Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **4**, November **2021**

Robust Digital-Redesign Tracking control for Uncertain Systems: PID sliding mode Control and PSO Algorithm

Jun-Juh Yan[1*], Jiunn-Shiou Fang [2], Jason Sheng-Hong Tsai[2], Chiao-Hsin Huang [2] and Shu-Mei Guo[3]

[1]Department of Electronic Engineering
National Chin-Yi University of Technology
Taichung 41107,Taiwan

[2]Department of Electrical Engineering
National Cheng-Kung University
Tainan 701, Taiwan

[3]Department of Computer Science and Information Engineering
National Cheng-Kung University
Tainan 701, Taiwan

*Corresponding Author: jjyan@stu.edu.tw

Received January 2021 Revised March 2021

Abstract. *A digital-redesigned robust tracker for sampled-data systems with bounded nonlinear disturbances is newly proposed based on an integration of the proportional-integral-derivative (PID) control, sliding mode control (SMC) and particle swarm optimization (PSO) algorithm. First, a new control method integrating the PID control, SMC and PSO algorithm for continuous-time systems with bounded nonlinear disturbances is newly proposed. Then, the newly developed digital-redesigned robust tracker for the sampled-data systems with disturbances is proposed. Except for the well-known superiorities of the PID controller, the integrated SMC is applied to suppress the unknown matched nonlinear disturbances. Finally, the PSO algorithm is utilized to optimally tune control parameters, such that the proposed approach achieves a satisfactory performance for the robust tracking control.*
Keywords: Digital redesign, PID controller, sliding mode control, PSO algorithm, tracker design.

1. **Introduction.** The PID control is one of the general popular control strategies due to its simple structure. Hence, PID control has been widely applied in many real industry systems [1, 2] In order to deal with some problems of PID controller, the traditional methods are proposed such as gain-phase margin, root locus, etc. [1]. In general, when the nonlinear vector exists in the controlled system, the traditional methods become difficult to find an optimal solution or a near-optimal solution [1, 2]. In the report [3], the D-type controller is proposed, and the stability of the closed-loop controlled system is discussed. The authors [4] illustrates that it is difficult to process the D-type controller by using the traditional state-feedback approach; hence, the Frobenius canonical form and pole-placement method are applied to the D-type controller design. Therefore, in this paper, we aim to propose a simple D-type design algorithm for PID controller design.

While nonlinear perturbation exists in the controlled system, the design of PID controller will be a challenge. To cope with this problem, we propose a new SMC-based PID

controller design approach for the controlled system with nonlinear perturbations. SMC is typically used for suppressing the bounded external disturbance. However, the chattering phenomenon might occur, which is a negative effect on the controlled system. The undesired chattering phenomenon should be reduced such that the sliding mode control force can be implemented. In some literatures, the smooth functions such as saturation function ‘ [5] and the scalar sign function [6] are utilized to overcome the chattering phenomenon. The integral SMC and adaptive control are proposed in literature [7]. The SMC-based adaptive PID controller is proposed for the chaotic system with uncertain disturbances [8]. The literature [9] adopts the predictive sliding mode tracking control to process the uncertain Steer-by-Wire system. The report [10] presents the discrete-time sliding mode controller integrating with state estimator and disturbances observer for the multiple-input/multiple-output controlled system, and the discrete-time sliding manifold is discussed. By applying SMC to cope with the bounded nonlinear disturbances, the system resembles a linear system such that the PSO algorithm can be applied to find the optimal solution or near-optimal solution for the PID controller design.

In the field of control systems, there are lots of literatures which discuss continuous-time controllers to control continuous systems; or discuss discrete-time controllers to control discrete systems. Besides these methods, digital redesign is another control method. In the digital redesign method, we construct an equivalent discrete controller to control the original continuous systems system. In contrast to directly design the discrete controllers, the digital redesign method is simpler since the complicated process is reduced. In [11], the synthesized discrete-time SMC confirms the sliding mode reaching a specified discrete-time sliding surface by using the linear matrix inequality approach. The literature [12] presents that the SMC is difficult to implement in a sampled-data system because the control force is zero-order-hold (Z.O.H.) between two sampling times. By applying Euler's approximation method, the continuous-time model can be discretized and then the discrete-time SMC is proposed by the discretization model [13]. Therefore, we apply Euler's approximation method into the digital redesign method such that the continuous-time controller is directly transferred into the discrete-time controller in this paper. The associated discussion about digital redesign method is quite less. In the beginning, we can review this literature [14]. The literature [14] explained how to utilize a predictive digital redesign method to design a linear quadratic analog tracker (LQAT). In the past, the method of directly designing discrete-time controllers for the sampled-data systems was proposed in the literature [15]. The literature [16] mentioned how to improve power system dynamic stability by using digital redesign of SMC. The report [17] discussed that the implementation of the control systems relies more on digital microprocessors and digital computers, so a digital redesign method of sliding mode control based on the LMI approach was proposed by them. Since the digital redesign based on SMC has not been well developed, this paper will further discuss a digital-redesign application in PID controller integrated with SMC-based tracker design.

In recent years, the applications of intelligent control have been widely used. Through the algorithms, the optimization problem can be solved. Particle swarm optimization (PSO) is one of the famous and well-developed algorithms which can find the optimal solutions for parameter design. In this paper, we used this method to find the optimal or near-optimal controller parameters so that the tracking performance can be improved. The history of developing the PSO algorithm can be traced to 1995 [18], the literature introduces the concept for the optimization of nonlinear functions and proposes the particle swarm methodology. The authors of [19] mention that PSO has the advantages of being easy in implementation and having few parameters to be adjusted but has a disadvantage of controlling the balance between the global search and the local search. As a result,

the variants of this original PSO have been widely developed, for example, in [20], the researchers modify the inertia weight and proposes a linearly decreasing inertia weight method (LPSO), which can reduce the iterations on average to find an optimal solution. Different from LPSO, in [21], the authors propose a random inertia weight factor method (RPSO), which can improve the performance for tracking dynamic systems. The literature [22] proposes another important variant of the standard PSO: the constriction factor approach PSO (CPSO), which can increase the ability to find optima of some well-studied test functions. Aside from the variants of PSO mentioned above, the report [19] proposes a multi-swarm cooperative particle swarm optimizer (MCPSO), which contains the slave swarms and the master swarm. Through cooperation between master and slave swarms, the efficiency of finding optimal solutions can be enhanced.

Based on our knowledge, until now, there are still few literature reports discussing the digital redesign of continuous-time sliding mode controllers. Since the chattering phenomenon affects the controlled system output directly when the controlled system contains a direct feed-through term, the SMC-based switching function needs to be selected properly. Therefore, there are some controller parameters which need to be adjusted properly, therefore, the MCPSO algorithm is introduced to cope with the controller parameters optimization; hence, the performance could be improved. According to the above description, we have developed the PID controller integrated with SMC-based tracker design, and we can discretize the tracker by using the digital redesign method for the sampled-data systems. By applying the digital redesign method, the continuous-time controller can be transferred to the discrete-time controller without losing the performance of the original continuous-time controller. The negative effect caused by bounded nonlinear disturbances can be reduced effectively by applying the SMC. In addition, the application of MCPSO algorithm can provide the optimal solution or near-optimal solution to optimize the performance of the system response. Finally, we can confirm that our method is correct and contributable through the experimental results.

Notation. w^T is used to denote the transport for a matrix w . $\|\bullet\|$ denotes the Euclidean norm of the vector. I_n is the identity matrix of $n \times n$. $|\bullet|$ represents the absolute value. $w^\dagger$ denotes to the pseudo inverse for a matrix w. $\text{sgn}(w) = [\text{sgn}(w_1), \text{sgn}(w_2), \cdots, \text{sgn}(w_m)]^T \in \Re^m$ and $\text{sgn}(w)$ is the sign function of w , if $w > 0$,$\text{sgn}(w) = 1$; if $w < 0$, $\text{sgn}(w) = -1$.

2. Digital Redesign Robust Tracker for Sampled-Data Systems with Nonlinear Disturbances Based on PID and SMC.

2.1. Derivation of LQAT-based PID controller.

Consider the controllable time-invariant system with a direct feed-through term and bounded nonlinear disturbance which occurs at the plant input described by

$$\dot{x}_c(t) = Ax_c(t) + B\left[u_c(t) + d\left(x_c(t), t\right)\right], \tag{1}$$

$$y_c(t) = Cx_c(t) + D\left[u_c(t) + d\left(x_c(t), t\right)\right], \tag{2}$$

where $A \in \Re^{n\times n}$,$B \in \Re^{n\times m}$,$C \in \Re^{p\times n}$ and $D \in \Re^{p\times m}$ are the system matrices, $x_c(t) \in \Re^n$ is the state vector, and $d\left(x_c(t), t\right) \in \Re^m$ is the bounded nonlinear disturbance which satisfies $\|d\left(x_c(t), t\right)\| \le d^u \|x_c(t)\|$ (d^u denotes the upper bound coefficient), $y_c(t) \in \Re^p$ is the output vector, and $u_c(t) \in \Re^m$ is the input vector. Note that $u_c(t) = u_c^*(t) + K_d\dot{x}_c(t)$, where K_d is the D-type controller gain at time t and $u_c^*(t)$ will be determined later.

