Online Learning ARMA Controllers With Guaranteed Closed-Loop Stability

Savaş Şahin and Cüneyt Güzeliş

***Abstract*— This paper presents a novel online block adaptive learning algorithm for autoregressive moving average (ARMA) controller design based on the real data measured from the plant. The method employs ARMA input–output models both for the plant and the resulting closed-loop system. In a sliding window, the plant model parameters are identified first offline using a supervised learning algorithm minimizing an ε-insensitive and regularized identification error, which is the window average of the distances between the measured plant output and the model output for the input provided by the controller. The optimal controller parameters are then determined again offline for another sliding window as the solution to a constrained optimization problem, where the cost is the ε-insensitive and regularized output tracking error and the constraints that are linear inequalities of the controller parameters are imposed for ensuring the closed-loop system to be Schur stable. Not only the identification phase but also the controller design phase uses the input–output samples measured from the plant during online learning. In the developed online controller design method, the controller parameters can always be kept in a parameter region providing Schur stability for the closed-loop system. The ε-insensitiveness provides robustness against disturbances, so does the regularization better generalization performance in the identification and the control. The method is tested on benchmark plants, including the inverted pendulum and dc motor models. The method is also tested on an emulated and also a real dc motor by online block adaptive learning ARMA controllers, in particular, Proportional-Integral-Derivative controllers.**

***Index Terms*— ε-insensitive, adaptive control, online learning, Schur stability, time-varying systems.**

I. Introduction

ADAPTIVE control is still an attractive area due to its capability of producing efficient solutions in control of time-varying plants possibly under the changing conditions [1]–[7].

In this paper, a new online block adaptive learning algorithm for dynamical adaptive controller design, which ensures Schur stability of the closed-loop system, is developed. The developed algorithm employs autoregressive moving average (ARMA) input–output models for both the plant and the closed-loop system consisting of the plant and controller. The proposed learning algorithm can be viewed as a scheme solving Diophantine equation in real-time using data measured from the real plant not from a model of the plant [1]. The developed online learning algorithm used for adaptive controller design has the following features.

1) It is based on linear ARMA models for the plant, controller, and closed-loop system.
2) The control law comprises a model identification phase and a controller design phase succeeding the model identification. In this sense, it is a kind of self-tuning adaptive control scheme [1].
3) The plant model parameters are identified in an offline manner for each sliding window by using (the plant's input–output) data measured at the outputs of the controller and plant such that the number of data used at each step is equal to the chosen length K of a sliding (identification) window.
4) The controller parameters are determined in an offline manner for each (controller design) sliding window by using (the closed-loop system's input–output) data where the actual input and also the desired output are the reference signal to be tracked and the number of data used for each parameter update is equal to the length L of the sliding window.
5) The sampling time used in the measurement of the data from the plant and from the reference is chosen fixed and different from the processing times needed for updating of the plant and controller parameters. The parameter updates are realized whenever the optimization processes related to the identification and resp. controller design are completed. Therefore, the data measurement and parameter update are not synchronous in general and the parameter update occurs in a random way with a variable step size. The sampling time is actually less than the processing time needed for determining the new parameter values since the constraints imposed for ensuring Schur stability approximately eightfold slow down the whole identification and the controller design process.
6) The proposed method defines a kind of model reference adaptive control (MRAC) when the desired output of the plant is provided by a stable reference model.
7) The plant model parameters and also the controller parameters are found by minimizing the identification and

Manuscript received December 11, 2014; revised September 10, 2015; accepted September 17, 2015. Date of publication October 8, 2015; date of current version October 17, 2016. This work was supported in part by the Scientific and Technological Research Council of Turkey under Grant 114E432 and in part by the Scientific Research Projects Office of İzmir Katip Çelebi University under Grant BAP-2013-1-FMBP-09.

S. Şahin is with the Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, İzmir Katip Çelebi University, İzmir 35620, Turkey (e-mail: sahin.savas@yahoo.com).

C. Güzeliş is with the Department of Electrical and Electronics Engineering, Faculty of Engineering, Üniversite Caddesi, Yaşar University, İzmir 35100, Turkey (e-mail: cuneyt.guzelis@yasar.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2480764

output tracking errors which are chosen as a regularized ε-insensitive loss functions.

8) Euclidean (ℓ_2) and absolute (ℓ_1) loss functions are used alternatively both for the identification and tracking error measures. The ε-insensitiveness [8] is observed in software-in-the-loop and prototyping simulation modes [9] to provide robustness against to noise of small variance. The ℓ_1 loss function and the regularization are seen to yield robustness against to disturbances and less generalization error, respectively.
9) The method imposes an additional constraint as the linear inequality system of the controller parameters in the minimization of the tracking error to ensure the Schur stability for the closed-loop system. Therefore, it provides an adaptive control law ensuring the Schur stability of the closed-loop system.
10) Design of the Schur stable (adaptive or fixed) Proportional-Integral-Derivative (PID) controllers is included as a special case of the proposed method.

The developed adaptive dynamical controllers, in particular PID controllers, are designed and tested in the controller-design-test-redesign-platform (CDTRP) [9] and are applied for controlling a real dc motor. The method and obtained application results contribute to the literature on data driven adaptive control, adaptive PID controller design [10]–[16], and control of real dc motor [17]–[20].

This paper is organized as follows. In Section II, the developed adaptive control scheme is briefly described. The proposed control method is presented in this section for the most general case based on considering ARMA models both for the plant and controller. The Schur stability conditions for the closed-loop system are derived in Section III in terms of controller parameters by using the identified plant parameters. Section IV presents the real-time simulation results obtained in the software-in-the-loop and prototyping modes of CDTRP [9] for a first-order time-varying dc motor model, a fourth-order dc motor model, both the linear and nonlinear inverted pendulum models and also a real dc motor. The performances of the proposed adaptive ARMA controllers and the conventional PID controllers are also compared in this section. The conclusions and possible future directions for further development of the proposed adaptive controller are given in Section V.

II. Supervised Learning Identification and Adaptive Control Based on ARMA Models

A closed-loop control system design problem can be defined as the identification of a partially known system when a plant model is available and when the desired system output is provided such as in tracking problems. The plant constitutes the known part of the control system. This known part is usually given by an identified model with known parameters. Therefore, designing control systems of two degrees of freedom can be defined as a supervised learning problem for a partially known input–output system model. A general two degrees of freedom control is shown in Fig. 1 [1] where reference $r(k)$ constitutes the closed-loop system input and the plant output $y(k)$ constitutes the actual output of the closed-loop system. The desired output $y_d(k)$ of the closed-loop system might be a constant or a time-varying reference signal $r(k)$ or the output of a reference model $y_m(k)$. In explicit terms, the design of the controller in Fig. 1 can be posed as the minimization of the error between the desired output $y_d(k)$ and the actual output $y(k)$ corresponding to reference input signal $r(k)$ in terms of the parameters of the controller.

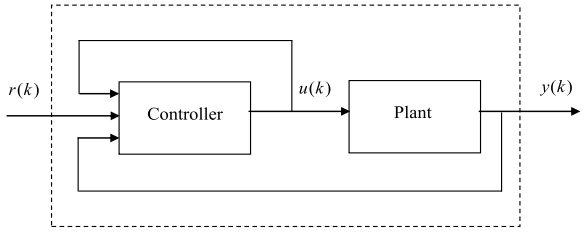

Fig. 1. General two degrees of freedom control scheme.

A. Derivation of ARMA Representation for Closed-Loop System

Assume the following ARMA model for a single-input single-output (SISO) plant in Fig. 1:

$$y(k) = \sum_{n=1}^{N} a_n y(k-n) + \sum_{n=0}^{M} b_n u(k-n). \tag{1}$$

Equation (1) can also be written in an implicit form in (2) with $a_0 = -1$

$$\sum_{n=0}^{N} a_n y(k-n) + \sum_{n=0}^{M} b_n u(k-n) = 0. \tag{2}$$

Next, assume that the controller with two degrees of freedom in Fig. 1 has an ARMA representation in

$$u(k) = \sum_{m=1}^{P} f_m u(k-m) + \sum_{m=0}^{R} c_m r(k-m) + \sum_{m=0}^{Q} d_m y(k-m). \tag{3}$$

Equation (3) can also be written as below in an implicit form with $f_0 = -1$

$$\sum_{m=0}^{P} f_m u(k-m) + \sum_{m=0}^{R} c_m r(k-m) + \sum_{m=0}^{Q} d_m y(k-m) = 0. \tag{4}$$

To obtain an input–output representation for the closed-loop system, the control input $u(k)$ should be eliminated by writing it in terms of the reference input $r(k)$ and the output $y(k)$ using the expression in (4). This can be achieved by obtaining the first term of (4) in any way, for instance by taking the weighted sum of the both sides of the plant ARMA representation in (2) with f_m weights, so obtaining

$$\sum_{m=0}^{P} f_m \sum_{n=0}^{N} a_n y(k-m-n) + \sum_{m=0}^{P} f_m \sum_{n=0}^{M} b_n u(k-m-n) = 0. \tag{5}$$

