Applying the Simplex Method to Design the Optimal Parameters of a Sliding Mode Controller for a System With Input Saturation and Unknown Disturbance Bound

CHYUN-CHAU FUH[ID]1 AND HSUN-HENG TSAI[ID]2

[1]Department of Mechanical and Mechatronic Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan
[2]Department of Biomechatronics Engineering, National Pingtung University of Science and Technology, Pingtung 91201, Taiwan

Corresponding author: Hsun-Heng Tsai (heng@mail.npust.edu.tw)

ABSTRACT Sliding mode controllers have wide practical applications due to their simple structure and robustness to disturbance. Although various sliding mode controllers have been proposed, most relevant studies have only investigated the conditions required to stabilize the system. For example, system stability can be guaranteed if the parameters in the control law satisfy specific inequalities or boundary conditions. However, the control power of a real system has a saturation limit, which increases the complexity of the system. Moreover, even if a stable controller is identified, the response performance of the system may not achieve the expected results. Therefore, the determination of the optimal parameters is a key step in designing a sliding mode controller. This paper proposes an error-integral-type sliding mode controller whose main characteristic is that irrespective of the system order, it only needs to determine three key parameters. We used the Nelder–Mead simplex method to assist designers in searching for optimal controller parameters more efficiently, even when the real system has the limitation of input saturation and the disturbance bound is unknown. Finally, we performed numerical simulations on a magnetic levitation system to demonstrate the feasibility and effectiveness of the proposed method.

INDEX TERMS Integral sliding mode controller, magnetic levitation system, optimal control, optimization method, saturation, simplex method.

I. INTRODUCTION

The basic principles and design methods of a variable-structure system (VSS) with sliding mode control (SMC) were first proposed by Emelyanov and several core researchers in the Soviet Union in the early 1950s [1]–[4]. Since the publication of the survey paper on VSS in *IEEE Transactions on Automatic Control i*n 1977 [5], SMC has been extensively studied and used in practical applications due to its simplicity and robustness with respect to external disturbances and modeling uncertainties [6]–[8]. In particular, sliding mode controllers are a special class of VSSs, and SMC is a type of nonlinear control method that alters the dynamics of a closed-loop system by using a high-frequency switching control law that forces the state trajectory to

"slide" along a cross section (manifold) of the system's normal behavior [6]–[9].

SMC design can be divided into two steps [10]: (1) constructing a switching surface (sliding surface, sliding function, and sliding vector) so that the system restricted to the switching surface produces a desired behavior and (2) determining a control law that can drive the state trajectory toward the sliding surface and maintain it there. The original SMC design contains an ideal switching function, namely sign(s), which must be achieved at an infinite switching frequency. However, this switching frequency cannot be achieved in a practical system. As a result, the real system trajectory invariably fluctuates in a small space on both sides of the sliding surface. This phenomenon is known as "chattering." The chattering phenomenon is usually undesirable, or even unallowable, because it may increase the actuator load or excite the high-frequency resonance of the system. Many

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo[ID].

methods have been proposed to improve the chattering phenomenon [11]–[16]. Of these, the simplest method is the boundary layer method proposed by Slotine and Sastry, which has been widely applied in real SMC systems [11]. This method replaces the discontinuous function, namely sign(s), with a saturation function.

For any practical system, the input is always limited in magnitude. Therefore, this limitation must be considered while designing a sliding mode controller. Some researchers have contributed in this regard. For example, Corradini and Orlando proposed the use of a nonlinear surface for manipulating actuator saturation [17]. Ferrara and Rubagotti proposed an effective algorithm to handle saturation in a higher-order sliding mode framework [18]. Bartoszewicz and Nowacka proposed an optimal sliding surface to handle input constraints [19]. However, the aforementioned methods are conservative or limited in terms of the system order.