The closed-loop stability of D-type controller is discussed in [3]. According to the result, the linear transformation can be presented, and the derivative term $\dot{x}_c(t)$ can be merged. (1) can be rewritten as

$$\left(I_n - BK_d\right)\dot{x}_c(t) = Ax_c(t) + B\left[u_c^*(t) + d\left(x_c(t), t\right)\right], \tag{3}$$

where $M = I_n - BK_d$. Thus, a new state space equation can be described by

$$\dot{x}_c(t) = A_{PID}x_c(t) + B_{PID}\left[u_c^*(t) + d\left(x_c(t), t\right)\right], \tag{4}$$

$$y_c(t) = C_{PID}x_c(t) + D_{PID}\left[u_c^*(t) + d\left(x_c(t), t\right)\right], \tag{5}$$

where

$$A_{PID} = M^{-1}A,\ B_{PID} = M^{-1}B,$$
$$C_{PID} = C + DK_dM^{-1}A \text{ and } D_{PID} = D + DK_dM^{-1}B.$$

Here, we neglect the bounded nonlinear disturbance $d\left(x_c(t), t\right)$ temporarily in order to design the PID controller gains. The bounded nonlinear disturbance $d\left(x_c(t), t\right)$ will be eliminated and discussed in the SMC later. Now, (4) and (5) can be rewritten as

$$\dot{x}_c(t) = A_{PID}x_c(t) + B_{PID}u_c^*(t), \tag{6}$$

$$y_c(t) = C_{PID}x_c(t) + D_{PID}u_c^*(t). \tag{7}$$

To avoid the D-type controller gain K_d with unreasonable value, the design algorithm should be discussed properly. In order to make the closed-loop eigenvalues of (6) be negative, let the matrix M satisfy $M = I_n - BK_d \geq \alpha I_n > 0$, where the parameter α is positive so that the transformed system can keep its property. Note that the rank of BK_d is m , which means that $M = I_n - BK_d$ has only m poles which can be placed. Some methods discuss the issue and propose such as pole placement to solve this problem. For the purpose to implement minimal parameters, one solution of K_d can be obtained by

$$K_d = (1 - \alpha)\, B^{\dagger}, \tag{8}$$

then the matrix M can be described as

$$M = I_n - (1 - \alpha)\, BB^{\dagger} > 0, \tag{9}$$

which suggests

$$I_n > (1 - \alpha)\, BB^{\dagger}.$$

In order to know the range of the parameter α , we take 2-norm for $I_n > (1 - \alpha)\, BB^{\dagger}$:

$$\|I_n\| > (1 - \alpha)\left\|BB^{\dagger}\right\|. \tag{10}$$

From (10), the parameter α should satisfy $1 - \alpha > 0$ so that the matrix M would be positive definite matrix. Also, the parameter α should satisfy (9). Therefore, the parameter α has the range $0 < \alpha < 1$. If the parameter α equals to 1, the PID-type controller reduces to the PI-type controller. Furthermore, the inverse of M must exist so that we can apply this transformed matrix to our approach. The following conditions are necessary to be satisfied.

Condition 1. $\left\|BB^{\dagger}\right\| = 1$.

Condition 2. Inverse of $\left(I_n - (\alpha I_n - I_n)BB^{\dagger}\right)$ exists.

Note that we assumed that the matrix $BB^{\dagger}$ is semi-positive definite. Thus, K_d would be a rational matrix as the condition exists. For the above calculation, we can know that the inverse of matrix M exists. With these conditions mentioned above, we can confirm that the transformed matrix is invertible for the linear transformation in our method.

Remark 1. If (8) and the above conditions 1 and 2 are satisfied in the D-type controller design algorithm, the inverse of the matrix M must exist. Since the D-type controller is sensitive to the variance of system states, the gain K_d should be selected properly. If the gain K_d achieved high gain property, the other gains of the controller such as the

P-type controller gain K_p and the I-type controller gain K_i (to be shown later) will be undesirable large, and the control force is huge too. Hence, a simpler D-type controller algorithm is proposed.

After the D-type controller is proposed, we can further construct an augmented matrix associated with $x_c(t)$ and $\int e_y(t)dt$. Let

$$\eta_c(t) = \begin{bmatrix} x_c(t) \\ \int e_y(t)dt \end{bmatrix}$$

be the new state variable, where $e_y(t) = y_c(t) - r_c(t)$ and $r_c(t)$, respectively, denote the tracking error and the desired reference trajectory. According to the new state variable, the system described in (6) and (7) can be rewritten to an augmented system:

$$\dot{\eta}_c(t) = \bar{A}_{PID}\eta_c(t) + \bar{B}_{PID}u_c^*(t) - r_{PID}(t), \tag{11}$$

$$y_c(t) = \bar{C}_{PID}\eta_c(t) + \bar{D}_{PID}u_c^*(t), \tag{12}$$

where

$$\bar{A}_{PID} = \begin{bmatrix} A_{PID} & 0 \\ C_{PID} & 0 \end{bmatrix}, \bar{B}_{PID} = \begin{bmatrix} B_{PID} \\ D_{PID} \end{bmatrix}, \bar{B}_{PID} = \begin{bmatrix} B_{PID} \\ D_{PID} \end{bmatrix},$$

$$\bar{C}_{PID} = \begin{bmatrix} C_{PID} & 0 \end{bmatrix}, \bar{D}_{PID} = D_{PID}, \quad r_{PID}(t) = \begin{bmatrix} 0 \\ r_c(t) \end{bmatrix}.$$

Lemma 1 ([23]). Let $\left(\bar{A}_{PID}, \bar{B}_{PID}\right)$ be the controllable pair of a given open-loop system. The Riccati equation with the matrix P being a solution can be described as follows:

$$\left(\bar{A}_{PID} + hI_n\right)^T P + P\left(\bar{A}_{PID} + hI_n\right) - P\bar{B}_{PID}{R_k}^{-1}\bar{B}_{PID}^T P + Q_k = 0, \tag{13}$$

where $h \geq 0$, and the matrix I_n is an identity matrix. From (13), the eigenvalues of the closed-loop system can be placed on the left of the $-h$ vertical line.

In order to do a tracker design, the PI-type controller gain K_{PI} can be designed by applying the linear quadratic method. Then the controller gain K_{PI} is described as

$$K_{PI} = \begin{bmatrix} K_p & K_i \end{bmatrix} = {R_c}^{-1}\left(\bar{B}_{PID}^T P + N^T\right), \tag{14}$$

where $R_c = R_k + \bar{D}_{PID}^T Q_k \bar{D}_{PID}$, $N = \bar{C}_{PID}^T Q_k \bar{D}_{PID}$ $K_p \in \Re^{m\times n}$ and $K_i \in \Re^{m\times p}$ The LQAT-based PI control law is applied, and it can be described by

$$u_c^*(t) = -K_{PI}\eta_c(t) + E_c r_c(t), \tag{15}$$

where E_c denotes the forward gain (the detail will be shown later).

The purpose of designing a tracker is to minimize the output error $e_y(t) = y_c(t) - r_c(t)$. Once the error converges to zero, it means that the output vector $y_c(t)$ would track the reference trajectory $r_c(t)$. The way to calculate the optimal gain K_p and K_t is to use a linear-quadratic state-feedback regulator with output weighting $Q_k \in \Re^{p\times p}$ and $R_k \in \Re^{m\times m}$. The quadratic cost function with output weighting is defined as

$$J = \frac{1}{2}\int_0^{t_{end}} \left\{ [y_c(\tau) - r_c(\tau)]^T Q_k [y_c(\tau) - r_c(\tau)] + u_c^{*T}(\tau) R_k u_c^*(\tau) \right\} d\tau, \tag{16}$$

where $y_c(\tau)$, $r_c(\tau)$ and $u_c^*(\tau)$ are the output vector, the reference and the control input vector, respectively. t_{end} denotes the final time. $Q_k = 10^q I_p$ is a positive definite or a positive semidefinite real symmetric matrix, where $q \geq 0$. R_k is a positive definite real symmetric matrix.

To calculate a lower value of the system output $y_c(t)$ in (16); hence, we assume $r_c(t) = 0 (r_c(\tau) = 0)$ first, and then final-value theorem can be adopted to minimize

the performance index [6]. According to Lemma 1, the optimal gain K_{PI} in (14) can be obtained by solving the matrix P and $K_{PI} = {R_c}^{-1}\left(\bar{B}_{PID}^T P + N^T\right)$. In order to solve the matrix P , an algebraic Riccati equation is considered:

$$\left(\bar{A}_{PID} + hI_n\right)^T P + P\left(\bar{A}_{PID} + hI_n\right) - \left(\bar{B}_{PID}^T P + N^T\right)^T {R_k}^{-1}\left(\bar{B}_{PID}^T P + N^T\right) + \bar{C}_{PID}^T Q_k \bar{C}_{PID} = 0. \quad (17)$$

Note that the gain K_p and K_i in the optimal gain K_{PI} are determined based on the linear model $\left(\bar{A}_{PID}, \bar{B}_{PID}, \bar{C}_{PID}, \bar{D}_{PID}\right)$. The forward gain E_c can be inferred by applying the final-value theorem.