The control input can be eliminated using (4) and interchanging the sums over m and n in the second term of (5), so the

closed ARMA representation is obtained as in

$$\sum_{m=0}^{P} f_m \sum_{n=0}^{N} a_n y(k-m-n) + \sum_{n=0}^{M} b_n \times \left\{ -\sum_{m=0}^{R} c_m r(k-m-n) - \sum_{m=0}^{Q} d_m y(k-m-n) \right\} = 0. \quad (6)$$

The closed loop-ARMA representation in (6) can be simplified as in (7) with newly defined parameters α_n and β_n which can be derived by equating (6) to (7) after moving $y(k)$ to the right-hand side of

$$y(k) = \sum_{n=0}^{\hat{N}} \alpha_n y(k-n) + \sum_{n=0}^{\hat{M}} \beta_n r(k-n). \quad (7)$$

Herein, $\hat{N} =: \max\{P+N, M+Q\}$ and $\hat{M} =: M+R$. The relations among the closed-loop parameters α_n and β_n, the plant ARMA parameters a_n and b_n, and the controller parameters c_m, d_m, and f_m can be obtained for different choices of P, N, M, R, and Q. These relations can be given as in (8)–(12), assuming $N = M = P = R = Q$ so $\hat{N} = \hat{M} = 2N$ without loss of generality

$$\alpha_0 =: 1 + a_0 f_0 - b_0 d_0 \quad (8)$$

$$\alpha_i =: \sum_{j=0}^{i} a_j f_{i-j} - \sum_{j=0}^{i} b_j d_{i-j} \quad \text{for } i \in \{1, 2, \ldots, N\} \quad (9)$$

$$\alpha_i =: \sum_{j=i-N}^{N} a_j f_{i-j} - \sum_{j=i-N}^{N} b_j d_{i-j} \quad \text{for } i \in \{N+1, N+2, \ldots, 2N\} \quad (10)$$

$$\beta_i =: -\sum_{j=0}^{i} b_j c_{i-j} \quad \text{for } i \in \{0, 1, \ldots, N\} \quad (11)$$

$$\beta_i =: -\sum_{j=i-N}^{N} b_j c_{i-j} \quad \text{for } i \in \{N+1, N+2, \ldots, 2N\}. \quad (12)$$

Equations (8)–(10) indeed define the well-known Diophantine equation [1]. Different control schemes, the so-called model following control, pole placement, self-tuning regulator adaptive control, and MRAC [1] can be derived based on the closed-loop ARMA representation given by (7)–(12). The adaptive control method proposed in this paper is also derived from the above closed-loop ARMA representation as described in the sequel.

Consider input–output data set $\{(u[k-s, N], y_a[k-s, N]\}_{s=0}^{K-1}$ measured from the actual plant in closed loop and the set of reference inputs-desired outputs $\{(r[k-s, N], y_d[k-s, N]\}_{s=0}^{L-1}$ where $x[t, N] =: [x(t), x(t-1), \ldots, x(t-N)]$ with $t = k-s$ represents the vector of current and past N sample values of a signal $x(t)$ which stands for $u(t)$, $r(t)$, $y_a(t)$, or $y_d(t)$.

B. Identification Phase

The $\{(u[k-s, N], y_a[k-s, N]\}_{s=0}^{K-1}$ set is obtained in closed loop by measuring the outputs of the controller and plant during time intervals $[k, k-K+1]$ specified with the sliding window of length K. Herein, for a digital implementation of the controller, the output of the controller is actually computed in closed loop by using (3) for: 1) the past values of the control input; 2) the past and current values of the plant output and the reference signal; and 3) the current values of the controller parameters, which are learned in the controller design window of length L coming just before the current identification window of length K. Identification of the plant parameters in the time interval $[k, k-K+1]$ is realized in an offline manner by minimizing the following identification error in (13) in terms of the plant ARMA parameters a_n, b_n, and λ is the scaling factor:

$$\frac{1}{K} \sum_{s=0}^{K-1} \ell_\varepsilon \left(y_a(k-s), \sum_{n=1}^{N} a_n y(k-s-n) + \sum_{n=0}^{N} b_n u(k-s-n) \right) + \lambda \left\| \begin{matrix} a \\ b \end{matrix} \right\|_2^2 \quad (13)$$

$\ell_\varepsilon(\circ, \circ)$ in (13) is the ε-insensitive loss function measuring the distance between the $(k-s)$th actual output sample $y_a(s)$ of the plant and the $(k-s)$th output $y(k-s) = \sum_{n=1}^{N} a_n y(k-s-n) + \sum_{n=0}^{N} b_n u(k-s-n)$ of the plant model. The loss function might be ℓ_2 Euclidean squared error or ℓ_1 absolute error, both with or without ε-insensitiveness. Euclidean squared ε-insensitive loss function is defined as $\ell_\varepsilon(y_a(s), y(s)) = (|y_a(s) - y(s)| - \varepsilon)^2$ if $|y_a(s) - y(s)| \geq \varepsilon$ and $\ell_\varepsilon(y_a(s), y(s)) = 0$ if $|y_a(s) - y(s)| < \varepsilon$. Absolute ε-insensitive loss function is defined as $\ell_\varepsilon(y_a(s), y(s)) = |y_a(s) - y(s)| - \varepsilon$ if $|y_a(s) - y(s)| \geq \varepsilon$ and $\ell_\varepsilon(y_a(s), y(s)) = 0$ if $|y_a(s) - y(s)| < \varepsilon$. Herein, ε-insensitiveness is introduced for having robustness against measurement noise and small changes in the plant's output. $\left\| \begin{matrix} a \\ b \end{matrix} \right\|_2^2 = \sum_{n=1}^{N} a_n^2 + \sum_{n=0}^{N} b_n^2$, i.e., the square of the Euclidean norm of the model parameters, is the regularization term which provides a smooth model avoiding overfitting.

C. Controller Design Phase

The $\{(r[k-s, N], y_d[k-s, N]\}_{s=0}^{L-1}$ set defines input–output behavior of the closed-loop system during time intervals $[k, k-L+1]$ specified with the sliding window of length L. For each sliding window, the controller parameters can be determined in an offline manner by minimizing the following closed loop output (tracking) error in (14) in terms of the controller parameters c_m, d_m, and f_m which are related to the closed-loop parameters α_n and β_n via (8)–(12):

$$\frac{1}{L} \sum_{s=0}^{L-1} \ell_\varepsilon \left(y_d(k-s), \sum_{n=0}^{2N} \alpha_n y(k-s-n) + \sum_{n=0}^{2N} \beta_n r(k-s-n) \right) + \lambda \left\| \begin{matrix} \alpha \\ \beta \end{matrix} \right\|_2^2. \quad (14)$$

In (14), $\ell_\varepsilon(\circ, \circ)$ is the ε-insensitive loss function, $y_d(k-s)$ is the $(k-s)$th sample of desired output and $\sum_{n=1}^{N} \alpha_n y(k-s-n) + \sum_{n=0}^{N} \beta_n u(k-s-n)$ is the $(k-s)$th sample of the actual output of the plant model.

The ε-insensitiveness and the regularization term are for having robustness and for reducing overfitting, respectively.

The second phase, where the control parameters are determined, defines a new kind of controller design method, which is, indeed, a supervised learning scheme applied on a partially known system since a part of the system parameters, i.e., the plant parameters, are already known in this phase. The plant parameters are determined first by minimizing (13) for the whole set of plant input–output measurements and then the controller parameters c_m, d_m, and f_m are found as minimizing (14) for the whole set of closed-loop system input–output samples. The plant parameters are updated according to K samples $\{(u[k-s,N], y_a[k-s,N]\}_{s=0}^{K-1}$ and then the controller parameters c_m, d_m, and f_m are updated according to L samples $\{(r[k-s,N], y_d[k-s,N]\}_{s=0}^{L-1}$. Where K and L are chosen sufficiently large for estimating the plant and controller parameters within reasonable accuracies and on the other hand chosen sufficiently small for responding changes in the plant reasonably fast.

Note that the plant and controller parameters are updated whenever their offline computations are ended. Therefore, their updates may occur at different time instants and within a sampling period, or within or not within the time durations specified by the associated window lengths.

III. Stability

This section provides a set of sufficient conditions for the Schur stability of the closed-loop system in (6). So, (7) is considered with the parameters in (8)–(12) and by assuming $N = M = P = R = Q$ so $\hat{N} = \hat{M} = 2N$ without loss of generality.

A linear time-invariant (LTI) discrete time system is bounded-input bounded-output (BIBO) stable if and only if all of the roots to the characteristic equation are located inside the unit disk [21]. This stability definition of discrete time systems is known as Schur stability. Let us assume there is no pole-zero cancellation and consider the denominator of a z-domain transfer function of an LTI discrete time system

$$p(z) = \theta_0 + \theta_1 z + \cdots + \theta_n z^n \tag{15}$$

where $\theta_i (i = 0, 1, \ldots, n)$ represent the coefficients of the system. It is known that the linear inequalities $\theta_n > \cdots > \theta_1 > \theta_0 > 0$ constitute a set of sufficient conditions for the Schur stability.

