JJEE
Volume 11 | Number 4 | Pages 656-669

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Synergy of Discrete Sliding Mode Control and Online Recursive Least Squares Estimation for DC Motor Applications

Nhut Thang Le[1], Cong Toai Truong[2], Minh Tri Nguyen[3], Huy Hung Nguyen[4], Van Tu Duong[5*], Tan Tien Nguyen[6]

[1,2,3,5,6] Key Laboratory of Digital Control and System Engineering (DCSELab), Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology, Vietnam.

[1,2,3,5,6] Vietnam National University Ho Chi Minh City, Linh Xuan Ward, Ho Chi Minh City 700000, Vietnam.

[4] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City 700000, Vietnam.

Email: dvtu@hcmut.edu.vn

Received: Aug 12, 2025 *Revised: Oct 22, 2025* *Accepted: Nov 14, 2025* *Available online: Jan 20, 2026*

***Abstract*—** This paper presents a control algorithm for single-input single-output systems with time-varying model parameters based on the integration of sliding mode control (SMC) and recursive least squares (RLS). The proposed algorithm is evaluated through simulations on a virtual direct current motor under abrupt parameter changes at $t = 10$ seconds and $t = 20$ seconds. Without RLS, the system output fails to converge to the desired velocity, while the presence of RLS reduces the error but results in slow convergence. Hence, the influence of control parameters, weighting coefficients (c), and forgetting factor (λ), along with their interaction, was analyzed. Specifically, reducing λ to 0.99 and increasing the SMC gain c up to 5 improves the convergence speed but introduces significant overshoot (up to 150 rpm). For this reason, a damping function is proposed and incorporated into the control signal. Simulation results show that the proposed controller completely eliminates overshoots at the initial time ($t < 2$ s), reduces settling time to under 2 seconds after each model change, and maintains steady-state errors within $|e| < 2$ rpm despite input disturbance in the range $[-0.1, 0.1]$ voltage. The overshoot at $t = 10$ seconds and $t = 20$ seconds is reduced to 80 rpm and 114 rpm, respectively, without causing instability. As a result, it confirms the effectiveness of the proposed method in achieving fast, robust, and smooth tracking performance under parameter uncertainties.

***Keywords*—** Sliding mode control; Discrete time; Recursive Least Squares; Hybrid control.

1. INTRODUCTION

In the era of dramatic advancements in robotics and intelligent control systems, robust and adaptive control strategies have significantly more requirements than ever [1]. In reality, these control approaches are applied in various fields of control applications [2-5], including the cleaning robot in the photovoltaic panels system [6, 7], the mobile robot [8, 9], the mechanical ventilator in the healthcare system[10-16] , etc. However, a significant challenge in practical control systems is the presence of system uncertainties and various noises, both of which can severely degrade the performance of traditional controllers. These uncertainties may arise from modeling errors, environmental fluctuations, unknown system dynamics, etc. Hence, there is a demand for control algorithms that can dynamically adapt to such variations while maintaining reliable operation.

* Corresponding author
DOI: https://doi.org/10.5455/jjee.204-1755003775

Accordingly, numerous control strategies have been developed to address the issues posed by uncertain or time-varying models. Among them, sliding mode control (SMC) is considered popular due to its inherent robustness against matched uncertainties and external disturbances [17]. On the other hand, conventional SMC is primarily designed in the continuous-time domain. When applied in digital systems with discrete-time sampling, classical SMC may exhibit undesirable behaviors such as chattering or control instability due to the mismatch between discrete signals and continuous design assumptions [18]. To overcome these limitations, discrete sliding mode control (DSMC) has been proposed as a powerful alternative [19, 20]. Unlike conventional SMC, DSMC considers the discrete nature of sampled signals and control inputs, leading to a more realistic and accurate control action in digital implementations [21]. DSMC modifies the sliding surface dynamics and control laws to operate in the discrete-time domain, mitigating chattering and enhancing real-time feasibility [22]. However, the effectiveness of DSMC still heavily depends on an accurate mathematical model of the system. In practice, the system model changes during operation because of disturbances, nonlinear factors, and external influences; therefore, the fixed-parameter design of DSMC became insufficient, prompting the need for adaptive solutions.

To address this issue, recent studies have explored integrating DSMC with online parameter estimation algorithms to achieve adaptive control in uncertain environments. One potential approach involves using the recursive least squares (RLS) algorithm for real-time system identification. RLS is a recursive algorithm that estimates system parameters by minimizing the cumulative squared error, providing fast convergence and high adaptability to changing dynamics [23]. In combination with DSMC, RLS can continuously update the control model, enabling the controller to adjust its actions based on the most recent system behavior. Several researchers have successfully demonstrated the effectiveness of combining RLS with sliding mode controllers. Actually, Kim et al. introduced a self-tuning sliding mode control method integrated with an RLS estimator for position tracking in direct current (DC) motor drives. The approach enhances tracking performance under varying load inertia by adapting the control input based on real-time parameter estimation. It improves energy efficiency by adjusting the input voltage according to updated system dynamics and demonstrates strong robustness against model uncertainties through hardware experiments. Nevertheless, the method requires careful tuning of the RLS forgetting factors and sufficient computational capacity, which may limit its applicability in low-resource embedded systems [24]. Additionally, Wang et al. proposed an adaptive terminal sliding mode controller augmented by an RLS-based parameter identifier for servo systems experiencing inertia variation. The controller achieves finite-time convergence and improved tracking accuracy by continuously updating model parameters in real-time. Through comprehensive simulations, the method demonstrates enhanced disturbance rejection and adaptability under parameter shifts. Nevertheless, it has only been validated in simulation environments, and its effectiveness may degrade under abrupt transitions or high-frequency disturbances if the characteristic model or estimation fails to capture rapid dynamics [25]. In addition, Sguarezi Filho et al. introduced a sliding mode control approach integrated with an RLS algorithm for voltage compensation in three-phase loads. The method enhances voltage regulation under sag conditions, improves energy efficiency by relying on steady-state operation, and demonstrates reliability across various scenarios through simulations and experiments. Nevertheless, it demands precise parameter tuning and consistent system monitoring and may have reduced adaptability during

transient conditions [26]. As a result, these studies confirm the value of integrating parameter estimation into sliding mode frameworks. However, many existing approaches are either limited to specific applications or require high computational resources, making them less suitable for systems with tight real-time constraints. Furthermore, Oh et al. [27] employed RLS to tune the switching gain for continuous SMC, aiming to balance robustness and energy efficiency. Valladolid et al. [28] combined RWLS–QSMC for steady-state error reduction, while Znidi et al. [29] used Matrix-RLS in discrete SMC to improve parameter convergence. However, these approaches typically use RLS as an auxiliary estimator rather than integrating it directly into the control law, and they do not explicitly address the dynamic effects that arise during fast adaptation.

