Yamamoto, "Design of data-driven PID controllers with adaptive updating rules," *Automatica*, vol. 121, Nov. 2020, Art. no. 109185.

[18] S. Makarem, B. Delibas, and B. Koc, "Data-driven tuning of PID controlled piezoelectric ultrasonic motor," *Actuators*, vol. 10, no. 7, p. 148, Jun. 2021.

[19] J. Li and T. Yu, "A novel data-driven controller for solid oxide fuel cell via deep reinforcement learning," *J. Cleaner Prod.*, vol. 321, Oct. 2021, Art. no. 128929.

[20] N. Zhu, X.-T. Gao, and C.-Q. Huang, "A data-driven approach for on-line auto-tuning of minimum variance PID controller," *ISA Trans.*, vol. 130, pp. 325–342, Nov. 2022.

[21] M. Rahmani and S. Redkar, "Data-driven Koopman fractional order PID control of a MEMS gyroscope using bat algorithm," *Neural Comput. Appl.*, vol. 2023, pp. 1–10, Jan. 2023.

[22] T. Shuprajhaa, S. K. Sujit, and K. Srinivasan, "Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes," *Appl. Soft Comput.*, vol. 128, Oct. 2022, Art. no. 109450.

[23] W. Wang, T. Chai, H. Wang, and Z. Wu, "Signal-compensation-based adaptive PID control for fused magnesia smelting processes," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9441–9451, Sep. 2023.
[24] M. S. Saad, H. Jamaluddin, and I. Z. M. Darus, "Implementation of PID controller tuning using differential evolution and genetic algorithms," *Int. J. Innov. Comput., Inf. Control*, vol. 8, no. 11, pp. 7761–7779, 2012.
[25] M. Zamani, M. Karimi-Ghartemani, N. Sadati, and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Eng. Pract.*, vol. 17, no. 12, pp. 1380–1387, Dec. 2009.
[26] C.-F. Hsu and B.-K. Lee, "FPGA-based adaptive PID control of a DC motor driver via sliding-mode approach," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11866–11872, Sep. 2011.

JEONGWOO LEE received the B.S. and M.S. degrees in mechanical engineering from Pusan National University, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in mechanical and aerospace engineering from Seoul National University, Seoul, in 2020. His research interests include integrated chassis control systems, electrified active safety systems, and motion control systems for automated driving of ground vehicles.

KWANGSEOK OH (Member, IEEE) received the B.S. degree in mechanical engineering from Hanyang University, Seoul, in 2009, and the M.S. and Ph.D. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, in 2013 and 2016, respectively. From 2016 to 2017, he was an Assistant Professor with the Automotive Engineering Department, Honam University. Since 2017, he has been an Associate Professor with the School of ICT, Robotics & Mechanical Engineering, Hankyong National University, Anseong-si, South Korea. His research interests include fail-safe systems for autonomous driving, adaptive and predictive control, and intelligent embedded system control.