Once the plant's ARMA parameters a_n, b_n's are obtained, the sufficient conditions for BIBO stability now imposed on α_n parameters as in (16) may be satisfied by selecting the appropriate controller parameters d_m and f_m

$$\alpha_0 > \cdots > \alpha_{2N-1} > \alpha_{2N} > 0. \tag{16}$$

Note that (5) can be written in the following generalized form:

$$\left\{ \sum_{m=0}^{N} f_m \sum_{n=0}^{N} a_n - \sum_{n=0}^{N} b_n \sum_{m=0}^{N} d_m \right\} y(k-m-n) = \sum_{n=0}^{N} b_n \sum_{m=0}^{N} c_m r(k-m-n). \tag{17}$$

It is seen that the controller parameters d_m and f_m's define the autoregressive part, so $\theta_i (i = 0, 1, \ldots, n)$ parameters of (15) while the other controller parameters c_m are irrelevant to the Schur stability, so can be chosen just for shaping the reference input for some compensation purposes.

Now, the condition $\alpha_0 > \cdots > \alpha_{2N-1} > \alpha_{2N} > 0$ can be written in terms of the controller parameters as follows:

$$\begin{aligned}
&b_0 d_1 > -1 + (b_0 - b_1) d_0 + (a_1 - a_0) f_0 + a_0 f_1 \\
&b_0 d_2 > (b_1 - b_2) d_0 + (b_0 - b_1) d_1 + (a_2 - a_1) f_0 \\
&\quad + (a_1 - a_0) f_1 + a_0 f_2 \\
&\quad \vdots \\
&(b_N - b_{N-1}) d_N b_N d_{N-1} - a_N f_{N-1} + (a_N - a_{N-1}) f_N \\
&-b_N d_N > -a_N f_N.
\end{aligned} \tag{18}$$

It is concluded that, in order to ensure the Schur stability of the closed-loop system, the linear inequality conditions of (18) may be considered as constraints in the optimization problem given in (14) whose solution provide the Schur stable models for the closed-loop control system. Another way of ensuring the Schur stability of the closed-loop system is to implement the optimization in (14) under the constraints in (17), and then solve the controller parameters from (8)–(12).

IV. Simulation Results and Real DC Motor Applications

The method is tested via five different experiments all realized in different modes of CDTRP [9]. In the first four experiments, the proposed adaptive controller and the chosen plants are both implemented as software simulations in the software-in-the-loop (precisely, S-S-S and S-E-S) mode of CDTRP [9]. The first experiment employs a first-order dc motor model with an additive time-varying parameter to test the adaptation ability of the proposed adaptive controller against to the changes in the plant. The second experiment employs a time-invariant fourth-order dc motor model to see the performance of a six-parameter (adaptive) ARMA controller for a relatively higher order, i.e., a fourth order, plant model. The third experiment employs a time-varying linear and also a time invariant nonlinear inverted pendulum model to test tracking performances of the developed control method on multiplicative time-varying plants and nonlinear plants. The fourth experiment is designed to test the effect of ε-insensitiveness on the performance of the proposed algorithm on a first-order dc motor model. The fifth experiment is designed to test the performance of a six-parameter adaptive ARMA controller under the additive disturbance when ε-insensitive intervals are available and also not available in the controller design for a tracking problem.

The fifth experiment is implemented in the prototyping (precisely, S-R-R) mode of CDTRP so that the controller is implemented as software while the plant is the real dc motor [9]. The robustness performances of a proposed six-parameter adaptive ARMA controller defining a PID controller as a special case and also a PID controller designed by the Ziegler–Nichols's first method are tested and their tracking performances are compared with each other for a

Algorithm 1 Simplified Pseudo Codes for c_m, d_m Update

- Define global variables: iteration index k, desired output y_d, actual output y_a, error e, control signal u, loss function ℓ_ε, identification window length K, controller design window length L;

begin; for each k;

- Determine values of variables;

if (k>0) e(k); $y_a(k)$, …$y_a(k-L)$; $y_d(k)$, …$y_d(k-L)$; else e=0; y=0; y_d=0; end;

- Minimize the following identification loss to compute the plant ARMA parameters a_n and b_n

$$\frac{1}{K}\sum_{s=0}^{K-1}\ell_\varepsilon(y_a(k-s),\sum_{n=1}^{N}a_n y(k-s-n)+\sum_{n=0}^{N}b_n u(k-s-n))+\lambda\left\|\begin{matrix}a\\b\end{matrix}\right\|_2^2$$

- Minimize the following tracking loss function to compute the controller parameters c_m, d_m under the Schur stability constraints in Equation 18.

$$\frac{1}{L}\sum_{s=0}^{L-1}\ell_\varepsilon(y_d(k-s),\sum_{n=0}^{2N}\alpha_n y(k-s-n)+\sum_{n=0}^{2N}\beta_n r(k-s-n))+\lambda\left\|\begin{matrix}\alpha\\\beta\end{matrix}\right\|_2^2$$

- Determine the control signal $u(k)$ by Equation 3 for the obtained controller parameters, the desired $y_d(k)$ and actual output $y_a(k)$, then go to begin.

real dc motor. Herein, the Euclidean squared error (ℓ_2) with ε-insensitiveness is used in both of the identification and tracking error minimization phases. The PID design by the proposed algorithm is implemented in the MATLAB numerical software package. The pseudocodes for the algorithm are given in Algorithm 1.

Experiment I (Time-Varying DC Motor Model): The developed adaptive control method is applied to control the following linear time-varying (indeed, nonautonomous) dc motor model [22]:

$$J\frac{dy(t)}{dt}=T_l(t)+K_c u(t) \tag{19}$$

where J is the total moment inertia of the motor axis, $u(t)$ is the motor current, K_c is the motor current constant, $y(t)$ is the angular speed, and $T_l(t)$ is the load disturbance torque. The friction compensation, compliance, and torque ripple are not involved in the model of (19). Such dc motors are widely used in the mechatronics systems [22] in which the dc motor parameters are $J=0.1$ kgm^2, $K_c=1$, and $T_l(t)=0.2\text{Sin}(2\pi ft)$ with $f=0.5$ Hz [9].

The following first-order LTI ARMA model is chosen for the identification of the time-varying dc motor model in (19):

$$y(k)=a_0 y(k)+a_1 y(k-1)+b_0 u(k). \tag{20}$$

The proposed adaptive controller and the chosen plant are both implemented as software simulations in the S-S-S mode of CDTRP [9] with choosing the sampling frequency as $f=10$ Hz. In the identification phase, the ARMA identification model parameters a_0, a_1, and b_0 are learned in each sliding window with length K by minimizing the regularized identification error in (13) for K input–output data $\{(u[k-s,N], y_a[k-s,N]\}_{s=0}^{K-1}$ measured from the plant of (19). Herein, the ARMA plant model parameters found in this way are then held constant along the controller design phase.

In the controller design phase, the following second-order six-parameter ARMA controller defined with the parameters c_0, c_1, c_2 and d_0, d_1, d_2 is chosen:

$$u(k)=c_0 r(k)+c_1 r(k-1)+c_2 r(k-2)+d_0 y(k)+d_1 y(k-1))+d_2 y(k-2). \tag{21}$$

The resulting closed-loop system model becomes also a second-order ARMA model in

$$y(k)=\alpha_0 y(k)+\alpha_1 y(k-1)+\alpha_2 y(k-2)+\beta_0 r(k)+\beta_1 r(k-1)+\beta_2 r(k-2) \tag{22}$$

with $\alpha_0=a_0-d_0b_0$, $\alpha_1=1-a_0+a_1-d_1b_0$, $\alpha_2=-a_1-d_2b_0$, $\beta_0=-c_0b_0$, $\beta_1=-c_1b_0$, and $\beta_2=-c_2b_0$. The closed-loop (Schur stable) ARMA system parameters α_0, α_1, α_2, β_0, β_1, and β_2 are learned via the minimization of the regularized tracking error in (14) under the Schur stability constraints of (18) for the L input-desired output data $\{(r[k-s,N], y_d[k-s,N]\}_{s=0}^{L-1}$. The optimal controller parameters c_0, c_1, c_2, d_0, d_1, and d_2 are then calculated in a unique way by solving the linear algebraic equations in (8)–(12).

In the experiments, the desired trajectory for the dc motor angular frequency is chosen as the square wave $r(k)=4\text{Square}(2\pi fk)$ with $f=0.2$ Hz and the sliding window lengths K and L are chosen identical to each other and equal to 3 and 10, respectively. The Euclidean squared error (ℓ_2) with no ε-insensitiveness is used in both of the identification and tracking error minimization phases.

In order to test the identification performance of the method for different choices of identification window length K, the controller window length L is chosen fixed as 10 and, for the chosen window length K, the input–output data measured from the time-varying plant in (19) fed by the output of the controller are used for learning the parameters of the plant ARMA model in (20).