This paper presents a hybrid control algorithm that combines the strengths of discrete sliding mode control and the RLS algorithm for single input-single output (SISO) systems with time-varying dynamics. The main difference lies in using RLS for online AutoRegressive with eXogenous inputs (ARX) model identification and deriving the discrete sliding control law from the updated parameters, creating a fully model-adaptive control scheme. This method is specifically designed for systems whose parameters change during operation, addressing the practical demand for model adaptation in real-time applications. The key idea is to treat the plant as a discrete-time auto-regressive with exogenous input model, where the model coefficients are continuously updated using the RLS estimator. The DSMC controller is then derived using these real-time parameter estimates to compute the control input.

The main contribution of this paper is:

- Proposed an adaptive control algorithm that integrates SMC with RLS for SISO systems with time-varying models.
- Conducted a simulation-based analysis of the effects of parameters c and λ on system settling time and output stability.
- Applied a damping function to the control signal and demonstrated its effectiveness in shortening settling time and eliminating oscillations.
- Verified the superior adaptability and performance of the proposed algorithm compared to a fixed-parameter PID controller under model variations.

Subsequent to the introductory background, the paper is organized as follows: Section II details the formulation of the ARX model and the implementation of the online RLS algorithm for real-time parameter estimation. Additionally, it presents the design of the DSMC based on the identified parameters and discusses the integration of a damping function into the control signal. Section III describes the simulations using a DC motor model in software-in-the-loop, including scenarios with time-varying parameters and disturbances. Alongside that, it analyses the simulation results, comparing the proposed algorithm's performance with that of a conventional PID controller in terms of settling time, overshoot, and steady-state error. Finally, Section IV concludes the paper by summarizing the main findings, outlining the limitations, and suggesting directions for future experimental validation and optimization.

2. DC MOTOR MODELING AND CONTROL METHODOLOGY

2.1. RLS-Based Online Identification

The DC motor is modeled as a SISO system represented by an ARX structure, where the input polynomial terms are derived from $u(k-1)$ and preceding time steps, represented by

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m), \quad (1)$$

where $a_1, \ldots, a_n$ $(n \in \mathbb{R})$ represent the coefficients of the past output polynomial, and $b_1, \ldots, b_m$ $(m \in \mathbb{R})$ correspond to those of the past input.

Define the parameter vector $\boldsymbol{\theta}$ as the set of coefficients associated with the system's autoregressive and exogenous components, i.e., $\boldsymbol{\theta} = [a_1 \quad \ldots \quad a_n \quad b_1 \quad \ldots \quad b_m]^T$ $(\boldsymbol{\theta} \in \mathbb{R}^{n+m})$. The regressor vector $\boldsymbol{\phi}$, consisting of time-shifted samples of the input and output signals, is expressed as $\boldsymbol{\phi}(k-1) = [-y(k-1) \quad \ldots \quad -y(k-n) \quad u(k-1) \quad \ldots \quad u(k-m)]^T$ $(\boldsymbol{\phi} \in \mathbb{R}^{n+m})$. Substituting these defined symbols into Eq. (1), the output at time k is determined as follows:

$$y(k) = \boldsymbol{\phi}^{\boldsymbol{T}}(k-1)\boldsymbol{\theta}. \quad (2)$$

Assuming that $\boldsymbol{\phi}$ can be determined in practice, the goal of the algorithm is to predict $\boldsymbol{\theta}$ based on the recorded data of $\boldsymbol{\phi}$. Let $\widehat{\boldsymbol{\theta}}$ be the prediction vector of $\boldsymbol{\theta}$, then the output from the prediction model at time k is given by: $\hat{y}(k) = \boldsymbol{\phi}^{\boldsymbol{T}}(k-1)\widehat{\boldsymbol{\theta}}(k-1)$. Let J be a cost function of the form of the cumulative square of the error described as follows:

$$J(k) = \sum_{i=1}^{k} \lambda^{k-i} \left(y(i) - \boldsymbol{\phi}^{\boldsymbol{T}}(i-1)\widehat{\boldsymbol{\theta}}(k) \right)^2, \quad (3)$$

where $\lambda = (0,1]$ is the forgetting factor, which regulates the weight of past data in the estimation process. The update mechanism for $\widehat{\boldsymbol{\theta}}$ is derived based on the minimization of a cost function J. Let $\boldsymbol{P}(k) = \left[\sum_{i=1}^{k} \lambda^{k-i} \boldsymbol{\phi}(i-1)\boldsymbol{\phi}^{\boldsymbol{T}}(i-1)\right]^{-1}$ be the covariance matrix, then $\boldsymbol{P}(k+1) = \left(\lambda \boldsymbol{P}^{-1}(k) + \boldsymbol{\phi}(k)\boldsymbol{\phi}^{\boldsymbol{T}}(k)\right)^{-1}$ is determined by applying a forward time shift to $P(k)$. The equation defining $\widehat{\boldsymbol{\theta}}(k+1)$ is based on the derivative from Eq. (3) and $\boldsymbol{P}(k+1)$ is defined by:

$$\widehat{\boldsymbol{\theta}}(k+1) = \widehat{\boldsymbol{\theta}}(k) + \boldsymbol{P}(k+1)\boldsymbol{\phi}(k)\varepsilon(k+1), \quad (4)$$

According to Sherman-Morrison [30], the covariance matrix at time $k+1$ is rewritten as:

$$\boldsymbol{P}(k+1) = \frac{1}{\lambda}\left[\boldsymbol{P}(k) - \frac{\boldsymbol{P}(k)\boldsymbol{\phi}(k)\boldsymbol{\phi}^{\boldsymbol{T}}(k)\boldsymbol{P}(k)}{\lambda + \boldsymbol{\phi}^{\boldsymbol{T}}(k)\boldsymbol{P}(k)\boldsymbol{\phi}(k)}\right]. \quad (5)$$

For the case where the model is previously undefined, the initial covariance matrix is defined by $\boldsymbol{P}(0) = \sigma \boldsymbol{I}$, where σ is the initial selection coefficient and $\boldsymbol{I}$ is the identity matrix. Besides, the estimated parameters and initial observations are set to zero. By employing the adaptive law defined in Eq. (4), the parameter vector $\widehat{\boldsymbol{\theta}}$ is continuously updated during the operation of the DC motor, enabling the identification model to closely follow the actual system dynamics. However, the computational burden of the RLS algorithm becomes a limiting factor when executed continuously. To address this issue, the present study introduces an error-magnitude-based triggering condition for RLS activation. Specifically, $\widehat{\boldsymbol{\theta}}$ is updated only when the relative error satisfies $(y_d - y)100/y_d > \rho$, where y_d denotes the desired output and ρ represents the triggering threshold. It should be noted that $\widehat{\boldsymbol{\theta}}$ retains its previous value when the RLS computation is suspended due to the threshold ρ. The pseudocode describes the model identification process in Table 1.