Form (17), we observer that the traditional LQAT cannot be directly adopted to design the control law $\bar{u}(t)$ due to Lemma 1; hence, the final-value theorem is applied in order to overcome this problem. The forward gain E_c can be determined by considering the augmented matrix in (11). In this case, the integrated term can be neglected according to the following statements.

A linear time-invariant system with the PI-type controller containing an underdetermined term $\bar{u}(t)$ is described by

$$\dot{x}_c(t) = Ax_c(t) + B\left(\bar{u}(t) - K_p x_c(t) - K_i \int e_y(t)dt\right),$$

$$y_c(t) = Cx_c(t) + D\left(\bar{u}(t) - K_p x_c(t) - K_i \int e_y(t)dt\right).$$

Take the Laplace transform of the tracking error $e_y(t) = y_c(t) - r_c(t)$ to obtain the following equations:

$$\begin{aligned} E(s) &= Y(s) - R_s \\ &= \left\{(C - DK_p)\left[sI_n - (A - BK_p)\right]^{-1}B + D\right\}\left(\tfrac{\bar{U}_s}{s} - K_I\tfrac{E(s)}{s}\right) - \tfrac{R_s}{s}, \end{aligned} \quad (18)$$

where $\bar{U}_s$ and R_s are the steady-state values of $\bar{u}(t)$ and $r_c(t)$, respectively. The steady-state values refer to what $\bar{u}(t)$ and $r_c(t)$ change relatively minor than the high-gain-property controlled system dynamics in any time period.

Use the final-value theorem to (18), one has

$$\begin{aligned} \lim_{s\to 0} sE(s) &= \lim_{s\to 0} s\left[W\left(\tfrac{\bar{U}_s}{s} - K_i\tfrac{E(s)}{s}\right) - \tfrac{R_s}{s}\right] \\ &= \lim_{s\to 0}\left[W\left(\bar{U}(s) - K_i E(s)\right) - R_s\right], \end{aligned} \quad (19)$$

where

$$W = (C - DK_p)\left[sI_n - (A - BK_p)\right]^{-1}B + D.$$

After rearranging, (19) can be represented as

$$\lim_{s\to 0}\left(sI_n + K_i W\right)E(s) = \lim_{s\to 0}\left(W\bar{U}_s - R_s\right). \quad (20)$$

From (20), it implies that once $\lim_{s\to 0}\left(W\bar{U}_s - R_s\right) = 0$, then $\lim_{s\to 0} sE(s) = 0$. Thus, we can infer that deriving the controller $\bar{u}(t)$ in (20) can be completed by applying the final-value theorem without considering the integration-term.

According to above discussion, the process of calculating K_{PI} and E_c can be simplified by neglecting the integration-term in the controlled system. In other words, it is available to consider the system described in (6) and (7) instead of (11) and (12). By replacing $u_c^*(t) = -K_p x_c(t) + E_c r_c(t)$ into (6) and (7), these two equations can be rewritten as

$$\dot{x}_c(t) = A_c x_c(t) + B_{PID} E_c r_c(t), \quad (21)$$

$$y_c(t) = C_c x_c(t) + D_{PID} E_c r_c(t), \quad (22)$$

where $A_c = A_{PID} - B_{PID}K_p$ and $C_c = C_{PID} - D_{PID}K_p$. To complete the tracker design, the forward gain can be inferred by applying final-value theorem. Taking the Laplace transform is needed, so (21) and (22) can be described by

$$X(s) = (sI - A_c)^{-1}\left(B_{PID}E_c\frac{R(s)}{s}\right),$$

$$\begin{aligned} Y(s) &= C_cX(s) + D_{PID}E_c\tfrac{R(s)}{s} \\ &= C_c(sI - A_c)^{-1}\left(B_{PID}E_c\tfrac{R(s)}{s}\right) + D_{PID}E_c\tfrac{R(s)}{s}. \end{aligned}$$

Since the tracker would be applied into the system, the final-value of the output should match that of the desired reference trajectory.

According to the Laplace transform of $y_c(t)$, the following tracking error can be implied:

$$y_c(\infty) - r_c(\infty) = \lim_{s\to 0} sY(s) - \lim_{s\to 0} sR(s) = 0, \tag{23}$$

where

$$\begin{aligned} y_c(\infty) &= \lim_{s\to 0} sY(s) \\ &= C_c(-A_c)^{-1}B_{PID}E_cR(s) + D_{PID}E_cR(s)\,|_{s\to 0} \\ &= \left[C_c(-A_c)^{-1}B_{PID} + D_{PID}\right]E_cR(s)\,|_{s\to 0} \end{aligned}$$

and

$$r_c(\infty) = \lim_{s\to 0} s\frac{R(s)}{s} = R(s)\,|_{s\to 0}\ .$$

By comparing $y_c(\infty)$ and $r_c(\infty)$, we have the following condition to satisfy (23):

$$E_c = \left[C_c(-A_c)^{-1}B_{PID} + D_{PID}\right]^{\dagger}. \tag{24}$$

Thus, the proposed PID controller in (15) can improve the output performance by tracking the reference trajectory. Notice that the proposed approach still works for the special case where $y_c(t) = Cx_c(t)$.

2.2. **Derivation of LQAT integrated with PID control and SMC.** Here, we take the disturbance into consideration and apply SMC to suppress the bounded nonlinear disturbance. The system with bounded nonlinear disturbance can be described as

$$\dot{x}_c(t) = Ax_c(t) + B\left[u_c(t) + d\left(x_c(t), t\right)\right], \tag{25}$$

$$y_c(t) = Cx_c(t) + D\left[u_c(t) + d\left(x_c(t), t\right)\right]. \tag{26}$$

The sliding manifold $s_c(t)$ is given as

$$s_c(t) = C_sx_c(t) - \int_0^{t_{end}} (C_sAx_c(t) + u_{Tc}(t))dt, \tag{27}$$

where

$$C_s = B^{\dagger}, \tag{28}$$

and $u_{Tc}(t)$ is the control law. From Section 2.1, we obtain

$$u_{Tc}(t) = -K_px_c(t) - K_i\int e_y(t) + K_d\dot{x}_c(t) + E_cr_c(t). \tag{29}$$

After differentiating $s_c(t)$, we obtain

$$\dot{s}_c(t) = C_sB\left[u_c(t) + d\left(x_c(t), t\right)\right] - u_{Tc}(t). \tag{30}$$

According to (28), (30) can be rewritten as

$$\dot{s}_c(t) = u_c(t) + d\left(x_c(t), t\right) - u_{Tc}(t). \tag{31}$$

The equivalent control $u_{eq}^*(t)$ in the sliding manifold ($\dot{s}_c(t) = 0$) is obtained by

$$u_{eq}^*(t) = u_{Tc}(t) - d\left(x_c(t), t\right). \tag{32}$$

Since the bounded nonlinear disturbance is unknown, a general form of the sliding mode controller is denoted as $u_{\pm} = -\gamma_c \left\|x_c(t)\right\| \operatorname{sgn}(s_c(t)) - \sigma s_c(t)$, and the control law in (31) can be rewritten as

$$u_c(t) = u_{Tc}(t) + u_{\pm} = u_{Tc}(t) - \gamma_c \left\|x_c(t)\right\| \operatorname{sgn}\left(s_c(t)\right) - \sigma s_c(t), \tag{33}$$

where σ is a positive constant which can accelerate the convergence rate of the sliding function (to be shown later) and γ_c is an arbitrary positive value. Thus, (31) can be rewritten as

$$\dot{s}_c(t) = d\left(x_c(t), t\right) - \gamma_c \left\|x_c(t)\right\| sgn\left(s_c(t)\right) - \sigma s_c(t), \tag{34}$$

where γ_c is selected to satisfy $\gamma_c \geq d^u$ such that $s_c(t) = 0$ is reached.

Theorem 1. Once the sliding surface is reached, the unknown bounded nonlinear disturbance can be suppressed.

Proof. Consider a candidate Lyapunov function,

$$V\left(s_c(t)\right) = \frac{1}{2} {s_c}^T(t) s_c(t), \tag{35}$$

and then take the derivative of $V\left(s_c(t)\right)$ in (35), which gives

$$\begin{aligned} \dot{V}\left(s_c(t)\right) &= {s_c}^T(t)\dot{s}_c(t) = {s_c}^T(t)\left(d\left(x_c(t), t\right) - \gamma_c \left\|x_c(t)\right\| \operatorname{sgn}\left(s_c(t)\right) - \sigma s_c(t)\right) \\ &\leq \left\|d\left(x_c(t), t\right)\right\| \left\|s_c(t)\right\| - \gamma_c \left\|x_c(t)\right\| \left\|s_c(t)\right\| - \sigma \left\|s_c(t)\right\|^2 \\ &\leq -\sigma \left\|s_c(t)\right\|^2 \leq 0, \end{aligned} \tag{36}$$

where $\left\|d\left(x_c(t), t\right)\right\| \leq d^u \left\|x_c(t)\right\|$ and $\sigma \geq 0$. A well-selected parameter σ can accelerate $\dot{V}\left(s_c(t)\right)$ to converge. By choosing a proper value of γ_c such that $\gamma_c \geq d^u \geq 0$, then $\dot{V}\left(s_c(t)\right) \leq 0$ is satisfied. Thus, the sliding mode function in (35) converges to zero and $u_c(t) = u_{eq}^*(t) = u_{Tc}(t) - d\left(x_c(t), t\right)$ in the sliding manifold, which means that the unknown disturbance can be suppressed.