It is seen from Fig. 2, the three parameter ARMA model in (20) whose parameters are updated for each window with length K identifies the time-varying plant defined in (19) well if the sliding window length is chosen as greater than or equal to 3 which is actually the number of samples and so the number of equations required to solve the three parameters of the ARMA model in (20). The effect of the identification window length K on the identification error performance for a time interval of 20 s covering $H=200$ samples is also measured in terms of normalized mean squared error (NMSE)

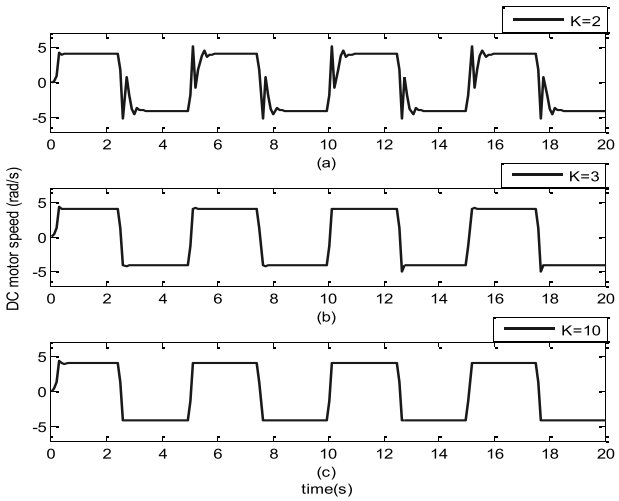

Fig. 2. (a)–(c) Identification results for the plant output $y(k)$, which is the angular speed of the dc motor model in (20) for the sliding window length K chosen as 2, 3, and 10. The controller design window length L is chosen as 10 in all of the three cases. The sampling period for the output $y(k)$ is 0.1 s.

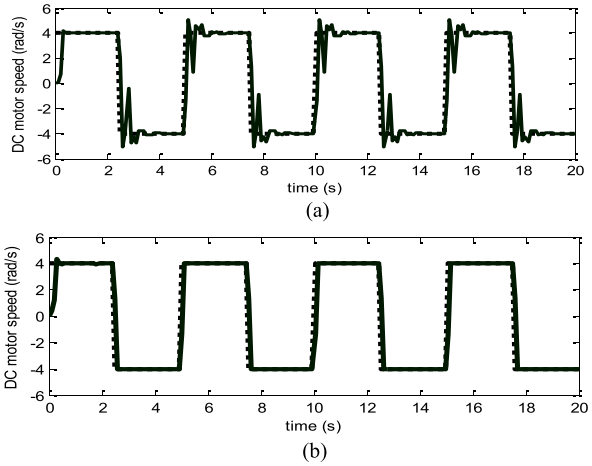

Fig. 3. (a) and (b) Tracking results for the angular speed $y(k)$ of the time-varying dc motor model in (19) for the tracking window length L chosen as 3 and 10. The window length K is chosen as 3 in both of the cases. Dotted line: reference signal. Solid line: actual output. The sampling period for the signals is 0.1 s.

where NMSE is defined as

$$\frac{\sum_{k=0}^{H-1}(y_a(k)-y(k))^2}{\sum_{k=0}^{H-1}\left(y_a(k)-\frac{1}{H}\sum_{s=0}^{H-1}y_a(k)\right)^2}.$$

NMSE for the identification under the disturbance is obtained as NMSE $= 0.4312$ for $K = 2$, NMSE $= 0.1562$ for $K = 3$, and NMSE $= 0.0935$ for $K = 10$. The identification NMSE is measured lower when no disturbance is applied, e.g., for $K = 3$, NMSE becomes 0.0997. So, for the considered plant in (19), window length $K = 3$ can be said the most appropriate one since it has the least computational load among the ones providing acceptable identification error.

After determining an appropriate identification window length K, namely, as 3, the tracking performance of the proposed method is tested for different choices of tracking window length L under no disturbance. The tracking performance results of the adaptive controller for the sliding window lengths $L = 3$ and $L = 10$ are shown in Fig. 3(a) and (b),

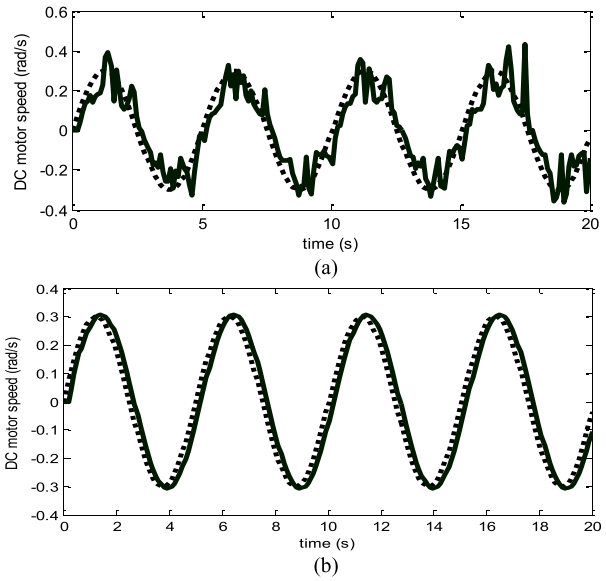

Fig. 4. (a) and (b) Tracking results for the angular speed $y(k)$ of the fourth-order dc motor model in (23) for the tracking window length L chosen as 3 and 10. Dotted line: reference signal. Solid line: actual output. The sampling period for the output $y(k)$ is 0.1 s.

respectively. As seen from Fig. 3(a) and (b), the proposed adaptive control method works well for the sliding window length $L = 10$ whereas the sliding window length $L = 3$ is not sufficient to determine an acceptable set of six controller parameters.

On the other hand, the effect of the disturbance on the control input is also measured in terms of the NMSE between the control inputs produced with and without disturbance as: NMSE $= 0.0531$ for $K = 3$ and $L = 10$. It is concluded that there is an effect of the torque disturbance on the control input, meaning that the control input is adjusted via the proposed adaptive algorithm to take an action for suppressing the unwanted effect of the disturbance on tracking performance.

Experiment II (Fourth-Order DC Motor Model): This experiment is designed to test particularly the tracking performance of the proposed controller design for a relatively higher order dc motor model. The following linear fourth-order dc motor model [23] is chosen to be controlled by the proposed adaptive control method:

$$\begin{aligned} y(k) = {} & 0.07831y(k-1) - 0.4989y(k-2) + 0.4058y(k-3) \\ & + 0.1755y(k-4) + 0.0618u(k-1) + 0.0086u(k-2) \\ & + 0.0387u(k-3) \end{aligned} \tag{23}$$

where $u(k)$ is the dc motor voltage and $y(k)$ is the dc motor angular speed [23]. The proposed adaptive controller and the chosen plant are both implemented as software simulations in the S-S-S mode of CDTRP [9] with choosing the sampling frequency as $f = 10$ Hz.

To see clearly the effect of the tracking window size L on the tracking performance of the proposed adaptive controller, the identification phase of the method is made inactive so that the fourth-order LTI ARMA model in (23) is used as the fixed plant model along controlling this fourth-order dc motor model by the proposed adaptive controller. The controller design

employs the second-order six-parameter ARMA controller in (21), so the resulting closed-loop system model becomes a second-order ARMA model in (22) The closed-loop (Schur stable) ARMA system parameters α_0, α_1, α_2, β_0, β_1, and β_2 are learned as in the first experiment via the minimization of the regularized tracking error in (14) under the Schur stability constraints of (18) for the L input-desired output data $\{(r[k-s,N], y_d[k-s,N]\}_{s=0}^{L-1}$. The optimal controller parameters c_0, c_1, c_2, d_0, d_1, and d_2 are then calculated as in the first experiment.

In the experiment, the desired trajectory for the dc motor angular frequency is chosen as the sine wave $r(k) = 0.3\text{Sin}(2\pi fk)$ with $f = 0.2$ Hz, which is borrowed from [22] and the sliding window length L is chosen equal to 3 and 10, respectively. The Euclidean ℓ_2 error measure without having ε-insensitiveness is used in both of the identification and tracking error minimization.

The resulting tracking performances of the adaptive controller for the tracking window lengths $L = 3$ and $L = 10$ are shown in Fig. 4(a) and (b), respectively. As seen from Fig. 4(a) and (b), the proposed adaptive control method works well for the tracking window length $L = 10$ due to the existence of sufficient number of data to determine acceptable values for six controller parameters.

Experiment III (Tracking Performance of the Proposed Method for Inverted Pendulum Models): The proposed control method is applied to a modified (time varying) linear inverted pendulum model and also on a nonlinear (time invariant) inverted pendulum model. These applications are implemented to test the performance of the proposed method on: 1) a plant with multiplicative time-varying parameter; 2) an unstable system for which having a short response time of the controller as a consequence of the computational load for determining the control input due to an output error is crucial; and 3) a nonlinear plant. All of the implementations are conducted in the S-E-S real-time operation mode [9] with a sampling period of $T = 0.15$ s where the controller is realized in the PC as software and the plant is the emulated dc motor model.