2.2. Controller design

By advancing Eq. (1) by one step, the ARX system model is represented by:

$$y(k+1) = A + b_1 u(k). \quad (6)$$

where $A \coloneqq -\sum_{i=1}^{n} a_i y(k+1-i) + \sum_{j=2}^{m} b_j u(k+1-i)$. The control law is designed to ensure that the output of the SISO system y asymptotically tracks the desired output y_d. Let e be the

error between the actual output of the SISO system and the desired output y_d, that expressed at time k is as follows:

$$e(k) = y(k) - y_d. \tag{7}$$

The sliding surface s at k time is formulated as a linear combination of errors, defined as:

$$s(k) = \sum_{i=1}^{n-1} c_i e(k-i) + e(k), \tag{8}$$

where $c_1, \dots, c_{n-1}$ denote the weighting coefficients of the sliding surface, functioning as adjustable parameters in the control design. Taking into account Eqs. (6) and (7), along with the time-shift operation in Eq. (8), the sliding surface at time $k+1$ is determined by $s(k+1) = \sum_{i=1}^{n-1} c_i e(k+1-i) + e(k+1)$, specifically rewritten by:

$$s(k+1) = \sum_{i=1}^{n-1} c_i e(k+1-i) + A - y_d + b_1 u(k). \tag{9}$$

According to Gao's reaching law [31], the dynamic equation of s in terms of the discrete domain is determined by:

$$s(k+1) = \alpha s(k) - \beta \text{sign}\big(s(k)\big), \tag{10}$$

where $\alpha \in (0,1)$ and $\beta > 0$. It is noteworthy that Gao's reaching law has been demonstrated to satisfy the necessary conditions for steering the system into the Quasi-Sliding mode [32]. The control law $u(k)$ is designed such that the sliding surface at the next time point $(k+1)$ has the form of Eq. (10). Substitute Eq. (9) into Eq. (10), the control law $u(k)$ is expressed by:

$$u(k) = \frac{1}{b_1}\left[\alpha s(k) - \beta \text{sign}\big(s(k)\big) - \sum_{i=1}^{n-1} c_i e(k+1-i) - A + y_d\right]. \tag{11}$$

Table 1. The pseudocode of the RLS algorithm.

Step	Command
1	Set initial time $k \leftarrow 1$
2	Compute the covariance matrix $\boldsymbol{P}(2)$ by (5);
3	Repeat
4	$k \leftarrow k+1$
5	$\hat{y}(k) = \boldsymbol{\phi}^{\boldsymbol{T}}(k-1)\widehat{\boldsymbol{\theta}}(k-1)$;
6	Record system's input and output data $\big(\phi(k), y(k)\big)$;
7	$\varepsilon(k) = y(k) - \hat{y}(k)$;
8	$\widehat{\boldsymbol{\theta}}(k) = \widehat{\boldsymbol{\theta}}(k-1) + \boldsymbol{P}(k)\boldsymbol{\phi}(k-1)\varepsilon(k)$ by (4);
9	Compute the covariance matrix $\boldsymbol{P}(k+1)$ by (5);
10	Repeat step 1 if $(y_d - y)100/y_d > \rho$;
11	Suspend RLS computation until $(y_d - y)100/y_d > \rho$;

Fig. 1 illustrates the schematic diagram of a discrete-time sliding mode control system combined with an online RLS algorithm. Specifically, the output y is discretized with a sampling period T, and subsequently compared with the desired output y_d to determine the error e (i.e., the input to the SMC controller). The Zero-Order Hold (ZOH) performs discrete-to-continuous conversion by holding each discrete control input constant over the entire sampling interval, resulting in a piecewise constant signal $u(t)$ for the continuous-time plant. Historical input–output data of the control system are recorded and provided to the RLS block for processing. Based on the discrepancy between the estimated model output $\hat{y}$ and the actual output y, the RLS algorithm updates the model parameters at each sampling interval.

Consequently, the SMC controller coefficients are likewise updated periodically. This continuous adaptation enables the controller to respond effectively to variations in the system dynamics caused by disturbances.

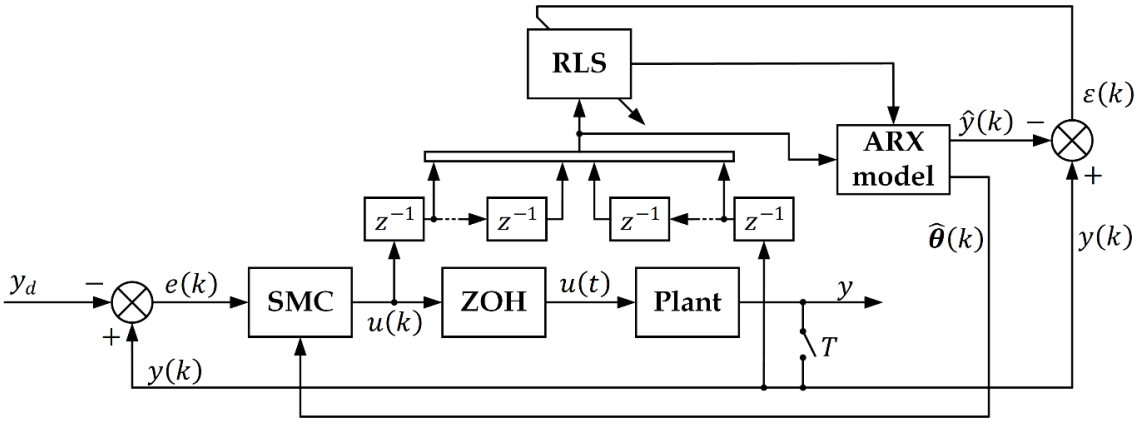

Fig. 1. Schematic diagram of the proposal controller.