Remark 3 ([5, 15]). To avoid the undesired chattering phenomenon, a smooth and continuous saturation function [5] can be used to replace the sign function:

$$\operatorname{sat}\left(s_c(t)\right) = \left[\begin{array}{ccc} \frac{s_{c1}(t)}{|s_{c1}(t)|+\varepsilon_1} & \cdots & \frac{s_{cm}(t)}{|s_{cm}(t)|+\varepsilon_m} \end{array} \right]^T, \tag{37}$$

where ε_i is an arbitrary and small positive constant. If ε_i equals to zero, the saturation function $\operatorname{sat}\left(s_c(t)\right)$ is equivalent to the sign function $\operatorname{sgn}\left(s_c(t)\right)$. When the controlled system contains the direct feed-through term, the undesired chattering phenomenon will affect the controlled system output directly. Therefore, the saturation function should be smooth enough, i.e. the parameter ε_i should be determined properly such that the chattering phenomenon can be minimized. Through the PSO algorithm (will be introduced later in Section 3), the appropriate value of the parameter ε_i can be found.

2.3. **Digital redesign of LQAT integrated with PID control and SMC.** The literature [17] mentions that implementation of the control systems relies more on digital microprocessors and computers. Thus, digital redesign is essential to be applied in our method. Since the bounded nonlinear disturbance $d\left(x_c(t), t\right)$ is unknown, the sliding mode controller in (33) should be considered in order to suppress the disturbance. To

begin with the digital redesign, Euler's approximation is applied, and then (34) can be rewritten as

$$s_d(kT_s+T_s) = (1 - T_s\sigma)\, s_d(kT_s)+T_s \left[d\left(x_d(kT_s), kT_s\right) - \gamma_d \left\|x_d(kT_s)\right\| \operatorname{sgn}\left(s_d(kT_s)\right)\right], \quad (38)$$

where $s_d(kT_s + T_s)$, γ_d and $d\left(x_d(kT_s), kT_s\right)$ denote the discrete-time values of $s_c(t)$, γ_c and $d\left(x_c(t), t\right)$. The parameter σ satisfies $(1 - T_s\sigma) > 0$ in the discrete-time SMC.

Lemma 2 ([11]). In the discrete-time SMC, once the parameter σ satisfies $(1 - T_s\sigma) > 0$, the convergence of the sliding mode function can be proved according to the following conditions:
If $s_d(kT_s) > 0, \Delta s_d(kT_s) = -T_s\sigma s_d(kT_s) + T_s \left[d\left(x_d(kT_s), kT_s\right) - \gamma_d \left\|x_d(kT_s)\right\| \operatorname{sgn}\left(s_d(kT_s)\right)\right] \leq 0$.
If $s_d(kT_s) < 0, \Delta s_d(kT_s) = -T_s\sigma s_d(kT_s) + T_s \left[d\left(x_d(kT_s), kT_s\right) - \gamma_d \left\|x_d(kT_s)\right\| \operatorname{sgn}\left(s_d(kT_s)\right)\right] \geq 0$.
As a result, $s_d(kT_s)$ would move toward to zero whether $s_d(kT_s)$ is a positive or negative value. In other words, the sliding surface $s_d(kT_s) = 0$ is reached. $d\left((x_d, kT_s), kT_s\right)$ is the discrete-time bounded nonlinear disturbance, so the upper bound of $d\left((x_d, kT_s), kT_s\right)$ is the same as that of the continuous-time bounded nonlinear disturbance $d\left(x_c(t), t\right)$. Thus, γ_d should satisfy the condition $\gamma_d \geq d^u \geq 0$, which is equivalent to the condition of γ_c , i.e. $\gamma_d = \gamma_c$.

From Lemma 2, the sliding surface $s_d(k) = 0$ is reached. According to Euler's approximation, let

$$s_I(t) = -\int_0^{t_{end}} (C_s A x_c(t) + u_{Tc}(t))dt, \quad (39)$$

and then the derivative of $s_I(t)$ is

$$\dot{s}_I(t) = -\left(C_s A x_c(t) + u_{Tc}(t)\right). \quad (40)$$

Thus, the sliding mode function can be discretized to

$$s_d(kT_s) = C_s x_d(kT_s) + s_I(kT_s), \quad (41)$$

where $s_I(kT_s) = s_I(kT_s - T_s) + T_s\left(-C_s A x_d(kT_s - T_s) - u_{Tc}(kT_s - T_s)\right)$. Since the sliding surface $s_d(kT_s) = 0$ is reached, the control law can be obtained by $u_c(t) = u_{eq}^*(t)$. Then the unknown disturbance can be suppressed such that the system can be rewritten as

$$\dot{x}_c(t) = A x_c(t) + B u_{Tc}(t), \quad (42)$$

$$y_c(t) = C x_c(t) + D u_{Tc}(t). \quad (43)$$

The control law $u_{Tc}(t)$ can be approximated to

$$u_{Td}(t) = u_{Td}(kT_s) \cong u_{Td}(kT_s + T_s), \quad for \quad kT_s \leq t < kT_s + T_s, \quad (44)$$

where $u_{Td}(kT_s)$ denotes the discrete-time control law.

In order to obtain $u_{Td}(kT_s)$, consider the augmented system state described as

$$\dot{\eta}_c(t) = \bar{A}\eta_c(t) + \bar{B}u_c(t), \quad (45)$$

where $\bar{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}$ and $\bar{B} = \begin{bmatrix} B \\ D \end{bmatrix}$.
The augmented system in (45) can be discretized to

$$\eta_d(kT_s + T_s) = \bar{G}\eta_d(kT_s) + \bar{H}u_{Td}(kT_s), \quad (46)$$

where $\bar{G} = e^{\bar{A}T_s}$ and $\bar{H} = \left(\bar{G} - I_n\right)\bar{A}^{-1}\bar{B}$ denote the discrete-time system matrices, and T_s is the sampling time.

Lemma 3. There are two cases to obtain $u_d^*(kT_s)$ depending on whether the direct

feed-through term exists in the sampled-data system.

Case 1: without direct feed-through term

The discrete-time system state and output can be obtained by

$$x_d(kT_s+T_s) = G_c x_d(kT_s) + HE_{dis} r_d(kT_s+T_s), \tag{47}$$

$$y_d(kT_s+T_s) = Cx_d(kT_s+T_s) = C\left(G_c x_d(kT_s) + HE_{dis} r_d(kT_s+T_s)\right). \tag{48}$$

Take the Z-transform of (47) and (48),

$$\begin{aligned} ZX(z) &= G_c X(z) + HE_{dis} R(z), \\ X(z) &= (ZI - G_c)^{-1} HE_{dis} R(z), \\ Y(z) &= C\left(G_c X(z) + HE_{dis} R(z)\right). \end{aligned} \tag{49}$$

By using an LQAT, the final-value of $y_d(kT_s+T_s)$ should match that of $r_d(kT_s+T_s)$:

$$\lim_{k\to\infty} y_d(kT_s+T_s) = \lim_{k\to\infty} r_d(kT_s+T_s). \tag{50}$$

The Z-transform of (50) is

$$\lim_{z\to 1}(z-1)Y(z) = \lim_{z\to 1}(z-1)R(z). \tag{51}$$

According to (48),

$$\lim_{k\to\infty} y_d(kT_s+T_s) = \lim_{k\to\infty} C\left(G_c x_d(kT_s) + HE_{dis} r_d(kT_s+T_s)\right),$$

which has a Z-transform described as

$$\lim_{z\to 1}(z-1)Y(z) = \lim_{z\to 1}(z-1)C\left(G_c X(z) + HE_{dis} R(z)\right). \tag{52}$$

Then substitute (49) into (52),

$$\begin{aligned} \lim_{z\to 1}(z-1)Y(z) &= \lim_{z\to 1}(z-1)C\left(G_c(ZI-G_c)^{-1}HE_{dis}R(z) + HE_{dis}R(z)\right) \\ &= C\left(G_c(I-G_c)^{-1} + I\right)HE_{dis} \cdot \lim_{z\to 1}(z-1)R(z) \\ &= C(I-G_c)^{-1}HE_{dis} \cdot \lim_{z\to 1}(z-1)R(z). \end{aligned}$$

To satisfy (50), $C(I-G_c)^{-1}HE_{dis} = I$ is needed. Thus, $\lim_{k\to\infty} y_d(kT_s+T_s) = \lim_{k\to\infty} r_d(kT_s+T_s)$.