It is known [24] that, if the angle of the rotation of the pendulum is of concern and the pendulum angle and the angular velocity are small, the dynamics of an inverted pendulum with 1 degree of freedom around the vertical position can be described by a 2-D LTI state equation model given in

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t) \\ \frac{dx_2(t)}{dt} &= \frac{M+m}{Ml} g x_1(t) - \frac{1}{Ml} u(t) \\ y(t) &= x_1(t) \end{aligned} \tag{24}$$

where $u(k)$ is the control input, $y(k)$ is the angle of the rotation of the pendulum rod about pivot point (i.e., pendulum angle), $x_1(k)$ and $x_2(k)$ are pendulum angle and angle velocity of the pendulum, respectively. For mass as $M = 2$ kg, mass of the ball as $m = 0.1$ kg, length of the rod as $l = 0.5$ m, and the gravity acceleration as $g = 9.81$ kg m/s^2 [24], the state model in (24) defines the following ARMA input and output model when backward Euler discretization with $T = 0.15$ s sampling time is used:

$$y(k) = 3.728y(k-1) - 1.864y(k-2) - 0.041u(k). \tag{25}$$

To test the proposed control algorithm on a pendulum model with a multiplicative time-varying parameter, the ARMA model in (25) is modified by introducing a disturbance $\Delta(k) = 0.4\text{Sin}(2\pi fk)$ with $f = 0.25$ Hz as the coefficient of $y(k-1)$, so the following time-varying pendulum model is obtained:

$$y(k) = 3.728\Delta(k)y(k-1) - 1.864y(k-2) - 0.041u(k). \tag{26}$$

In order to control the obtained second-order plants in (25) and (26), the second-order ARMA controller in (21) is chosen. So, the resulting closed loop-system model becomes also a second-order ARMA model in (22) and then the closed-loop ARMA system parameters α_0, α_1, α_2, β_0, β_1, and β_2 are found by minimizing the regularized tracking error in (14) under the Schur stability constraints of (18). In learning the closed-loop model parameters, $L = 10$ input-desired output data $\{(r[k-s,N], y_d[k-s,N]\}_{s=0}^{9}$ are used. In the experiment, the pendulum angle is desired to be zero, i.e., the desired trajectory is $r(k) = 0$. The identification window length is chosen as $K = 3$ and the controller design window length L is chosen equal to 10. The Euclidean ℓ_2 error measure without having ε-insensitiveness is used in both of the identification and tracking error minimization.

The resulting tracking performance of the adaptive controller for the time-invariant pendulum model in (25) is shown in Fig. 5(a). As shown in Fig. 5(b), the performance of the proposed adaptive controller is still acceptable for the time-varying pendulum model in (26) if the amplitude of the time-varying parameter $\Delta(k)$ is not larger than 0.4; it is observed in the simulations that the larger amplitude values for $\Delta(k)$ cause unstability as driving the pendulum away from the zero angle position.

The performance of the proposed adaptive controller on the nonlinear pendulum model in (27), as shown at the bottom of this page, which is more realistic model from the one in (24) is also tested by choosing again the second-order ARMA models for both of the plant and controller with the window sizes $K = 3$ and $L = 10$. To obtain input–output data $\{(u[k-s,N], y_a[k-s,N]\}_{s=0}^{2}$ from the plant, the differential equations describing the model in (27) are solved with the sampling time $T = 0.15$ s in closed loop. As seen from

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t) \\ \frac{dx_2(t)}{dt} &= \frac{-(M+m)\text{gSin}x_1(k) + \text{mlCos}x_1(k)\text{Sin}x_1(k)x_2^2(k) + \text{Cos}x_1(k)u(k)}{\text{mlCos}^2x_1(k) - (M+m)l} \\ y(t) &= x_1(t) \end{aligned} \tag{27}$$

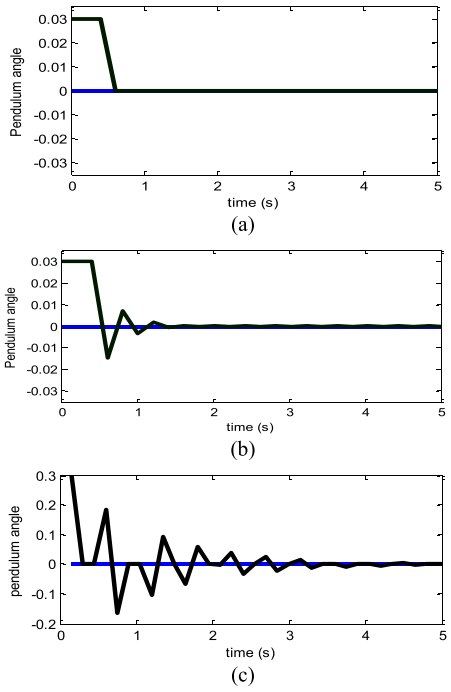

Fig. 5. Tracking results for the pendulum angle $y(k)$ of the inverted pendulum models. Performances of the developed controller for the (a) time invariant model in (25), (b) time-varying model in (26), and (c) nonlinear inverted pendulum model in (27). Dotted line: reference signal. Solid line: actual output. The pendulum angle is in radian.

Fig. 5(c), it is observed that the proposed control algorithm produces a stabilizing control input within a sufficiently short time so capable of restoring the position of the rod even after being disturbed causing a 30° deviation away from the vertical axis.

Experiment IV (Tracking Performance of the Proposed Method With and Without ε-Insensitiveness for a First-Order DC Motor Model): The developed adaptive control method is applied to control a first-order dc motor model described in [9] in order to test the performance of the proposed adaptive ARMA controller under disturbance when ε-insensitiveness is available and not available in the identification and tracking error measures.

The dc motor benchmark plant model is controlled by the proposed adaptive ARMA controller in the S-E-S real-time operation mode [9] where the controller is realized in the PC as software but the plant is the emulated dc motor model in

$$y(k) = 0.25y(k-1) + 318.5u(k) \tag{28}$$

where $u(k)$ is the dc motor voltage, $y(k)$ is the dc motor angular speed, and $T = 0.15$ s is the sampling time [9]. A first-order LTI ARMA model is used as the fixed plant model along controlling this dc motor model by the proposed adaptive controller. As the controller, the second-order ARMA controller in (21) is chosen. So, the resulting closed-loop system model becomes also a second-order ARMA model in (22) and then the closed-loop ARMA system parameters α_0, α_1, α_2, β_0, β_1, and β_2 are found by minimizing the regularized tracking error in (14) under the Schur stability constraints of (18). In learning the closed-loop model parameters, $L = 10$ input-desired output data $\{(r[k-s, N], y_d[k-s, N]\}_{s=0}^{9}$ are

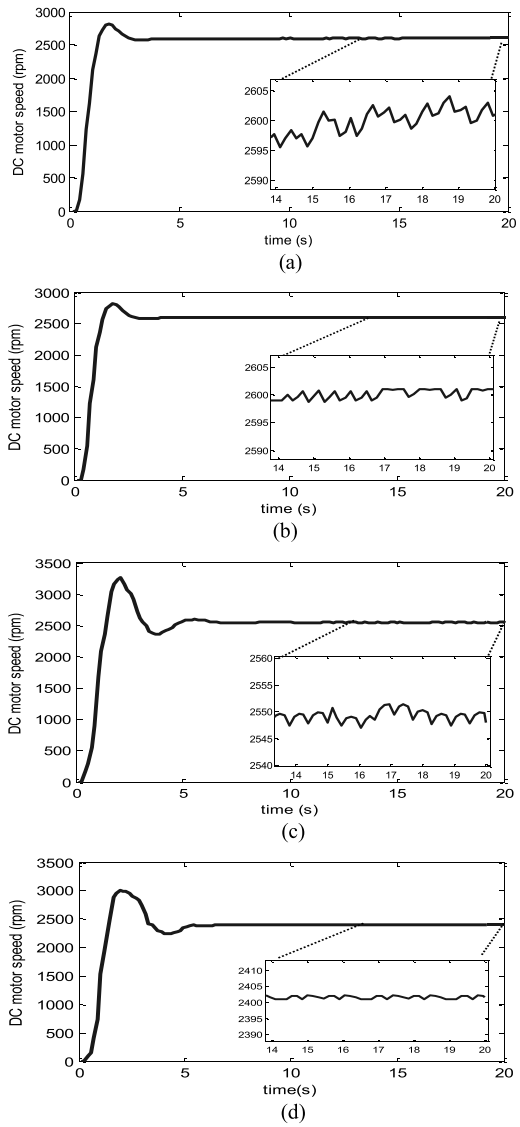

Fig. 6. Angular speed $y(k)$ of the dc motor when the proposed adaptive control method is applied with the sliding window length L chosen as 10. (a) Without disturbance and without ε-insensitiveness. (b) Without disturbance and with ε-insensitiveness. (c) With disturbance and without ε-insensitiveness. (d) With disturbance and with ε-insensitiveness.

used. The optimal controller parameters c_0, c_1, c_2, d_0, d_1, and d_2 are obtained by solving the linear algebraic equations in (8)–(12) with the found optimal closed-loop model parameters. In the first stage, the Euclidean squared error ℓ_2 with no ε-insensitiveness is used in both of the identification and tracking error minimization phases. The desired trajectory for the dc motor angular speed in both of the design and test is chosen as $r(k) = 2500$ r/min.