3. RESULTS AND DISCUSSION

Since the controller's computation depends on the model parameters, specifying an initial model is essential prior to conducting the simulation. Accordingly, this article uses a multi-physics DC motor model developed in Simscape, with parameters shown in Table 2, and considers a velocity control problem as the case study. The dynamic behavior captured by the Simscape model represents a standard DC motor, with the input as voltage (V) and the output as the motor shaft's angular velocity (rpm). However, the parameters and responses of the virtual model may inadequately represent those of a real physical system. Besides, the Simscape model captures the main system dynamics, while certain real characteristics, such as temperature variation, parameter drift, and unmodeled nonlinearities, are beyond its scope. A simulation of the RLS algorithm using the available block of Matlab Simulink, as shown in Fig. 2, is first performed to determine the initial ARX model of the Simscape DC motor. In which, the Simscape DC motor block in blue represents the physical system, where electrical and mechanical domains are interconnected through voltage, current, torque, and rotational speed signals. The RLS block in black performs online estimation of ARX parameters using the measured input voltage and motor speed. The Regressors block in green generates the delayed input–output data required for the recursive least-squares computation. In this scheme, black lines denote the signal flow within the Simulink computational environment, whereas colored Simscape lines correspond to physical electrical and mechanical connections. The parameter λ is configured as 0.995, and the ARX model is defined as Eq. (1) with $m = 2$ and $n = 1$. The simulation time is set to 1000 seconds with a sampling time of 0.05 seconds. The input voltage signal of the Simscape DC motor generated from the Step block has a value of 10 volts when $t < 500$ seconds and 20 volts at the remaining times. The past data of voltage and velocity are recorded in the Regressors block, which includes a combination of Delay blocks. Following the simulation, the ARX model is identified with $a_1 = -0.747,\ a_2 = 0.242, b = 8.849$, and is expressed as:

$$y(k) = -0.747y(k-1) + 0.242y(k-2) + 8.849u(k-1). \quad (12)$$

This result is assigned to $\widehat{\boldsymbol{\theta}}(1)$ at the initial time when simulating the control algorithm.

Table 2. Parameters of the Simscape DC motor.

Parameter	Value	Unit
Armature inductance	0.0410	mH
Stall torque	15560	mNm
No-load speed	4035	rpm
Rated DC supply voltage	24	V
No-load current	1020	mA
Rotor inertia	1290	cm^2g
Torque at the motor shaft	0.2	Nm

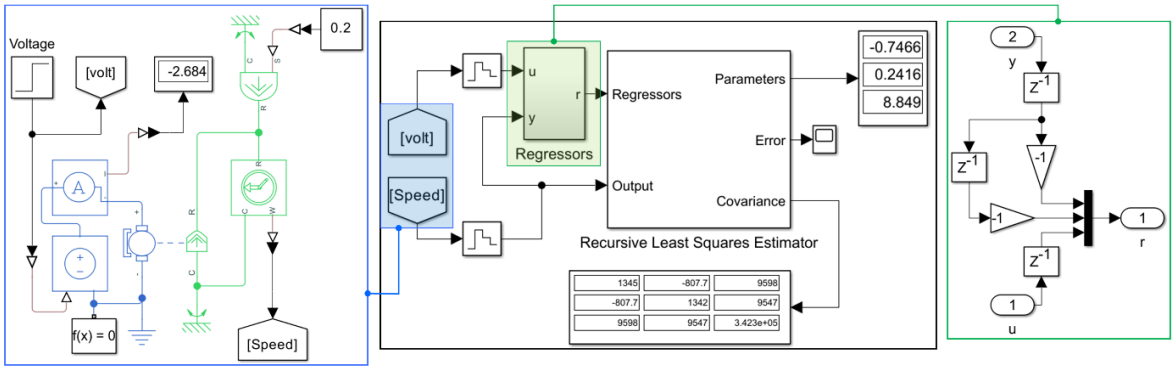

Fig. 2. Simscape DC motor model and block diagram of RLS algorithm simulation using Simulink.

The simulation scenarios are constructed to evaluate the system's adaptability to time-varying conditions for 30 seconds (sampling time is 0.05 seconds). To emulate the effects of system modeling uncertainties, the ARX model used for simulation is varied at two specific time instants, as described in Table 3, instead of being fixed as in Eq. (12). The input disturbance d, generated from a uniform distribution [33] within the range $[-0.1, 0.1]$ voltage, is applied at the control input to reproduce practical imperfections such as voltage ripple, PWM jitter, and electrical noise typically observed in real actuator systems. The trigger threshold of the RLS algorithm (ρ) is defined as 5% for the simulation process. Since the ARX model is predefined, the initial covariance matrix is initialized with a small value $\sigma = 10^{-3}$. The gain parameters of the SMC controller are preliminarily selected as $c = 0.05$, $\beta = 0.01$, and $\alpha = 0.01$. In this study, four simulation scenarios are implemented to evaluate the performance of the proposed controller as described in Table 4.

The forgetting factor λ is simulated with two values: 0.99, offering high sensitivity but low stability, and 0.999, providing better stability with less sensitivity

Fig. 3 depicts the response of the virtual DC motor in two cases, without and with the intervention of the RLS algorithm. In the absence of the RLS algorithm, as shown in Fig. 3a, the system velocity converges around the desired velocity with an error $|e| \leq 1$ when $t < 10$ seconds (i.e. before the time the model is changed). This shows that the SMC controller works effectively if the model parameters are predicted correctly. However, the output is unable to attain the desired velocity beyond the 10th second, specifically, the steady-state errors are -9.5 rpm and -1.5 rpm at $10 \leq t < 20$ seconds and $t \geq 20$ seconds, respectively. This result indicates that the controller is capable of maintaining a stable output but lacks the capacity to eliminate the steady-state errors when the system parameters are mispredicted. In the case of

applying the RLS algorithm with $\lambda = 0.999$, the simulation results show that the output tends to improve the error. Following approximately 10 seconds of response fluctuation, the error is reduced to -4.17 rpm corresponding to the variation at $t = 10$ seconds, and 1.87 rpm for the remaining change. However, the results showed that 10 seconds was insufficient time for the velocity to converge to the desired velocity. This behavior primarily results from two factors: the RLS algorithm exhibits a slow parameter update rate, and the SMC controller requires a prolonged duration to stabilize the system. Consequently, reducing λ can increase the sensitivity of the RLS algorithm, while increasing c contributes to faster convergence of the SMC controller. However, both approaches may potentially lead to instability and induce significant oscillations in the system response.

Table 3. Parameter variation scenarios of the ARX model.

Time [s]	a_1	a_2	b
$t < 10$	-0.747	0.242	8.849
$10 \leq t < 20$	-0.647	0.392	10.949
$t > 20$	-0.767	0.192	7.349

Table 4. Simulation scenarios to evaluate the performance of the proposed controller.

Scenarios	Description
1	Evaluation of the system output with and without RLS intervention, where λ is set to 0.999 and the reference output is 100 rpm.
2	Investigation of the effects of the control parameters $\lambda$and c, considering two cases: $\lambda = 0.99, c = 0.05$ and $\lambda = 0.99, c = 0.5$.
3	Comparison of the proposed controller performance with that of a PID controller designed using the Ziegler–Nichols method.
4	Evaluation of the system output response under varying reference signals and external disturbance effects.