Therefore, this paper proposes an approach for designing an optimal sliding mode controller for a system with input saturation and an unknown disturbance bound. This method uses the Nelder–Mead (N–M) algorithm to tune the parameters of the sliding mode controller such that the designer's pre-specified performance index reaches a minimum value. The simplex method (also known as the downhill simplex method) was originally proposed by Spendley, Hext, and Himsworth [20] and later improved by Nelder and Mead [21]. Hereafter, we refer to the improved method as the N–M simplex method. This numerical method is easy to implement and has been widely used to determine the minimum or maximum value of the objective function in a multidimensional parameter space. In particular, the N–M simplex method does not require the derivative of the objective function; thus, it is highly suitable for discontinuities or problems in which the objective functions include noise [22]–[26]. In addition, the proposed method offers the following advantages: 1) The algorithms used in linear and nonlinear searches for optimal controller parameters are the same. 2) This method can be used regardless of the stability of the open-loop system. 3) For systems with uncertainties, this methodology can be used to determine the optimal controller parameters.

The main contribution of this study is that the techniques proposed herein can be used to simultaneously solve the aforementioned problems. To demonstrate the feasibility and effectiveness of the proposed methodology, simulations of the magnetic levitation control system with input saturation were performed to evaluate controller performance.

II. SYSTEM DESCRIPTION
Fully complying with linear time-invariant conditions in practical systems is rarely possible because most practical systems are nonlinear. However, theories based on linear time-invariant systems are mature and highly convenient to use. Therefore, before designing a controller, the system is usually linearized into an approximate linear time-invariant mathematical model.

Let us consider an nth-order controllable system that is described as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d}(\mathbf{x}, t) \tag{1a}$$

$$\mathbf{y} = \mathbf{G}\mathbf{x} \tag{1b}$$

where $\mathbf{x} = [x_1 \ x_2 \ldots x_n]^T$, $\mathbf{u} = [u_1 \ u_2 \ldots u_{m_u}]^T$, and $\mathbf{y} = [y_1 \ y_2 \ldots y_{m_y}]^T$ denote the system state, control input, and system output, respectively. Moreover, all elements of the coefficient matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}$ are constant and known in advance, and $(\mathbf{A}, \mathbf{B})$ is controllable. The disturbance is represented by $\mathbf{d}(\mathbf{x}, t)$ and includes external disturbance, parameter uncertainty, and unmodeled dynamics. Let us assume that $\mathbf{d}(\mathbf{x}, t)$ can be decomposed into matched and unmatched disturbances as follows:

$$\mathbf{d}(\mathbf{x}, t) = \mathbf{B}\mathbf{d}_m(\mathbf{x}, t) + \mathbf{B}_r \mathbf{d}_r \tag{2}$$

where $\mathbf{B}_r \in \mathbf{R}^{n \times (n-m_u)}$, $\mathbf{B}^T \mathbf{B}_r = \mathbf{0}$, and $\mathbf{d}_r$ is a constant vector-type unmatched disturbance. The system state $\mathbf{x}$ and output $\mathbf{y}$ are measurable. The control aims to converge the output $\mathbf{y}$ to a constant vector $\mathbf{y}_d$. For design convenience, let the error vector be defined as follows:

$$\mathbf{e} = \mathbf{y} - \mathbf{y}_d \tag{3}$$

Then, we define a new state vector as follows:

$$\mathbf{z} = \mathbf{W} \int_0^t \mathbf{e} d\tau, \quad \mathbf{z}(0) = \mathbf{0}_{m_y \times 1} \tag{4}$$

where $\mathbf{W}$ indicates a weight matrix and has the following form:

$$\mathbf{W} = diag(w_1, w_2, \ldots, w_{m_y}) \tag{5}$$

Thus,

$$\dot{\mathbf{z}} = \mathbf{W}\mathbf{e} = \mathbf{W}(\mathbf{y} - \mathbf{y}_d) = \mathbf{W}\mathbf{G}\mathbf{x} - \mathbf{W}\mathbf{y}_d \tag{6}$$

We multiply an appropriate weight matrix $\mathbf{W}$ with the error integrator (4) to ensure that the integral controller can still reflect its effectiveness when the absolute value of the error is too small.