As seen from Fig. 6(a), the performance of the proposed adaptive controller is acceptable under no disturbance even when ε-insensitiveness is not available. However, the steady-state angular speed contains a fluctuation with the amplitude of 5–10 r/min which is visible by zooming out the angular speed. It is interesting to note that the period of the fluctuation is identical to the tracking window length L. To test the effect of ε-insensitiveness, 50 r/min-insensitiveness is then employed in the loss function $\ell_\varepsilon(\circ, \circ)$. Then, as seen from Fig. 6(b), the

TABLE I
COMPARISONS OF TRACKING ERROR NMSE PERFORMANCES WITH AND WITHOUT DISTURBANCES VERSUS ε-INSENSITIVENESS

$\varepsilon(rpm)$ Tracking Error NMSE	0	25	50	100	200
Without disturbance	0.392	0.252	0.133	0.503	0.739
With disturbance	0.481	0.297	0.226	0.628	0.810

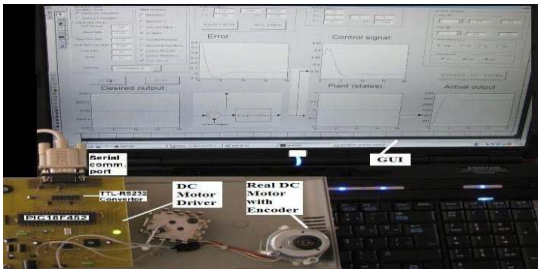

Fig. 7. Real dc motor with electronic driver card and its GUI on the computer.

steady-state tracking performance of the proposed adaptive controller is much better with ε-insensitiveness (under no disturbance). Such an ε-insensitiveness reduces the amplitude of the fluctuation down to 1 and 2 r/min level, on the contrary it increases the frequency of the fluctuation. It is observed in the simulations that choosing length of ε-insensitiveness band larger than 50 r/min causes large deviations in the steady-state angular speed.

In order to test the effect of disturbances, the disturbance $\text{dist}(k) = 0.1\text{Sin}(2\pi fk)$ with $f = 1$ Hz is applied to the control input as an additive external input. As seen in Fig. 6(c), the performance of the proposed adaptive controller shows an acceptable performance even without ε-insensitiveness. Under the applied disturbance, the steady-state angular speed contains a fluctuation with the amplitude of 5–10 r/min again. The effect of disturbance is drastic in the transient regime. As seen from Fig. 6(d), employing 50 r/min-insensitiveness in the loss function $\ell_\varepsilon(\circ, \circ)$ reduces the amplitude of the fluctuation again down to 1 and 2 r/min level and also percent overshoot to 25% level which might be considered as acceptable [24]. As shown in Table I, the effect of the disturbance on the tracking performance of the proposed algorithm can be reduced by choosing a nonzero but not large value of ε insensitiveness.

It should be noted that a steady-state error of 50–100 r/min is observed in the above tests and this error seems to be independent of the choice of the level of ε-insensitiveness. The large steady-state error is actually a consequence of missing integral action term in the controller defined by (21). As will be seen in Experiment V, the steady-state error can be reduced by having an integral action term in the controller which is given in (30) of Experiment V.

Experiment V (Tracking Performance of the Proposed Method for PID Controller Design With ε-Insensitiveness on a Real DC Motor): The fifth experiment is implemented in the prototyping (precisely, S-R-R) mode of CDTRP so that the controller is implemented as software and the dc motor plant shown in Fig. 7 is the real dc motor [9]. In this experiment, the proposed adaptive ARMA controller with six parameters and also a PID controller designed by the Ziegler–Nichols's first method are tested and their tracking performances are compared with each other. As a distinguishing feature, an integral action is included in the ARMA controller, providing relatively small steady-state errors.

In discrete time, the conventional PID controller is written in the following form:

$$u(k) = u(k-1) + \left[K_p + K_i T_s + \frac{K_d}{T_s}\right] e(k) - \left[K_p + 2\frac{K_d}{T_s}\right] e(k-1) + \left[\frac{K_d}{T_s}\right] e(k-2) \quad (29)$$

where $e(k) = r(k) - y(k)$ is the error between desired and actual outputs, K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, and T_s is the sampling time. Six parameters adaptive ARMA controller is written in (30) which is a special case of (4) for determining the conventional PID parameters

$$u(k) = u(k-1) + (c_0 r(k) + d_0 y(k)) + (c_1 r(k-1) + d_1 y(k-1)) + (c_2 r(k-2) + d_2 y(k-2)) \quad (30)$$

where $c_0 = -d_0 = K_p + K_i T_s + K_d/T_s$, $c_1 = -d_1 = -K_p - 2K_d/T_s$, and $c_2 = -d_2 = K_d/T_s$. It is seen that positive K_p, K_i, and K_d parameters are obtained when $d_0 < 0$, $d_1 > 0$, and $d_2 < 0$. So, for an SISO plant having a general ARMA representation in (1), the closed-loop ARMA representation is obtained as follows:

$$\begin{aligned} y(k) &= a_0 y(k) + (1 - a_0 + a_1) y(k-1) \\ &\quad + \sum_{n=2}^{N} (a_n - a_{n-1}) y(k-n) - a_N y(k-N-1) \\ &\quad - c_0 \left[b_0 r(k) + b_1 r(k-1) + \sum_{n=2}^{M} b_n r(k-n) \right] \\ &\quad - d_0 \left[b_0 y(k) + b_1 y(k-1) + \sum_{n=2}^{M} b_n y(k-n) \right] \\ &\quad - c_1 \left[b_0 r(k-1) + \sum_{n=2}^{M} b_{n-1} r(k-n) + b_M r(k-M-1) \right] \\ &\quad - d_1 \left[b_0 y(k-1) + \sum_{n=2}^{M} b_{n-1} y(k-n) + b_M y(k-M-1) \right] \\ &\quad - c_2 \left[\sum_{n=2}^{M} b_{n-2} r(k-n) + b_M r(k-M-2) \right. \\ &\qquad \left. + b_{M-1} r(k-M-1) \right] \\ &\quad - d_2 \left[\sum_{n=2}^{M} b_{n-2} y(k-n) + b_M y(k-M-2) \right]. \end{aligned} \quad (31)$$

TABLE II
COMPARISON OF THE PROPOSED ALGORITHM TO FIVE OTHER OPTIMIZATION ALGORITHMS WITH d_m UPDATE

Matlab Function	Optimization Algorithm	Function Evaluation (Tracking Error)	Iterations	Time (s)	K_p	K_i	K_d
lsglin	Linear Least-Squares	2.4360e-07	3	0.5156	2.0092	4.0736	0.2500
fmincon	Interior-Point	1.1314e-13	10	3.0313	9.1316	41.6534	1.2283
fmincon	Quasi-Newton Line Search	6.2546e-15	1	0.4063	3.0000	12.000	0.2500
fsolve	Levenberg-Marquardt	3.2572e-11	1	0.3281	3.0000	12.000	0.2500
fminsearch	Support Vector Machine	2.9878e-12	1	0.8281	3.0000	12.000	0.2500

TABLE III
COMPARISON OF THE PROPOSED ALGORITHM TO FIVE OTHER OPTIMIZATION ALGORITHMS WITH α_n UPDATE

Matlab Function	Optimization Algorithm	Function Evaluation (Tracking Error)	Iterations	Time (s)	K_p	K_i	K_d
lsglin	Linear Least-Squares	1.1902e-08	1	0.5313	0.2225	5.4420	0.1221
fmincon	Interior-Point	1.6154e-14	11	1.6094	0.6064	23.5777	0.4640
fmincon	Quasi-Newton Line Search	6.2546e-15	1	0.4063	0.1375	0.2544	0.0185
fsolve	Levenberg-Marquardt	3.2572e-11	1	0.3594	0.1375	0.2544	0.0185
fminsearch	Support Vector Machine	2.9878e-12	1	2.2969	0.1375	0.2539	0.0185

In the identification phase, the ARMA identification model parameters a_0, a_1, and b_0 for the plant model $y(k) = a_0 y(k) + a_1 y(k-1) + b_0 u(k)$ are learned in each sliding window with length K by minimizing the regularized identification error defined with ε-insensitive $\ell_\varepsilon(\circ, \circ)$ loss function in (13) for K input–output data $\{(u[k-s, N], y[k-s, N]\}_{s=0}^{K-1}$ measured from the real plant. Where $u(k)$ is the control input of the real dc motor and $y(k)$ is the dc motor angular speed (r/min). The ARMA plant model parameters found in this way are then held constant along the controller design phase.

In the controller design phase, a second-order ARMA controller defined with the parameters c_0, c_1, c_2, d_0, d_1, and d_2 is chosen. The resulting closed-loop system model becomes a second-order ARMA model in (21). The closed-loop (Schur stable) ARMA system parameters α_0, α_1, α_2, β_0, β_1, and β_2 are learned via the minimization of the regularized tracking error in (14) under the Schur stability constraints of (18) for the L input-desired output data $\{(r[k-s, N], y_d[k-s, N]\}_{s=0}^{L-1}$. The optimal controller parameters c_0, c_1, c_2, d_0, d_1, and d_2 are then calculated in a unique way by solving the linear algebraic equations in (8)–(12) with considering $a_2 = b_1 = b_2 = 0$ for the considered plant model and $f_0 = -1$.