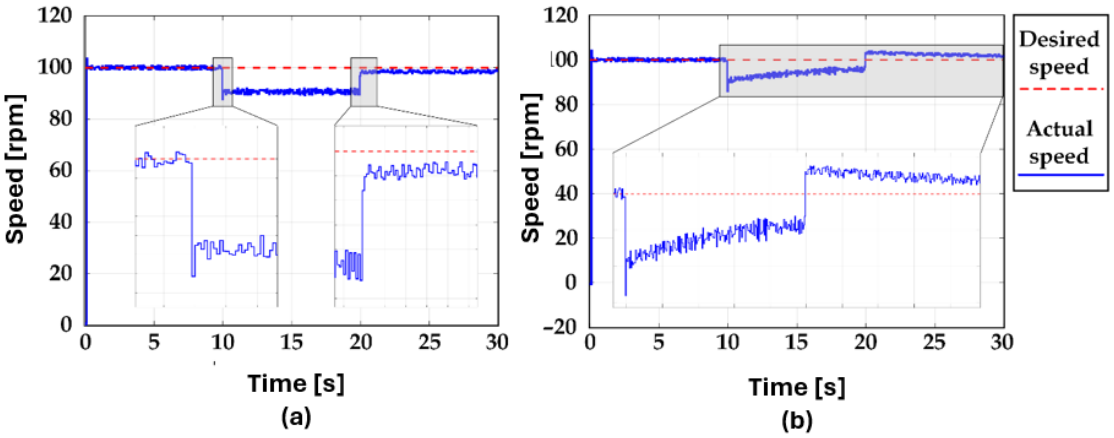

Fig. 3. Speed response of the system is controlled by SMC in case of a) without combination; and b) combined with RLS algorithm (λ = 0.999).

Fig. 4 illustrates the output signal controlled by the SMC combined with the RLS algorithm under two scenarios: (Fig. 4a) reducing λ only, and (Fig. 4b) reducing λ while simultaneously increasing c. In the case of Fig. 4a, λ is reduced to 0.99 to increase the sensitivity of the RLS algorithm while c is kept at 0.05. The overshoot percentages in this case are 3.12%, 15.24% and 7.26% for the initialization, first and last model changes, respectively. The results indicate that the error drops to -1.5 rpm within 7 seconds following the initial fluctuation, and further declines to -0.5 rpm within 5 seconds from the final variation

onward. Besides, the velocity tends to approach the target velocity, but the adjustment occurs at a significantly diminished rate after the previous rapid drop. However, the response time is insufficiently fast for the control system to converge within 10 seconds, as shown in Fig. 3. In the case of combining both λ and c as shown in Fig. 4b, the results show that the controlled steady-state error converges to zero, and the overshoot percentage is controlled by 50% at the initial time, 25% and 6.25% for the model changes, respectively. However, the system's output signal fluctuates significantly, especially at the initial time (speed increases to 150 rpm). This is because the increase in c leads to large and sudden changes in the control signal. Additionally, the system exhibits a response time of about 10 seconds for the initial variation and about 3 seconds for the subsequent ones. This indicates that the response time has not significantly improved, primarily due to the emergence of strong oscillations despite the enhanced speed of the algorithm.

To solve this problem, this study proposes incorporating a damping function, implemented as a first-order filter, into the control signal as follows:

$$u_{damp}(k) = (1-\gamma)u(k) + \gamma u(k-1), \quad (13)$$

where γ is a confidence coefficient with a value in the range (0,1). Eq. (13) incorporates the influence of the previous control signal $u(k-1)$ into the current signal $u(k)$, thereby smoothing the control input and mitigating abrupt changes in $u(k)$. A larger γ smooths the control signal with more delay, while a smaller γ allows faster response but less damping.

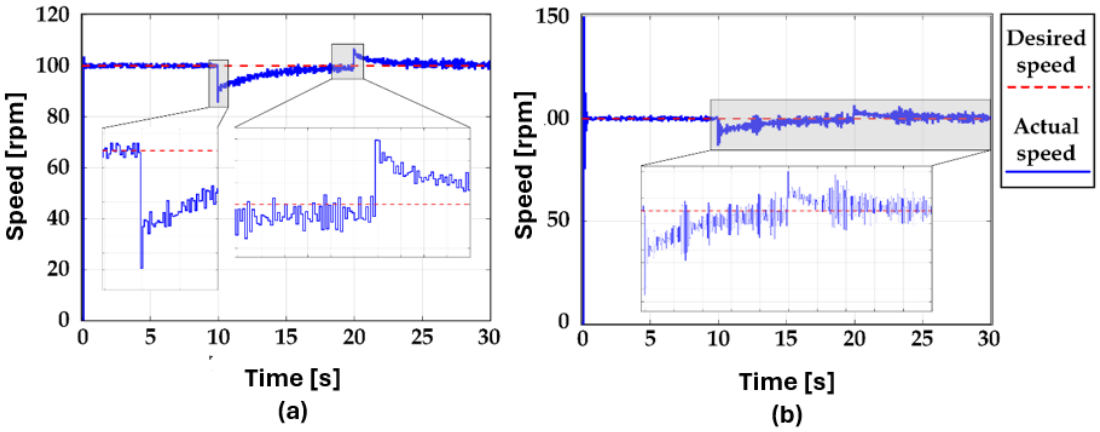

Fig. 4. Speed response of the system under the proposed controller in the case of a) λ = 0.99, c = 0.05; and b) λ = 0.99, c = 0.5.

Fig. 5a depicts the output response under the proposed algorithm with the inclusion of the damping function described in Eq. (13). In which, γ is chosen to be 0.99, while the controller parameters are set as $\lambda = 0.99$ and $c = 18$. The purpose of increasing c compared to previous cases is to strongly improve the response time and to evaluate the effectiveness of the damping function. Furthermore, a proportional-integral-derivative (PID) controller is simulated ($K_p = 0.05$, $K_i = 0.4$, $K_d = 0.001$) to compare the performance with the proposed algorithm, the results are depicted in Fig. 5b. For the proposed algorithm, the simulation results show a positive improvement in both the settling time, and steady-state error compared to the previous cases. Specifically, the transition times are 0.45, 0.55 and 0.5 seconds at the initial time and the times when the model is changed, respectively. It can be seen that the transient performance is maintained with negligible deviation, regardless of the model changes. Besides, the results show that the overshoot percentage is controlled by 2.67% at the initial time, 18.19% at the first transformation of the model ($t = 10$ seconds), and 11.10% at the remaining event ($t = 20$ seconds). At the steady state, the output velocity fluctuates around the desired value with an error $|e| < 2$ rpm, which is caused by the

disturbance. It is important that the velocity tends towards the desired velocity after the overshoot without any significant fluctuations. The result is attributed to strong SMC action (large c), fast RLS updates (small λ), and control signal smoothing via the damping function. For the results of the PID controller, the transient time is shown to be worse than the proposed algorithm, namely about 1.5 seconds at the initial time, 1.2 seconds at $t = 10$ seconds, and about 1.3 seconds for the remaining time of the model change. However, the overshoot percentage is controlled with improved results, namely 14.11% at $t = 10$ seconds, 8.11% at $t = 20$ seconds, and no overshoot occurs at the initial time. The reason is that the slow response results in better control over overshoot oscillations. The system error is controlled to fluctuate within 2 rpm, similar to the case of operation by the proposed controller. It is noted that the gains of the PID controller are selected based on the Ziegler–Nichols method [34], and their values are maintained during the simulation. This shows that within the small variation range of the system model, the PID parameter set still ensures the control of the system while the response behavior does not change significantly. However, controlling performance metrics such as settling time, overshoot, and steady-state error is infeasible due to closed-loop dynamics altering the desired pole locations used in gain design. Therefore, it can be concluded that both the proposed algorithm and the conventional PID are capable of controlling within the small variation range of the model parameters.