Case 2: with direct feed-through term

The discrete-time system state and output can be obtained by

$$x_d(kT_s+T_s) = G_c x_d(kT_s) + HE_{dis} r_d(kT_s+T_s), \tag{53}$$

$$y_d(kT_s) = C_c x_d(kT_s) + DE_{dis} r_d(kT_s+T_s), \tag{54}$$

Take the Z-transform of (53) and (54),

$$\begin{aligned} ZX(z) &= G_c X(z) + HE_{dis} R(z), \\ X(z) &= (ZI - G_c)^{-1} HE_{dis} R(z), \\ Y(z) &= C_c X(z) + DE_{dis} R(z). \end{aligned} \tag{55}$$

By using an LQAT, the final-value of $y_d(kT_s)$ should match that of $r_d(kT_s)$:

$$\lim_{k\to\infty} y_d(kT_s) = \lim_{k\to\infty} r_d(kT_s). \tag{56}$$

The Z-transform of (56) is

$$\lim_{z\to 1}(z-1)Y(z)=\lim_{z\to 1}(z-1)R(z). \tag{57}$$

According to (54),

$$\lim_{k\to\infty} y_d(kT_s)=\lim_{k\to\infty} C_c x_d(kT_s)+DE_{dis}r_d(kT_s+T_s).$$

which has a Z-transform described as

$$\lim_{z\to 1}(z-1)Y(z)=\lim_{z\to 1}(z-1)\left(C_cX(z)+DE_{dis}R(z)\right). \tag{58}$$

Then substitute (55) into (58),

$$\begin{aligned}\lim_{z\to 1}(z-1)Y(z)&=\lim_{z\to 1}(z-1)\left(C_c(ZI-G_c)^{-1}HE_{dis}R(z)+DE_{dis}R(z)\right)\\&=\lim_{z\to 1}(z-1)\left(C_c(ZI-G_c)^{-1}H+D\right)E_{dis}R(z)\\&=\left(C_c(I-G_c)^{-1}H+D\right)\cdot\lim_{z\to 1}(z-1)E_{dis}R(z).\end{aligned}$$

To satisfy (56), $\left(C_c(I-G_c)^{-1}H+D\right)=I$ is needed. Thus, $\lim_{k\to\infty} y_d(kT_s)=\lim_{k\to\infty} r_d(kT_s)$.

From Case 1, the control law in a sampled-data system which contains no direct feed-through term is $u_d^*(kT_s)=-K_cx_d(kT_s+T_s)+E_cr_d(kT_s+T_s)$; from Case 2, the control law in a sampled-data system which contains direct feed-through term is $u_d^*(kT_s)=-K_cx_d(kT_s+T_s)+E_cr_d(kT_s)$.

According to Lemma 3, (29) can be approximated to

$$\begin{aligned}u_{Td}(kT_s)&=-K_{PI}\eta_d(kT_s+T_s)+K_D\tfrac{\eta_d(kT_s+T_s)-\eta_d(kT_s)}{T_s}+E_cr_d(kT_s)\\&=-K_{PI}\left\{\bar{G}\eta_d(kT_s)+\bar{H}u_{Td}(kT_s)\right\}+\tfrac{K_D}{T_s}\left\{\bar{G}\eta_d(kT_s)+\bar{H}u_{Td}(kT_s)\right\}-\tfrac{K_D}{T_s}\eta_d(kT_s)+E_cr_d(kT_s)\\&=-\left(K_{PI}\bar{H}+\tfrac{K_D}{T_s}\bar{H}\right)u_{Td}(kT_s)-\left(K_{PI}\bar{G}-\tfrac{K_D}{T_s}\bar{G}+\tfrac{K_D}{T_s}\right)\eta_d(kT_s)+E_cr_d(kT_s),\end{aligned} \tag{59}$$

where $K_D=\begin{bmatrix}K_d & 0\end{bmatrix}$. After rearranging,

$$u_{Td}(kT_s)=\left[I-\left(K_{PI}\bar{H}+\tfrac{K_D}{T_s}\bar{H}\right)\right]^{-1}\left[-\left(K_{PI}\bar{G}-\tfrac{K_D}{T_s}\bar{G}+\tfrac{K_D}{T_s}\right)\eta_d(kT_s)+E_cr_d(kT_s)\right]. \tag{60}$$

Therefore, the controller gain K_{dis} and the forward gain E_{dis} in the discrete-time system are denoted to

$$K_{dis}=\left[I-\left(K_{PI}\bar{H}+\frac{K_D}{T_s}\bar{H}\right)\right]^{-1}\left(K_{PI}\bar{G}-\frac{K_D}{T_s}\bar{G}+\frac{K_D}{T_s}\right), \tag{61}$$

$$E_{dis}=\left[I-\left(K_{PI}\bar{H}+\frac{K_D}{T_s}\bar{H}\right)\right]^{-1}E_c. \tag{62}$$

It is essential to have a smooth control law so that the chattering phenomenon can be avoided. In our approach, there are two factors which influence the control law to be smooth or not. Firstly, the switching control law contains the sign function such that the chattering phenomenon might occur, which is not desirable for our design. Secondly, the PID controller is sensitive to the derivative term of the system. These factors can be solved in the following discussion. In (33), the control law contains a sign function which is not a smooth function. Therefore, a saturation function is introduced in (37). So, (33) can be replaced by

$$u_c(t)=u_{Tc}(t)-\gamma_c\left\|x_c(t)\right\|\mathrm{sat}(s_c(t))-\sigma s_c(t). \tag{63}$$

From the above discussion, the digital redesign control law can be obtained by

$$u_d(kT_s) = u_{Td}(kT_s) - \gamma_d \|x_d(kT_s)\| \operatorname{sat}(s_d(kT_s)) - \sigma s_d(kT_s). \tag{64}$$

Note that the saturation function in the switching control law contains the parameter ε_i , which needs to be selected properly. If ε_i is selected improperly, the chattering phenomenon might still occur, which is not desirable for our design. For the second factor mentioned above, digital redesign can reduce the sensitivity of the system variation caused by the derivative term. In (60), $u_{Td}(kT_s)$ contains no derivative term after digital redesign, and therefore it is a smooth control law. According to the above discussion, the control force would be smooth and make the system track the trajectory.

3. **Digital-Redesigned Robust Tracker for Sampled-Data Systems with Nonlinear Disturbances Based on PID Control, SMC and PSO Algorithm.** In section 2, there are some specific parameters which need to be adjusted properly. To find out the near-optimal solution of these parameters, the multi-swarm cooperative particle swarm optimizer (MCPSO) algorithm is applied in our approach. A brief introduction of MCPSO will be shown in this section.

3.1. **A brief introduction of a standard PSO (SPSO).** A typical PSO algorithm originates from a flock of birds flying through a D-dimensional space. A bird is called a particle in the PSO algorithm. The idea is that the bird would adjust its route while searching space according to both its own experience and its neighbors' experience. $z_i = (z_{i1}, z_{i2}, ..., z_{iD})$ denotes the ith particle in the D-dimensional space, where $z_{id} \in [l_d, u_d]$, $d \in [1, D]$, and l_d, u_d are the lower bound and the upper bound of the dth dimension, respectively. The velocity of the ith particle can be denoted as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The SPSO algorithm can be described as

$$v_i(t+1) = v_i(t) + r_1 c_1 (p_i - z_i(t)) + r_2 c_2 (p_g - z_i(t)), \tag{65}$$

$$z_i(t+1) = z_i(t) + v_i(t), \tag{66}$$

where r_1 and r_2 are random values between 0 and 1, c_1 and c_2 are the acceleration constants, p_1 is the best previous position of the ith particle, and p_g is the global best position among all the particles in the swarm.

3.2. **Modified PSO algorithms.**

3.2.1. *Linearly decreasing inertia weight method of PSO (LPSO).* The literature [18] modifies (65) by adding an inertia term w :

$$v_i(t+1) = w \times v_i(t) + r_1 c_1 (p_i - z_i(t)) + r_2 c_2 (p_g - z_i(t)), \tag{67}$$

$$z_i(t+1) = z_i(t) + v_i(t). \tag{68}$$

The inertia term can balance the global and local explorations if selected properly. Usually, w decreases linearly from 0.9 to 0.4 during a whole process. The inertia term w is commonly chosen according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times \text{iter}, \tag{69}$$

where w_{max} is the initial weight, and w_{min} is the final weight. $iter_{max}$ is the maximum time of iterations and is the current number of iterations. In [19], this version of SPSO is referred to as LPSO.

3.2.2. *Random weight method of PSO (RPSO).* Different from LPSO, [21] proposes a random inertia weight factor to modify the velocity of the particles. In the method, the inertia weight factor would change randomly as the following equation:

$$w = 0.5 - \frac{\text{rand}(\bullet)}{2}, \tag{70}$$

where rand($\bullet$) is a uniformly distributed random number between 0 and 1. In [19], this method is referred to as RPSO.

3.2.3. *Constriction factor approach PSO (CPSO).* The third modified PSO algorithm is the constriction factor approach PSO (CPSO), which is proposed by [22]. This modified SPSO can be described by the following equations

$$v_i(t+1) = \kappa\left(v_i(t) + r_1c_1\left(p_i - z_i(t)\right) + r_2c_2\left(p_g - z_i(t)\right)\right), \tag{71}$$

$$z_i(t+1) = z_i(t) + v_i(t), \tag{72}$$

$$\kappa = \frac{2}{\left|2 - \varphi + \sqrt{\varphi^2 - 4\varphi}\right|}, \quad where \quad \varphi = c_1 + c_2, \varphi > 4. \tag{73}$$

Note that the constriction factor κ is a function of c_1 and c_2 .