In the experiments, the desired trajectory for the dc motor angular speed is chosen as $r(k) = 2500$ r/min and the sliding window lengths K is chosen as 3 and L is also chosen as 10. To find optimal adaptive ARMA controller parameters, the optimization toolbox in MATLAB is used. The Euclidean ℓ_2 loss function with an ε-insensitiveness of 50 r/min is used both for the identification and tracking error measures.

In order to find PID parameters by the algorithm given with the pseudocodes in Algorithm 1, five different optimization algorithms from MATLAB functions, namely, linear least squares, interior-point, quasi-Newton line search, Levenberg–Marquardt, and support vector machine (SVM) are employed in the experiments for the minimization of the regularized loss functions in (13) and then (14) under the Schur stability constraints in (18). In addition to the Schur stability constraints in (18), $d_0 < 0$, $d_1 > 0$, and $d_2 < 0$ constraints are also imposed to have positive K_p, K_i, and K_d parameters.

The results obtained by five different methods are compared with each other in terms of tracking error and computational cost, and given in Tables II and III. Table II is obtained for implementing the tracking error minimization over d_m while Table III is for the minimization over α_n. In the minimization over α_n, the constraints $d_0 < 0$, $d_1 > 0$, and $d_2 < 0$ providing positive K_p, K_i, and K_d PID parameters are given in terms of α_n parameters as follows:

$$d_0 = \frac{\alpha_0 - 1 - a_0 f_0}{-b_0} < 0$$
$$d_1 = \frac{-\alpha_1 - a_1 f_0 - b_1 d_0}{b_0} > 0$$
$$d_2 = \frac{-\alpha_2}{b_0} < 0. \tag{32}$$

Tables II and III provide the average values of the ε-insensitive ℓ_2 loss other than SVM which uses ε-insensitive ℓ_1 loss, the number of iterations and the time spent in a typical window with length $L = 10$ for five different algorithms used in the tracking error minimization phase. The minimizations have been realized by all of the optimization algorithms available in MATLAB numerical software. Only the functional ones are presented in the revised Tables II and III.

The minimization results show that, except for the interior point method, all of linear least squares, quasi-Newton line search, Levenberg–Marquardt, and SVM methods needs a computation time less than 1500 ms in total for the identification and controller design for a specific sliding window, meaning they are feasible since these computations are completed within a duration less than the specified

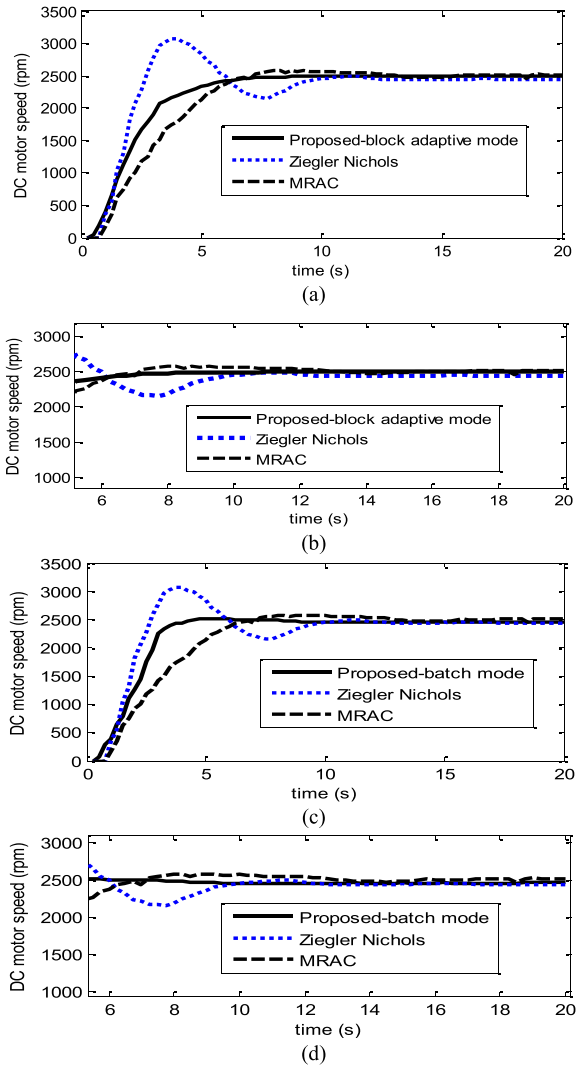

Fig. 8. Angular speed $y(k)$ of the real dc motor when the proposed adaptive control method with ε-insensitiveness of 50 r/min is applied with (a) sliding window length L chosen as 10, (b) zoomed-in view snapshot of (a) during the steady state, (c) batch mode length L chosen as 150, and (d) zoomed-in view snapshot of (c) during the steady state. The steady-state error is 2.5 r/min for $L = 10$ and 33 r/min for $L = 150$.

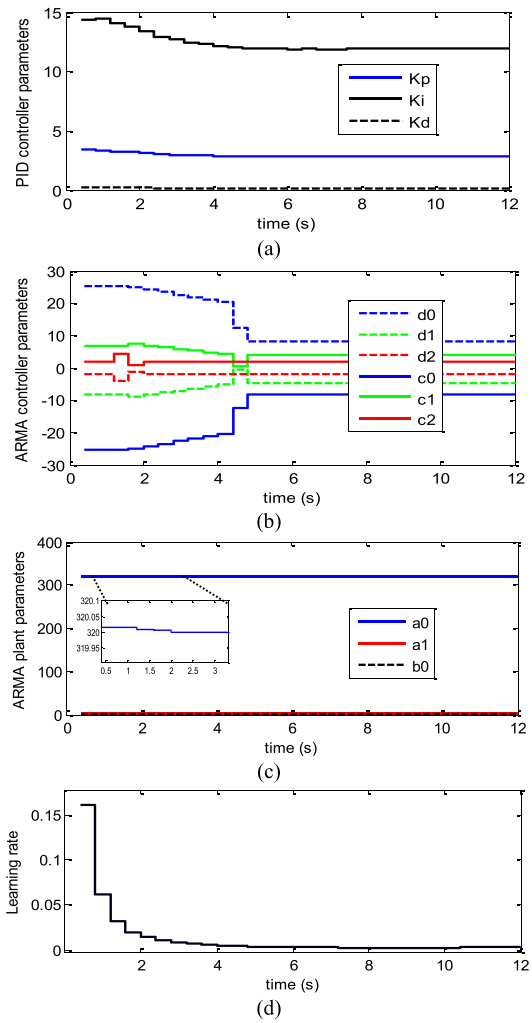

Fig. 9. Time evolution of parameters of the real dc motor when the proposed adaptive control method with ε-insensitiveness of 50 r/min is applied with the sliding window length L chosen as 10 for (a) PID, (b) ARMA controller, and (c) ARMA plant model. (d) Time evolution of the learning rate.

window length of 1500 ms. It should be noted that the plant and controller parameters are updated whenever their offline computations are ended. Their updates may occur even within a sampling period, or within or not within the time durations specified by the associated window lengths. If the offline computation of the set of plant parameters or the set of control parameters is not completed within a sampling period, then the upcoming (sample) data will not be considered in the computations of the parameters, so this data and the other ones coming after it until the time instant at which the computation is ended will be missed to be considered in parameter update processes.

It should be noted that the identification error minimization is realized by the same method, which is used in the tracking error minimization. The time consumptions for the identification are observed to spend negligible times around 0.2 s for the sliding window length K chosen as 3 since the Schur stability constraints are not imposed on the identification model.

It is seen from Table III, the controller design based on the minimization over α_n takes longer time duration due to the computations needed to determine d_0, d_1, and d_2 after obtaining optimal values over α_n; actually such a design procedure is not appropriate for online learning of the controller parameters in a block adaptive mode with length $L = 10$ since the computation time usually exceeds the window duration of 1.5 s for the considered sampling period of 150 ms.

The performance of the real dc motor for the proposed method employing the Levenberg–Marquardt algorithm in the minimization is compared with the performance obtained for the MRAC and the Ziegler–Nichols's first method. The angular speed of the motor measured by an encoder is seen in Fig. 8(a)–(d). Fig. 8(a) and (b) represents the measured speed for the sliding window length L chosen as 10 in comparison with the results by the MRAC and the Ziegler–Nichols's first method. Fig. 8(c) and (d) represents the measured speed for the batch mode which is actually with the length $L = 150$ and the results by the MRAC and the Ziegler–Nichols's first method. It is observed that the proposed block adaptive learning method with the sliding window length $L = 10$ shows a better tracking performance in comparison with the PID controller designed by the Ziegler–Nichols's first method and also to the MRAC. Herein, the PID parameters

designed by the Ziegler–Nichols's first method are chosen as $K_p = 54.54$, $K_i = 2479.3$, and $K_d = 0.3$ which are calculated from the data obtained by the measurements on the same real dc motor done with the work presented in [9].

Time evolution of the PID, ARMA controller and respectively ARMA plant model parameters of the real dc motor are depicted in Fig. 9(a), (b), and (c), respectively for the proposed adaptive control method applied with ε-insensitiveness of 50 r/min and with the sliding window length L of 10. The final values of the PID parameters computed by the proposed block adaptive control algorithm are given at the last three columns of Tables II and III. To monitor the convergence of the learning process when linear least square or Levenberg–Marquardt minimization algorithms are used, the time evolution of the learning rate which is computed in terms of the inverse of the maximum eigenvalue of the covariance matrix related to the data in the sliding window is determined by the maxlinlr command available in MATLAB and given in Fig. 9(d).