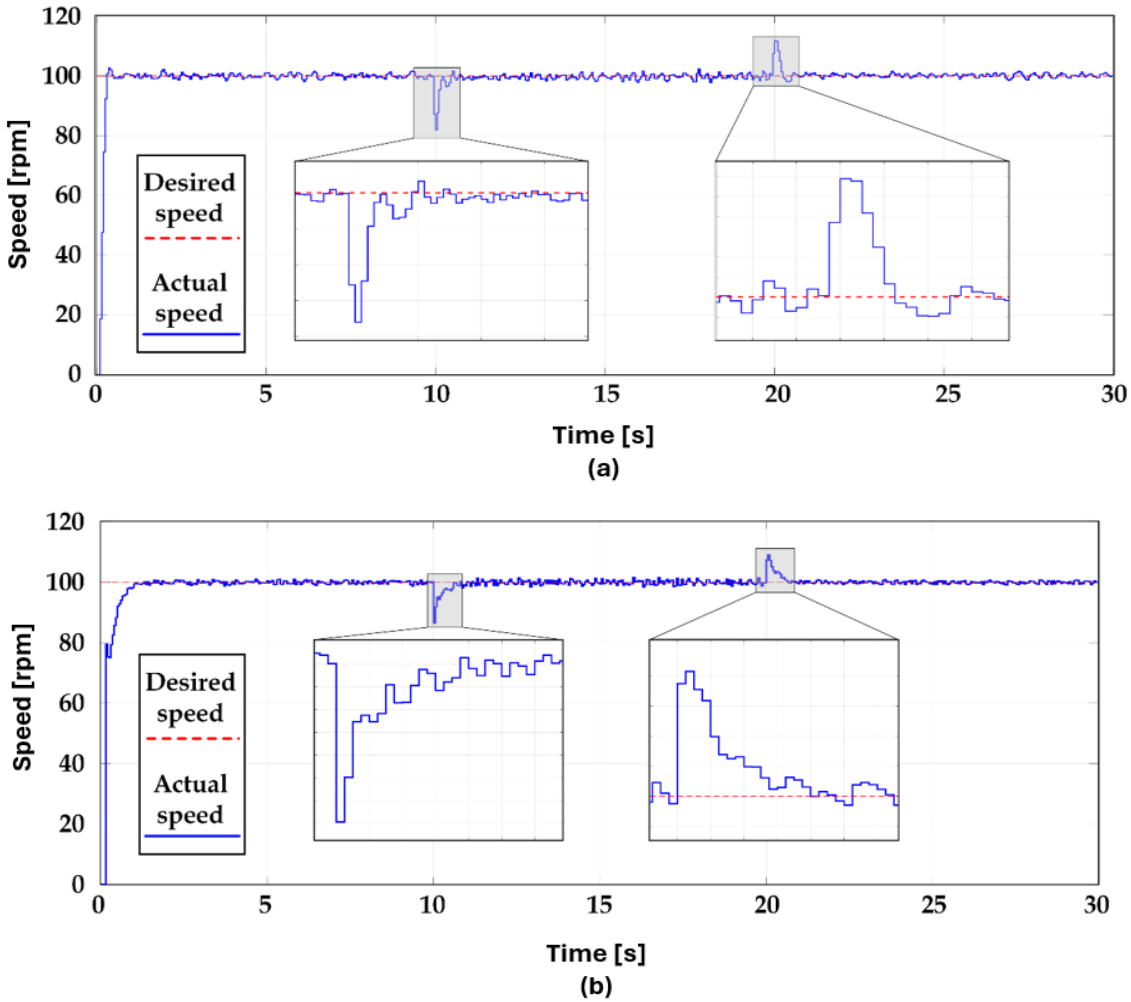

Fig. 5. Speed response of the system under a) the proposed controller (λ = 0.99, c = 5, β = 0.99); and b) PID controller (K_p = 0.05, K_I = 0.4, K_d = 0.001).

Fig. 6 illustrates the output response and tracking error of the Simscape DC motor model under varying reference signals and external disturbances. Specifically, the desired output is defined as a harmonic oscillation with a frequency of 0.05 Hz given by:

$$y_d(t) = 100\sin(2\pi 0.05t). \quad (14)$$

In addition, an external disturbance scenario is introduced to induce output deviations at the two moments of model variation, namely +15 rpm at $t = 10$ seconds and −20 rpm at $t = 20$ seconds. The control parameters ($\lambda, c, \alpha, \beta, \gamma$) and model parameter variation scenarios are set similar to the simulation scenario in Fig. 5. The output signal in Fig. 6a indicates that the system closely follows the reference oscillation, with a settling time below 0.3 seconds under external disturbances. Furthermore, the steady-state error observed in Fig. 6b remains approximately 6 rpm for most of the simulation period, increasing to 18.89 rpm at $t = 10$ seconds and 24.63 rpm at $t = 20$ seconds. These results indicate that the Simscape DC motor model is stably controlled under the proposed algorithm throughout the entire simulation, despite model variations and disturbances. From another perspective, the tracking error tends to decrease as the reference signal converges to its boundary, where the variation of the harmonic function becomes significantly attenuated. Combined with previous results, it is evident that the proposed control algorithm achieves its best performance when the desired output remains constant or exhibits minor fluctuations. Moreover, the settling-time characteristics in the scenarios of Fig. 5 and Fig. 6 are similar, both remaining below 0.5 seconds. This consistency indicates that the proposed algorithm maintains stable dynamic behavior under equivalent control parameter settings, regardless of disturbances or reference variations.