3.3. **COM-MCPSO algorithm.** After the brief reviews about LPSO, RPSO and CPSO, COM-MCPSO can be introduced in the following section. [19] presents a new optimization algorithm: MCPSO. Inspired by the phenomenon of symbiosis in natural ecosystems, they found that a master-slave model is advisable to be incorporated into the SPSO, so the MCPSO is developed. In this method, one master swarm and some slave swarms are included in a population. The symbiotic relationship between the master swarm and slave swarms can help balance the exploration and exploitation, which is a critical key to the success of an optimization task.

In MCPSO algorithm, each slave indicates a kind of PSO, for example, we use three slaves, LPSO, RPSO, and CPSO, in our approach. The master swarm can choose the best of all received individuals after the slave swarms share their best positions with the master swarm. Then, the master swarm can evolve according to the following equations:

$$v_i^M(t+1) = w \times v_i^M(t) + r_1c_1\left(p_i^M - z_i^M(t)\right) + \Phi r_2c_2\left(p_g^M - z_i^M(t)\right) + (1-\Phi)\, r_3c_3\left(p_g^S - z_i^M(t)\right), \tag{74}$$

$$z_i^M(t+1) = z_i^M(t) + v_i^M(t), \tag{75}$$

where M and S denote the master swarm and slave swarm, respectively. r_3 is a random number between 0 and 1, and c_3 is the acceleration constant. According to [19], is a migration factor used for migration problem, which is given by

$$\Phi = \begin{cases} 0 & G_{best}^S < G_{best}^M, \\ 0.5 & G_{best}^S = G_{best}^M, \\ 1 & G_{best}^S > G_{best}^M, \end{cases} \tag{76}$$

where p_g^M and p_g^S are the best previous particles in master swarm and in slave swarms, respectively. G_{best}^M denotes the cost value determined by p_g^M ;G_{best}^S denotes the cost value determined by p_g^S .

The MCPSO algorithm we have discussed above is called as competitive version MCPSO, or COM-MCPSO. In the method, the master swarm improves its particles according to the direct competition with the slave swarms. Thus, the chance of finding the most fitted particle in all the swarms (master swarm or slave swarms) is raised such that the master swarm can be guided to the best direction among all the swarms.

3.4. **Digital redesign robust tracker for sampled-data systems with nonlinear disturbances based on PID control, SMC and PSO algorithm.** Now, we integrate the MCPSO algorithm with the PID control, SMC and the digital redesign robust tracker for sampled-data systems with bounded nonlinear disturbances, such that some design parameters of the controller can be optimally tuned. Our goal is to minimize the output error, chattering phenomenon and improve the transient response. As a result, a cost function used to evaluate the performance is designed as

$$J_{\text{cost}} = \int_0^{t_{end}} \sum_{i=1}^{p} |e_{y,i}(t)| + \sum_{i=1}^{p} |\Delta e_{y,i}(t)| dt, \tag{77}$$

where $e_y(t) = y_c(t) - r_c(t)$, $\Delta e_y(t) = e_y(t) - e_y(t-1)$.

In our approach, there are three parameters can be determined by applying MCPSO algorithm: α for the D-type controller design in (8),ε for the SMC design in (37) and Q_k for the tracker design in (16). After the parameters are decided by MCPSO algorithm, the system output error and chattering phenomenon can be minimized, and the transient response can be improved too. The detailed experiment process will be discussed in the following section.

3.5. **Experimental setting.** There are several parameters which should be determined by the users. The report [19] mentions that the parameters used for LPSO, RPSO, CPSO are recommended in [24–26], which are shown in the following descriptions:
1. LPSO: $c_1 = c_2 = 2.0$, $w_{min} = 0.4$ and $w_{max} = 0.9$
2. CPSO: $c_1 = c_2 = 2.0$ and $\kappa = 0.729$
3. RPSO: $c_1 = c_2 = 1.494$
4. MCPSO: $c_1 = c_2 = 2.05$, $c_3 = 2.0$, $w_{min} = 0.4$ and $w_{max} = 0.9$

The selection of the inertia weight is important for conducting a PSO algorithm. If the inertia weight is selected properly, it can provide a balance between the local search and the global search. Hence, the optimal or near optimal solution can be reached more accurately and efficiently. In [19], the inertia weight for LPSO and MCPSO are set to decaying numbers started at 0.9 and ended at 0.4 (see [20]); for CPSO, the constriction factor κ was set at 0.729.

Refer to [19], the number of slave swarms is set as 3. The three slave swarms, LPSO, RPSO, and CPSO, are the improved versions of the SPSO. In our experiments, the parameters of these slave swarms are set as those mentioned above. The population size of each swarm was set to 50, and the maximum number of iterations was set to 100 generations.

Our target is to find out the proper values of α in D-type controller design, ε in SMC saturation function and in the tracker design. In order to enhance the opportunity of finding the near-optimal value, ε can be presented as $\varepsilon = 10^{\delta}$. The upper bounds and the lower bounds are set as $\alpha = [0.1, 1]$, $\delta = [-5, -1]$ and $Q_k = [10^3, 10^7]$.

4. **An Illustrative example.** Consider the controllable time-invariant system with bounded nonlinear disturbance described by

$$\dot{x}_c(t) = Ax_c(t) + B\left[u_c(t) + d\left(x_c(t), t\right)\right],$$
$$y_c(t) = Cx_c(t) + D\left[u_c(t) + d\left(x_c(t), t\right)\right],$$

where $A = \begin{bmatrix} -35 & 35 & 0 \\ -7 & 28 & 0 \\ 0 & 0 & -3 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, $C = \begin{bmatrix} -0.5 & 5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$, $D = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}$,and the initial condition is selected as $x_c(0) = \begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix}^T$. The

bounded nonlinear disturbance and reference $r_c(t)$ are respectively given by

$$d\left(x_c(t),t\right)=\sin\left(x_{c1}(t)x_{c2}(t)\right)\begin{bmatrix} x_{c1}(t) \\ -x_{c1}(t)\end{bmatrix} \text{ and } r_c(t)=\begin{bmatrix} r_{c1}(t) \\ r_{c2}(t)\end{bmatrix}=\begin{bmatrix} \sin(5t) \\ -\cos(5t)\end{bmatrix}.$$

After the near-optimal solution search of the MCPSO algorithm, the matrix pair $\{Q_k,R_k\}=\left\{10^7 I_2, I_2\right\}$ are selected for the controller design. The parameter α is set as $\alpha=0.9922$, which yields $K_d=\begin{bmatrix} 0 & 0.0078 & 0 \\ 0 & 0 & 0.0078\end{bmatrix}$ and $M=I_3-BK_d=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9922 & 0 \\ 0 & 0 & 0.9922\end{bmatrix}$.And the parameter in saturation function is selected as $\delta=-1$, which means $\varepsilon=10^{-1}=0.1$.

The PI-type controller gain matrix can be obtained by

$$K_{PI}=\begin{bmatrix} K_p & K_i\end{bmatrix}$$

$$=\begin{bmatrix} -5.0157 & 49.8276 & 0 & 1.9843 & 0 \\ 0 & 0 & 2.4569 & 0 & 0.9922\end{bmatrix},$$

where $K_p\in\Re^{2\times 3}$ and $K_p\in\Re^{2\times 2}$. The forward gain in continuous time is $E_c=\begin{bmatrix} 9.9216 & 0 \\ 0 & 4.9608\end{bmatrix}$. The controller gain and the forward gain in discrete time are $K_{dis}=\begin{bmatrix} -4.9262 & 48.7676 & 0 & 1.9031 & 0 \\ 0 & 0 & 2.4864 & 0 & 0.9973\end{bmatrix}$, $E_{dis}=\begin{bmatrix} 9.5153 & 0 \\ 0 & 4.9866\end{bmatrix}$ for the selected sampling time $T_s=10^{-3}$sec. The sliding manifold and the control law are given in (41) and (60), respectively, where

$$\bar{A}_{PID}=\begin{bmatrix} -35 & 35 & 0 & 0 & 0 \\ -7.0553 & 28.2212 & 0 & 0 & 0 \\ 0 & 0 & -3.0237 & 0 & 0 \\ -0.5055 & 5.0221 & 0 & 0 & 0 \\ 0 & 0 & 0.4953 & 0 & 0\end{bmatrix},\bar{B}_{PID}=\begin{bmatrix} 0 & 0 \\ 1.0079 & 0 \\ 0 & 1.0079 \\ 0.1008 & 0 \\ 0 & 0.2016\end{bmatrix},$$

$$\bar{C}_{PID}=\begin{bmatrix} -0.5055 & 5.0221 & 0 & 0 & 0 \\ 0 & 0 & 0.4953 & 0 & 0\end{bmatrix},\bar{D}_{PID}=\begin{bmatrix} 0.1008 & 0 \\ 0 & 0.2016\end{bmatrix},$$