V. Conclusion

An online block adaptive learning algorithm for adaptive ARMA controller design is presented in this paper. The algorithm is implemented in two phases. The first phase identifies a plant model within a sliding window by using the data measured from the plant in the same window. The controller parameters are then determined in the second phase from the reference and actual output data within each (sliding) window whose length chosen larger than the one for the identification window. The tasks realized in these two phases are indeed the minimizations of the identification and tracking errors which are chosen as the ε-insensitive ℓ_2 or ℓ_1 loss functions. The main distinguishing feature of the proposed controller-learning algorithm is the guarantee of the Schur stability of the closed-loop system.

Online minimization algorithms such as pattern mode least mean square algorithm may also be used with choosing window length of 1 in the proposed online adaptive control algorithm. However, such a choice may, in general, yields a poor plant model in the identification due to the insufficient information carried by a single data about the general input–output relation of the plant and also brings sensitive dependence to the noise contaminating the measured data in determining controller parameters. Another possibility of applying a pattern mode of update in the proposed adaptive algorithm is to learn plant parameters and controller parameters by a pattern mode algorithm for the set of data available in a chosen sliding window with length K greater than 1. However, it is predictable that the performance of a pattern mode algorithm is poorer than its batch mode counterpart when the number of training samples is low. Considering the above facts, a block adaptive control algorithm is preferred in the paper such that the plant and then the controller parameters are updated whenever their offline computations in terms of a finite number of data available in the current window are ended.

There are some tradeoffs in applying the developed online block adaptive control algorithm. For a large window length, the number of data becomes high, so do the identification and tracking performances for the current window. However, the total processing time needed for computing the plant and the controller parameters may exceed the sampling period even the duration specified by the length of the window, so then the upcoming (sample) data and the other ones coming after it until the time instant at which the computation is ended will be missed to be considered in parameter update processes. On the other hand, the higher the sampling frequency for the overall control system the higher the number of data available in the window, so the better the identification and tracking performances, but again processing load for high number of sample data may cause exceeding the time limit in the window. Considering these facts, one may conclude that the proposed online block adaptive control algorithm in the presented form is more suitable for relatively slow systems requiring moderate or low sampling frequencies and also slowly changing systems. The performance of the proposed algorithm can be increased by overcoming the mentioned computational problems with some techniques such as employing parallel computing for succeeding sliding windows not to miss any data and also introducing the time efficient minimization algorithms.

The developed online controller design algorithm is tested on an emulated dc motor and also on a real dc motor as well as benchmark plant models. The introduced online block adaptive learning of controller has the potential of being used efficiently and successfully in real-world applications for time varying, unstable, and nonlinear plants, especially the ones possessing low order dynamics and/or slowly changing whose model identification can be completed within small time intervals.

References

[1] K. J. Åström and B. Wittenmark, *Adaptive Control*. Boston, MA, USA: Longman, 1994.

[2] Q. Zhao, H. Xu, and S. Jagannathan, "Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 486–499, Mar. 2015.

[3] F. Blanchini, T. Parisini, F. A. Pellegrino, and G. Pin, "High-gain adaptive control: A derivative-based approach," *IEEE Trans. Autom. Control*, vol. 54, no. 9, pp. 2164–2169, Sep. 2009.

[4] G. Battistelli, J. P. Hespanha, E. Mosca, and P. Tesi, "Model-free adaptive switching control of time-varying plants," *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 1208–1220, May 2013.

[5] T. Waegeman, F. Wyffels, and B. Schrauwen, "Feedback control by online learning an inverse model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1637–1648, Oct. 2012.

[6] F.-Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[7] Y. Zhu and Z. Hou, "Data-driven MFAC for a class of discrete-time nonlinear systems with RBFNN," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 1013–1020, May 2014.

[8] A. J. Smola, N. Murata, B. Schölkopf, and K.-R. Müller, "Asymptotically optimal choice of ε–loss for support vector machines," in *Proc. 8th Int. Conf. Artif. Neural Netw. Perspect. Neural Comput.*, Skövde, Sweden, Sep. 1998, pp. 105–110.

[9] S. Şahin, Y. İşler, and C. Güzeliş, "A microcontroller based test platform for controller design," in *Proc. IEEE Int. Symp. Ind. Electron.*, Bari, Italy, Jul. 2010, pp. 36–41.

[10] K. J. Åström and T. Hägglund, "Revisiting the Ziegler–Nichols step response method for PID control," *J. Process Control*, vol. 14, no. 6, pp. 635–650, Sep. 2004.

[11] A. R. Benaskeur and A. Desbiens, "Backstepping-based adaptive PID control," *IEE Proc. Control Theory Appl.*, vol. 149, no. 1, pp. 54–59, Jan. 2002.
[12] 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.
[13] S. Skoczowski, S. Domek, K. Pietrusewicz, and B. Broel-Plater, "A method for improving the robustness of PID control," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1669–1676, Dec. 2005.
[14] H. Zhang, Y. Shi, and A. S. Mehr, "Robust $\mathcal{H}_\infty$ PID control for multivariable networked control systems with disturbance/noise attenuation," *Int. J. Robust Nonlinear Control*, vol. 22, no. 2, pp. 183–204, Jan. 2012.
[15] T. Yamamoto, K. Takao, and T. Yamada, "Design of a data-driven PID controller," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 1, pp. 29–39, Jan. 2009.
[16] A. S. Zayed, A. Hussain, and R. A. Abdullah, "A novel multiple-controller incorporating a radial basis function neural network based generalized learning model," *Neurocomputing*, vol. 69, nos. 16–18, pp. 1868–1881, Oct. 2006.
[17] S.-M. Baek and T.-Y. Kuc, "An adaptive PID learning control of DC motors," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Comput. Cybern. Simulation*, vol. 3. Orlando, FL, USA, Oct. 1997, pp. 2877–2882.
[18] R. Cao, Z. Hou, and W. Zhang, "The model-free learning enhanced motion control of DC motor," in *Proc. Int. Conf. Electr. Mach. Syst.*, Seoul, Korea, Oct. 2008, pp. 1222–1226.
[19] S. Ang, C. Rongmin, and Z. Huixing, "The linear motor control based on ELVIS," in *Proc. 24th Chin. Control Decision Conf.*, Taiyuan, China, May 2012, pp. 2505–2509.
[20] D. Salomonsson, L. Söder, and A. Sannino, "An adaptive control system for a dc microgrid for data centers," *IEEE Trans. Ind. Appl.*, vol. 44, no. 6, pp. 1910–1917, Nov./Dec. 2008.
[21] G. Keqin, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. New York, NY, USA: Springer-Verlag, 2003.
[22] C. Canudas, K. J. Åström, and K. Braun, "Adaptive friction compensation in dc-motor drives," *IEEE J. Robot. Autom.*, vol. 3, no. 6, pp. 681–685, Dec. 1987.
[23] T. Tutunji, M. Molhim, and E. Turki, "Mechatronic systems identification using an impulse response recursive algorithm," *Simul. Model. Pract. Theory*, vol. 15, no. 8, pp. 970–988, Sep. 2007.
[24] K. Ogata, *Modern Control Engineering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.

Savaş Şahin received the B.Sc. degree in electronics and communication engineering from Kocaeli University, Kocaeli, Turkey, in 1996, the M.Sc. degree from the Department of Electrical and Electronics Engineering, Ege University, İzmir, Turkey, in 2003, and the Ph.D. degree from the Department of Electrical and Electronics Engineering, Dokuz Eylül University, İzmir, in 2010.

He was an Instructor with the Department of Control and Automation, Ege Vocational School, Ege University, from 2000 to 2012. He has been an Assistant Professor with the Department of Electrical and Electronics Engineering, İzmir Katip Çelebi University, İzmir, since 2012. He was a Visiting Researcher with the Delft Center for System and Control, Delft University of Technology, Delft, The Netherlands, in 2013, for three months. He is involved in several projects on controller design and engineering education. His current research interests include control systems, industrial automation, chaotic systems, and artificial neural networks.

Cüneyt Güzeliş received the B.Sc., M.Sc., and Ph.D. degrees from Istanbul Technical University, Istanbul, Turkey, in 1981, 1984, and 1988, respectively, all in electrical engineering.

He was with Istanbul Technical University from 1982 to 2000, where he became a Full Professor. He was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, from 1989 to 1991, as a Visiting Researcher and Lecturer. He was with the Department of Electrical and Electronics Engineering, Dokuz Eylül University, İzmir, Turkey, from 2000 to 2011. He was with the Department of Electrical and Electronics Engineering, Faculty of Engineering and Computer Sciences, İzmir University of Economics, İzmir, from 2011 to 2015. He is currently with the Department of Electrical and Electronics Engineering, Faculty of Engineering, Yaşar University, İzmir. His current research interests include artificial neural networks, biomedical signal and image processing, nonlinear circuits-systems, and control, and educational systems.