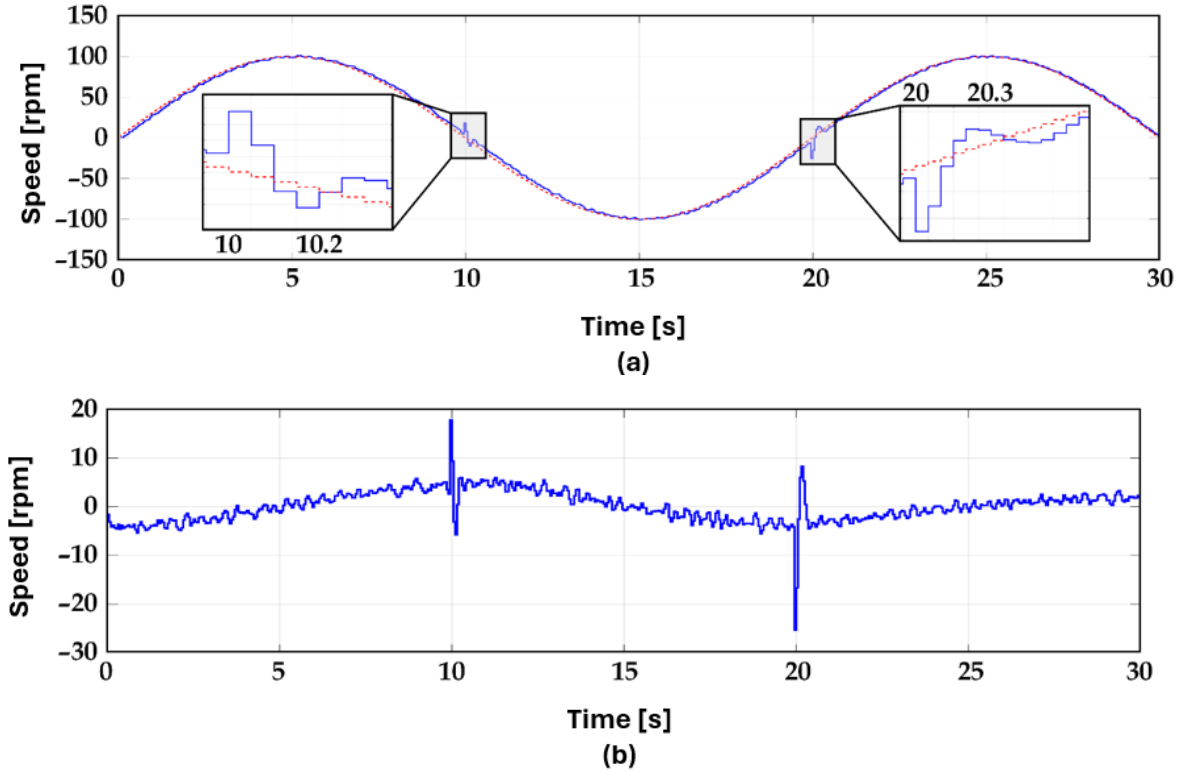

Fig. 6. a) Speed feedback; and (b) control error of the system under the proposed algorithm under the condition of varying reference signal and applied external force.

4. CONCLUSIONS

This paper proposed a control algorithm for the SISO system with model parameter changes based on the combination of SMC and RLS algorithms. Based on the simulation, the system output is confirmed to be unable to converge to the desired velocity without the intervention of the RLS algorithm. In addition, the simulation demonstrated the impact of the coefficients c, λ and their cross-influence on the system output. Specifically, the results showed that increasing c and decreasing λ lead to improved settling time but caused

instability in the control system. Although decreasing λ alone reduces the settling time, the improvement is marginal. Importantly, this study proposed a damping function for the control signal as a mandatory addition. The strong control of this function in eliminating sudden and significant oscillations in the control signal is clearly demonstrated. As a result, the overshoot phenomenon is completely eliminated at the initial time, the settling time is less than 2 seconds, and there is no significant oscillation at the time when the model fluctuates. Besides, the system error converges to zero with a small oscillation region ($|e| < 2$). Furthermore, an additional simulation was performed with the system control scenario using a PID controller with the aim of comparing the performance with the proposed algorithm. The results show that both algorithms are capable of controlling the output when the model remains relatively unchanged. However, the PID controller is inferior in controlling the desired behavior because the control parameters are fixed regardless of any changes in the model. Additionally, the simulation results confirm the algorithm's stability, achieving a settling time below 0.5 seconds and maintaining the steady-state error within about 6 rpm under varying references and disturbances.

On the other hand, the limitation of the study's results is only demonstrated by simulation; therefore, the similarity to real systems lacks the capacity to be fully confirmed. Additionally, the influence of input disturbances is considered to be negligible with small amplitudes $[-0.1, 0.1]$ voltage; mainly focusing on the times when the model simulation suddenly changed. Moreover, updating the model online at each sampling time causes a computational burden on the hardware.

For future work, an experimental platform based on a nonlinear SISO system is built to practically evaluate the proposed controller's performance. Besides, external forces, such as friction, vibration, etc., will be introduced to create situations that cause errors in the model. To reduce the computational load on hardware, studies related to determining the conditions that enable or disable the intervention of the RLS algorithm will be carried out.

Acknowledgement: This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number TX2025-20b-01. We acknowledge the support of time and facilities from Key Laboratory of Digital Control and System Engineering (DCSELab), Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] Y. Tu, R. Wang, W. Su, "Active disturbance rejection control—new trends in agricultural cybernetics in the future: a comprehensive review," *Machines,* vol. 13, no. 2, p. 111, 2025, doi: 10.3390/machines13020111.

[2] J. Tavoosi, "Stable backstepping sliding mode control based on ANFIS2 for a class of nonlinear systems," *Jordan Journal of Electrical Engineering*, vol. 6, no. 1, pp. 49-62, 2020, doi: 10.5455/jjee.204-1580573666.

[3] G. Yang, Z. Li, T. Li, J. Chang, "Fractional-order complementary terminal sliding mode control of active power filter based on improved reptile search algorithm," *Jordan Journal of Electrical Engineering*, vol. 11, no. 2, pp. 413-429, 2025, doi: 10.5455/jjee.204-1734527692.

[4] T. Yetayew, D. Tesfaye, "A comparative analysis of metaheuristic algorithms tuned super twisting sliding mode control of a self-balancing segway," *Jordan Journal of Electrical Engineering*, vol. 10, no. 4, p. 1, Aug. 2024, doi: 10.5455/jjee.204-1706551479.

[5] C. Truong, T. Phan, V. Duong, H. Nguyen, T. Nguyen, T. Nguyen, "Adaptive Bayesian optimization for proportional derivative control in double-acting piston pump ventilators," *Discover Applied Sciences*, vol. 7, no. 8, p. 848, 2025, doi: 10.1007/s42452-025-07176-x.

[6] N. Le, M. Nguyen, T. Phan, C. Truong, V. Duong, H. Nguyen, T. Nguyen, "Development of a multi-suspension unit for solar cleaning robots to mitigate vibration impact on photovoltaic panels," *Applied Sciences*, vol. 13, no. 22, p. 12104, 2023, doi: 10.3390/app132212104.