Due to the introduction of new state variables, the system (1) can be expanded as follows:

$$\dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p (\mathbf{u} + \mathbf{d}_m(\mathbf{x}, t)) + \mathbf{f} \tag{7}$$

where

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \mathbf{A}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{W}\mathbf{G} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_p = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{B}_r \mathbf{d}_r \\ -\mathbf{W}\mathbf{y}_d \end{bmatrix} \tag{8}$$

The augmented system (7) must still be a controllable system. Thus, $\mathbf{G}$ in the augmented system (7) must satisfy the following condition:

$$rank \begin{bmatrix} \mathbf{A}_p & \mathbf{A}_p\mathbf{B}_p & \cdots & \mathbf{A}_p^{n+m_y-1}\mathbf{B}_p \end{bmatrix} = n + m_y \tag{9}$$

We design a sliding mode controller according to (7) and (8). As the sliding vector includes $\mathbf{z}$, namely the integrator defined by (4), the proposed control law belongs to an integral

sliding mode controller. The controller design process can be divided into two steps: constructing a sliding surface and determining a control law. The following text describes the design of the sliding function. Because the sliding function plays a major role in SMC systems, many techniques have been proposed for designing the sliding function [27]–[29]. In this study, the Lyapunov method is used to design the sliding function. This popular and simple method is based on the principle of Lyapunov theory [28]. Let us assume that the state feedback matrix $\mathbf{K}_p$ is obtained using the pole assignment method; thus, all the eigenvalues of $\mathbf{A}_p - \mathbf{B}_p\mathbf{K}_p$ are present in the left half of the complex plane. Then, the control input is selected as $\mathbf{u} = -\mathbf{K}_p\mathbf{x}_p + \mathbf{v}$, and the system expression in (7) can be rewritten as follows:

$$\begin{aligned}\dot{\mathbf{x}}_p &= \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\left(-\mathbf{K}_p\mathbf{x}_p + \mathbf{v} + \mathbf{d}_m(\mathbf{x}, t)\right) + \mathbf{f} \\ &= \left(\mathbf{A}_p - \mathbf{B}_p\mathbf{K}_p\right)\mathbf{x}_p + \mathbf{B}_p\left(\mathbf{v} + \mathbf{d}_m(\mathbf{x}, t)\right) + \mathbf{f} \quad (10)\end{aligned}$$

where $\bar{\mathbf{A}}_p = \mathbf{A}_p - \mathbf{B}_p\mathbf{K}_p$. Because $\bar{\mathbf{A}}_p$ is a Hurwitz matrix, a unique positive definite matrix $\mathbf{P}$ exists that satisfies the following Lyapunov equation:

$$\bar{\mathbf{A}}_p^T\mathbf{P} + \mathbf{P}\bar{\mathbf{A}}_p = -\mathbf{I} \quad (11)$$

For analyzing the system stability, let us consider the following Lyapunov candidate function:

$$V_1(\mathbf{x}_p) = \mathbf{x}_p^T\mathbf{P}\mathbf{x}_p \quad (12)$$

The time derivative of this function is expressed as follows:

$$\dot{V}_1(\mathbf{x}_p) = -\mathbf{x}_p^T\mathbf{x}_p + 2\mathbf{x}_p^T\mathbf{P}\mathbf{B}_p\left(\mathbf{v} + \mathbf{d}_m\right) + 2\mathbf{x}_p^T\mathbf{P}\mathbf{B}_r\mathbf{d}_r \quad (13)$$

If the unmatched noise $\mathbf{d}_r$ does not exist and $\mathbf{B}_p^T\mathbf{P}\mathbf{x}_p = \mathbf{0}$, the aforementioned function can be rewritten as follows:

$$\dot{V}_1(\mathbf{x}_p) = -\mathbf{x}_p^T\mathbf{x}_p < 0, \quad \forall\mathbf{x}_p \neq \mathbf{0} \quad (14)$$

The aforementioned equation indicates that $V_1(\mathbf{x}_p)$ is a Lyapunov function, and we can ensure that $\lim_{t\to\infty}\mathbf{x}_p(t) = \mathbf{0}$. Therefore, in the absence of $\mathbf{d}_r$, we can select the original sliding function as follows:

$$\mathbf{s}_0 = \mathbf{C}_0\mathbf{x}_p = \mathbf{B}_p^T\mathbf{P}\mathbf{x}_p \in \mathbf{R}^{m_u \times 1} \quad (15)$$

The stability of the closed-loop system without $\mathbf{d}_r$ can be guaranteed once the system is under the sliding mode $\mathbf{s}_0 = \mathbf{0}$.