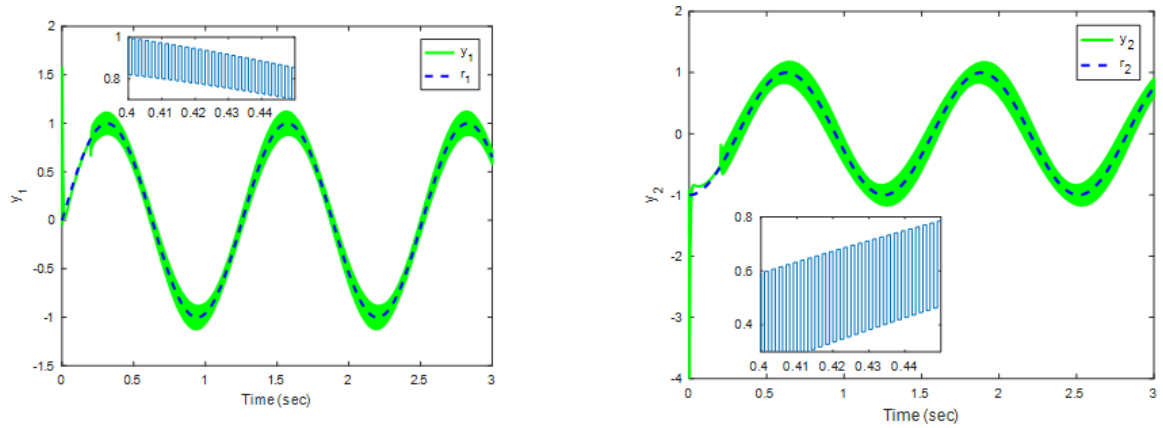
$$C_s=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1\end{bmatrix}, \text{ and } \gamma_c \text{ is given as } \sqrt{2}.$$

In order to analysis the difference between applying PSO algorithm and without applying PSO algorithm, the following experiments are set as the same initial values, where $\alpha=0.5$, $\delta=-5$ and $Q_k=10^3$.

Figs. 1-4 represent the performance without applying PSO algorithm; Figs. 5-9 represents the improved performance with applying PSO algorithm. Fig. 1 and Fig. 5 show the tracking performance between the controlled system outputs $y_d(kT_s)$ and the reference trajectory $r_d(kT_s)$. Fig. 2 and Fig. 6 display the system states of the controlled system. Fig. 3 and Fig. 7 demonstrate the control laws $u_d(kT_s)$ of the controlled system. Fig. 4 and Fig. 8 illustrate the sliding surface. Fig. 9 demonstrates the decaying cost values in MCPSO algorithm.

Note that Figs. 1-4 can be compared with the Figs. 5-8. Figs.5-8 indicate a satisfied performance base on the proposed tracker for the system with unknown disturbances. By comparing Fig. 1 with Fig. 5, the transient response improves significantly after applying

PSO algorithm. Compared with Fig. 3 and Fig. 7, the negative effect of chattering phenomenon is suppressed apparently.



(a) y_1 (b) y_2

FIGURE 1. The system outputs (a) y_1 (b) y_2 of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (without MCPSO algorithm).

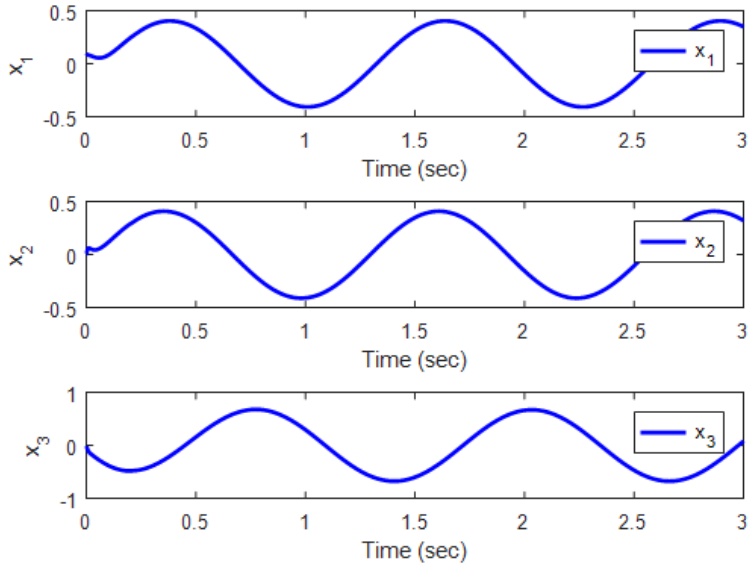
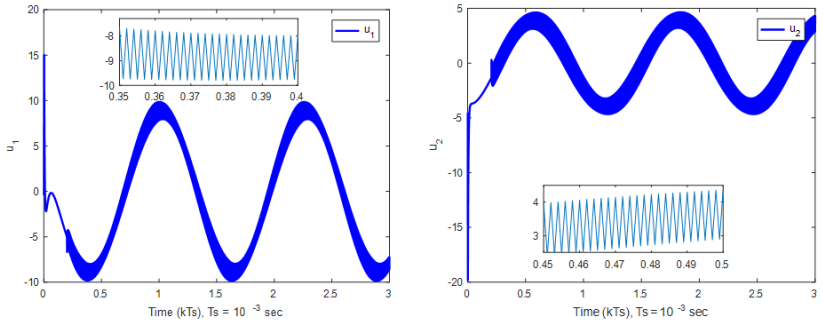


FIGURE 2. The system states of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (without MCPSO algorithm).



(a) u_1 (b) u_2

FIGURE 3. The control law (a) u_1 (b) u_2 of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (without MCPSO algorithm).

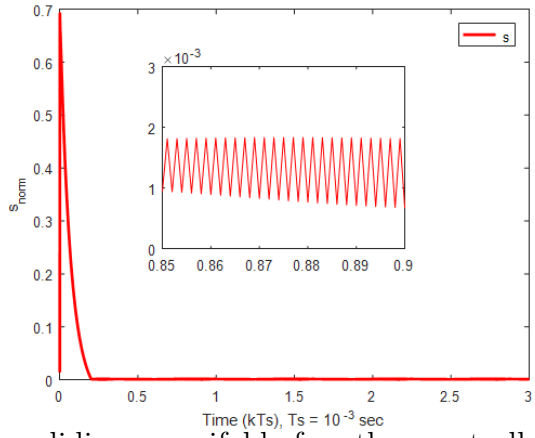
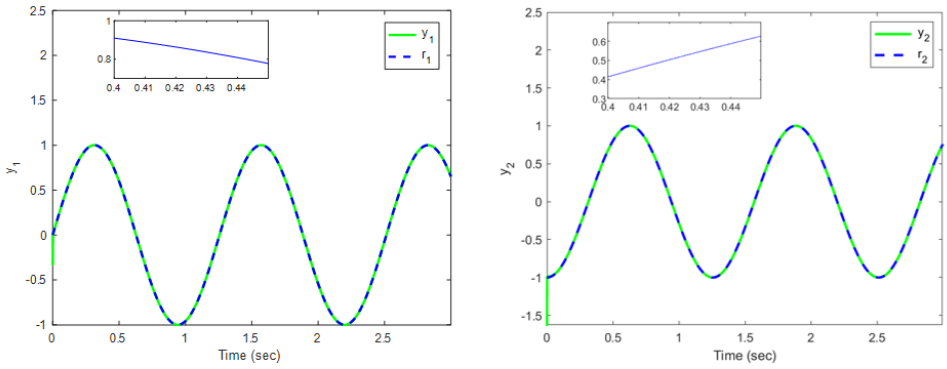


FIGURE 4. The sliding manifold for the controllable time-invariant sampled-data system with bounded nonlinear disturbance (without MCPSO algorithm).



(a) y_1 (b) y_2

FIGURE 5. The system outputs (a) y_1 (b) y_2 of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (with MCPSO algorithm).

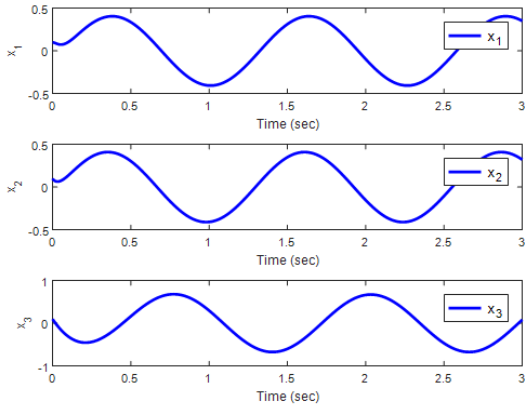
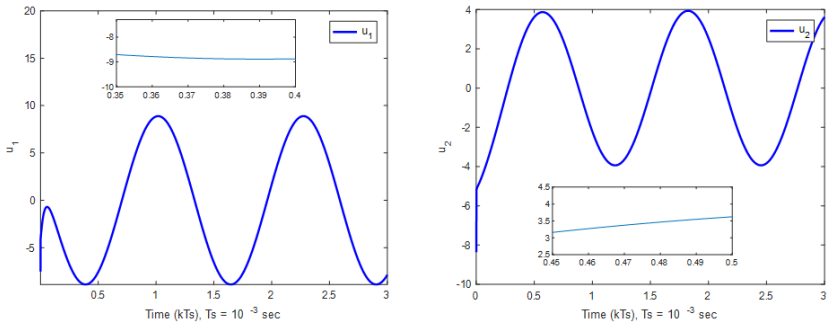


FIGURE 6. The system states of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (with MCPSO algorithm).