[7] T. Phan, M. Nguyen, M. Auffray, N. Le, C. Truong, V. Duong, H. Nguyen, T. Nguyen, "Research impact of solar panel cleaning robot on photovoltaic panel's deflection," in 4th International Conference on Applied Convergence Engineering, 2023, doi: 10.48550/arXiv.2306.05340.

[8] N. Le, M. Nguyen, C. Truong, V. Duong, H. Nguyen, "Recursive least squares algorithm for online parameter identification of DC motor: theory and practice," 10th International Conference on Engineering and Emerging Technologies, 2024, doi: : 10.1109/ICEET65156.2024.10913584.

[9] L. NT, T. CT, N. HH, N. TT, D. VT, "Yaw angle determination of a mobile robot operating on an inclined plane using accelerometer and gyroscope," *Measurement*, vol. 247, p. 116806 , 2025, doi: 10.1016/j.measurement.2025.116806.

[10] C. Truong, K. Huynh, V. Duong, H. Nguyen, L. Pham, T. Nguyen, "Characteristic of paddle squeezing angle and ambu bag air volume in bag valve mask ventilator," 17th International Conference on Intelligent Unmanned Systems, 2021, doi: 10.48550/arXiv.2109.08019.

[11] C. Truong, K. Huynh, V. Duong, H. Nguyen, L. Pham, N. Tien, "Model free volume and pressure cycled control of automatic bag valve mask ventilator," *AIMS Bioeng*, vol. 8, no. 3, pp. 192–207, 2021, doi: 10.3934/bioeng.2021017.

[12] C. Truong *et al.*, "Model indentification of two double-acting pistons pump A NARX network approach -submit," International Conference on Ubiquitous Robots, 2023, doi: 10.1109/UR57808.2023.10202388.

[13] T. Phan, L. Pham, C. Truong, V. Duong, H. Nguyen, "Adaptive sliding mode control for a blower-based breathing simulator with unknown and time-varying airway resistance and compliance," 10th International Conference on Engineering and Emerging Technologies, 2024, doi: 10.1109/ICEET65156.2024.10913788.

[14] C. Truong, K. Huynh, V. Duong, H. Nguyen, L. Pham, T. Nguyen, "Linear regression model and least square method for experimental identification of AMBU bag in simple ventilator," *International Journal of Intelligent Unmanned Systems*, vol. 11, no. 3, pp. 378–395, 2023, doi: 10.1108/IJIUS-07-2021-0072.

[15] C. Truong, T. Phan, V. Duong, H. Nguyen, T. Nguyen, "Model identification of ventilation air pump utilizing ridge-momentum regression and grid-based structure optimization," *Mathematical Biosciences & Engineering*, vol. 22, no. 8, pp. 2020–2038, 2025, doi: 10.3934/mbe.2025074.

[16] C. Truong, T. Phan, M. Tran, H. Nguyen, V. Duong, T. Nguyen, "Disturbance observer-based sliding mode control for ventilation blower-based systems: controller design and simulation," *International Journal of Online and Biomedical Engineering*, vol. 21, no. 11, pp. 132–151, 2025, doi: 10.3991/ijoe.v21i11.56677.

[17] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 212–222, 1977, doi: 10.1109/TAC.1977.1101446.

[18] K. Abidi and J. Xu, *Advanced Discrete-Time Control*, Springer , 2015.

[19] T. Yoshimura, "Discrete-time adaptive sliding mode control for a class of uncertain time delay systems," *JVC/Journal of Vibration and Control*, vol. 17, no. 7, pp. 1009–1020, 2011, doi: 10.1177/1077546310367890.

[20] B. Wu, S. Li, X. Wang, "Discrete-time adaptive sliding mode control of autonomous underwater vehicle in the dive plane," Intelligent Robotics and Applications: Second International Conference, 2009, doi: 10.1007/978-3-642-10817-4_15.

[21] W. Gao, J. Hung, "Variable structure control of nonlinear systems: a new approach," *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 45–55, 1993, doi: 10.1109/41.184820.

[22] M. Amini, M. Shahbakhti, S. Pan, "Adaptive discrete second-order sliding mode control with application to nonlinear automotive systems," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 140, no. 12, 2018, doi: 10.1115/1.4040208.

[23] A. Simpkins, "System identification: theory for the user, 2nd edition (Ljung, L.; 1999)," *IEEE Robot Autom Mag*, vol. 19, no. 2, pp. 95–96, 2012, doi: 10.1109/MRA.2012.2192817.

[24] K. Oh, J. Seo, "Recursive least squares based sliding mode approach for position control of DC motors with self-tuning rule," *Journal of Mechanical Science and Technology*, vol. 34, 2020, doi: 10.1007/s12206-020-1124-1.

[25] X. Wang, Y. Wu, F. Yan, Y. Gao, Z. Xu, "Adaptive terminal sliding-mode control for servo systems with inertia variations," MATEC Web of Conferences, 2018. doi: 10.1051/matecconf/201816005004.

[26] A. Sguarezi Filho, D. Fernandes, J. Suárez, F. Costa, J. Altuna, "Recursive least squares and sliding mode control for voltage compensation of three-phase loads," *Journal of Control, Automation and Electrical Systems*, vol. 29, no. 6, pp. 769–777, 2018, doi: 10.1007/s40313-018-0405-8.

[27] K. Oh, J. Seo, "Recursive least squares based sliding mode approach for position control of DC motors with self-tuning rule," *Journal of Mechanical Science and Technology*, vol. 34, no. 12, pp. 5223–5237, 2020, doi: 10.1007/s12206-020-1124-1.

[28] J. Valladolid, J. Ortiz, L. Minchala, "*Adaptive Quasi-Sliding Mode Control Based on a Recursive Weighted Least Square Estimator for a DC Motor*," IEEE Conference on Control Applications, 2016, doi: 10.1109/CCA.2016.7587925.

[29] A. Znidi, K. Dehri and A. S. Nouri, "*Adaptive sliding mode control for discrete uncertain systems using matrix RLS algorithm*," International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, Hammamet, Tunisia, 2014, pp. 953-957, doi: 10.1109/STA.2014.7086761.

[30] K. Petersen, M, Pedersen, *The Matrix Cookbook*. Technical University of Denmark, 2012, http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html.

[31] W. Gao, Y. Wang, A. Homaifa, "Discrete-time variable structure control systems," vol. 42, no. 2, pp. 117-122 , 1995, doi: 10.1109/41.370376.

[32] J. Samantaray, S. Chakrabarty, *Discrete Time Sliding Mode Control*, IntechOpen, 2020.

[33] D. Wackerly, W. Mendenhall, R. Scheaffer, *Mathematical Statistics with Applications*, Thomson Learning Academic Resource Center, 2008.

[34] K. Åström, T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, no. 6, pp. 635–650, 2004, doi: 10.1016/j.jprocont.2004.01.002.