(a) u_1 (b) u_2

FIGURE 7. The control law (a) u_1 (b) u_2 of the controllable time-invariant sampled-data system with bounded nonlinear disturbance (with MCPSO algorithm).

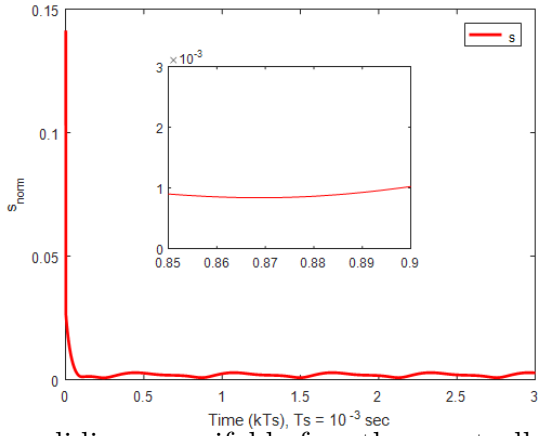


FIGURE 8. The sliding manifold for the controllable time-invariant sampled-data system with bounded nonlinear disturbance (with MCPSO algorithm).

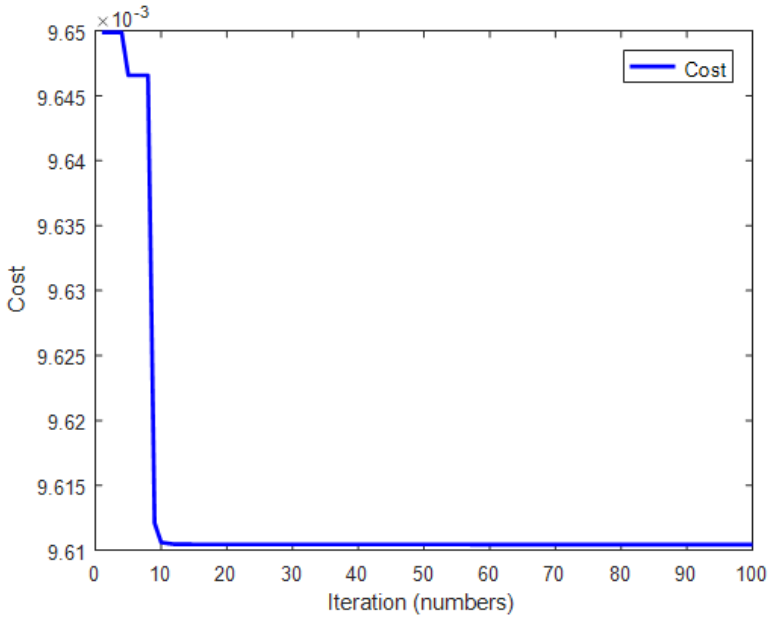


FIGURE 9. Cost values described in (65) during the total 100 iterations.

5. **Conclusion.** Given the sampled-data systems with bounded nonlinear disturbances, this paper mainly focuses on the new method for the digital-redesign robust tracker based on an integration of the PID control, SMC and PSO algorithm in order to discretize the original continuous-controller for the sampled-data systems. The discussion about discrete-time sliding mode controller is more complicated than that about continuous-time sliding mode controller. Thus, this paper proposes a new digital redesign approach integrated with PID control, SMC and PSO algorithm to transfer the continuous-time controller to the discrete-time controller and preserve the performance of the original continuous-time controller. The application of SMC can suppress the negative effect induced by the bounded nonlinear disturbances such that the system resembles a linear system; consequently, the digital redesign method can be applied. Then, the MCPSO algorithm is applied to find the optimal solution or near-optimal solution for the integrated LQAT, PID controller and sliding mode controller design. After the MCPSO optimization, the chattering phenomenon in SMC is reduced and the transient response is improved too. The proposed new method in the paper is verified to be effective from the simulation results.

Author Contributions: All authors contributed to the paper. J.S.F. and C.H.H. proposed the research idea of the sampled data system. C.H.H conducted the theoretical derivation and wrote the manuscript with the supervision from J.S.H.T. and J.J.Y.. S.M.G. is responsible for the simulation design.

Funding: This work was financially supported by the Ministry of Science and Technology, Taiwan, under grant MOST- 110-2221-E-167 -030 and MOST-110-2218-E-006 -014 -MBK.

Conflict of Interest: The authors declare no conflict of interest.

REFERENCES

[1]] Z.L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Transactions on Energy Conversion*, vol. 19, no.2, pp. 384-391, 2004.

[2] Y.Y. Nazaruddin, A.D. Andrini and B. Anditio, PSO based PID controller for quadrotor with virtual sensor, *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 358-363, 2018.
[3] D. Rosinová, V. Veselý, Robust PID decentralized controller design using LMI, International Journal of Computers, Communications & Control, vol. II, no. 2, pp. 195-204, 2007.
[4] T. Abdelaziz, M. Valasek, Pole-placement for SISO linear systems by state-derivative feedback, *IEEE Proceedings Control Theory and Applications*, vol.151, no. 4, pp. 377-385, 2004.
[5] J.S. Lin, C.F. Huang, T.L. Liao and J.J. Yan, Design and implementation of digital secure communication based on synchronized chaotic systems, *Digital Signal Processing*, vol.20, no. 1, pp. 229-237, 2010.
[6] M. Singla, L.S. Shieh, G. Song and L. Xie, A new optimal sliding mode controller design using scalar sign function, *ISA transactions*, vol.53, no. 2, pp. 267-279, 2014.
[7] N. Wang, Q. Deng, G. Xie and X. Pan, Hybrid finite-time trajectory tracking control of a quadrotor, *ISA Transactions*, vol. 90, pp. 278-286, 2019.
[8] W.D. Chang, J.J. Yan, Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems, Chaos, *Solitons & Fractals*, vol.26, no. 1, pp. 167-175, 2005.
[9] C. Huang , F. Naghdy and H. Du, Sliding mode predictive tracking control for uncertain Steer-by-Wire system, *Control Engineering Practice*, vol.85, pp.194-205, 2019.
[10] J.L. Chang, Combining state estimate and disturbance observer in discrete-time sliding mode controller design, *Asian Journal of Control*, vol.10, no.5, pp. 515-524, 2008.
[11] M. Yan, Y. Shi, Robust discrete-time sliding mode control for uncertain systems with time-varying state delay, *IET Control Theory Applications*, vol.2, no.8, pp. 662-674, 2008.
[12] T.H. Yan, B. He, X.D. Chen and X.S. Xul, The discrete-time sliding mode control with computation time delay for repeatable run-out compensation of hard disk drives, *Mathematical Problems in Engineering*, vol. 2013, 505846, 2013.
[13] G. Sun, L. Wu, Z. Kuang, Z. Ma and J. Liu, Practical tracking control of linear motor via fractional-order sliding mode, *Automatica*, vol. 94, pp. 221-235, 2018.
[14] S.M. Guo, L.S. Shieh, G. Chen and C.F. Lin, Effective chaotic orbit tracker: a prediction-based digital redesign approach, *IEEE Transactions on Circuits and Systems—I. Fundamental Theory and Applications*, vol. 47, no. 11, pp. 1557-1569, 2000.
[15] J.S.H. Tsai, J.S. Fang, J.J. Yan, M.C. Dai, S.M. Guo and L.S. Shieh, Hybrid robust discrete sliding mode control for generalized continuous chaotic systems subject to external disturbances, *Nonlinear Analysis: Hybrid Systems*, vol.29, pp.74-84, 2018.
[16] A. Abera, B. Bandyopadhyay, Digital redesign of sliding mode control with application to power system stabilizer, *34th Annual Conference of IEEE Industrial Electronics*, pp.164-169, 2008.
[17] X. Li, C, Wang, Q. Wen, X. Yu and B. Wang, Digital redesign of sliding mode control of LTI systems, *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pp. 2431-2435, 2006.
[18] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.
[19] B. Niu, Y. Zhu, X. He and H. Wu, MCPSO: A multi-swarm cooperative particle swarm optimizer, *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1050-1062, 2007.
[20] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73, 1998.
[21] R.C. Eberhart, Y. Shi, Tracking and optimizing dynamic systems with particle swarms, *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 94-97, 2001.
[22] M. Clerc, J. Kennedy, The particle swarm: explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.
[23] S.L. Shieh, M.H. Dib and G. Sekar, Continuous-time quadratic regulators and pseudo-continuous-time quadratic regulators with pole placement in a specific region, *IEEE Proceedings D -Control Theory and Applications*, vol. 134, no.5, pp. 338-346, 1987.
[24] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, *Proceedings of Congress on Evolutionary Computation*, pp. 1945-1949,1999.
[25] R. Mendes, P. Cortez, M. Rocha and J. Neves, Particle swarms for feedforward neural network training, *Proceedings of the International Joint Conference on Neural Networks*, pp.1895-1899, 2002.
[26] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu and L.M. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, *Information Processing Letters*, vol.93, no.5, pp. 255-261, 2005.