To further simplify the control law and make the system enter the sliding mode immediately from the beginning, we assume the practical sliding function as follows:

$$\mathbf{s} = \mathbf{C}\mathbf{x}_p - \mathbf{C}_1\mathbf{x}(0) \quad (16)$$

where

$$\mathbf{C} = (\mathbf{C}_0\mathbf{B}_P)^{-1}\mathbf{C}_0 = \begin{bmatrix}\mathbf{C}_1 & \mathbf{C}_2\end{bmatrix}, \quad \mathbf{C}_1 \in \mathbf{R}^{m_u \times n}, \ \mathbf{C}_1 \in \mathbf{R}^{m_u \times m_y} \quad (17)$$

and

$$\mathbf{C}\mathbf{B}_p = \mathbf{I}_{m_u} \quad (18)$$

In (16), we introduce an initial value term $\mathbf{C}_1\mathbf{x}(0)$. Because this term does not appear in the equivalent control [30], we do not need to consider it when analyzing the stability of the sliding mode. In addition, because of adding the initial value $\mathbf{C}_1\mathbf{x}(0)$, the system enters the sliding mode $\mathbf{s}\left(\mathbf{x}_p(0)\right) = \mathbf{0}$ immediately after startup. Therefore, the system is not affected by the matched disturbance $\mathbf{d}_m$ throughout its operation. The system is only affected by the unmatched disturbance $\mathbf{f}$. However, $\mathbf{f}$ is only a constant vector and thus does not affect the system stability, except for causing a steady-state error, that is, $\mathbf{x}_p(\infty) \to \bar{\mathbf{x}}_p$, where $\bar{\mathbf{x}}_p$ is a constant vector. Although the system indicates a constant error vector $\bar{\mathbf{x}}_p$, $\dot{\mathbf{x}}_p(\infty) \to \mathbf{0}$ can be guaranteed. From the expression $\mathbf{x}_p = \begin{bmatrix}\mathbf{x} \\ \mathbf{z}\end{bmatrix}$ and (3), we obtain the following equation:

$$\dot{\mathbf{z}}(\infty) = \mathbf{e}(\infty) = \mathbf{y}(\infty) - \mathbf{y}_d \to \mathbf{0} \quad (19)$$

Therefore, we can reach the control goal.

Let us define a control law as follows:

$$\mathbf{u} = -\mathbf{K}\mathbf{x}_p - (\sigma + \delta)\frac{\mathbf{s}}{\|\mathbf{s}\|} \quad (20)$$

where

$$\mathbf{K} \equiv \left(\mathbf{C}\mathbf{B}_p\right)^{-1}\mathbf{C}\mathbf{A}_p = \mathbf{C}\mathbf{A}_p \quad (21)$$

The parameter σ is a known positive constant, and the disturbance satisfies the boundary condition.

$$\|\mathbf{d}_m(\cdot) + \mathbf{C}\mathbf{f}\| < \delta \quad (22)$$

Consider the following Lyapunov function:

$$V_2(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T\mathbf{s} \quad (23)$$

Differentiating $V_2(\mathbf{s})$ with respect to time yields the following equation:

$$\begin{aligned}\mathbf{s}^T\dot{\mathbf{s}} &= \mathbf{s}^T\mathbf{C}\left(\mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\left(\mathbf{u} + \mathbf{d}_m(\mathbf{x}, t)\right) + \mathbf{f}\right) \\ &= \mathbf{s}^T\mathbf{C}\left(\mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\left(-\mathbf{C}\mathbf{A}_p\mathbf{x}_p - (\sigma + \delta)\frac{\mathbf{s}}{\|\mathbf{s}\|}\right.\right. \\ &\quad \left.\left. + \mathbf{d}_m(\mathbf{x}, t)\right) + \mathbf{f}\right) \\ &= \mathbf{s}^T\left(-(\sigma + \delta)\frac{\mathbf{s}}{\|\mathbf{s}\|} + \mathbf{d}_m(\mathbf{x}, t) + \mathbf{C}\mathbf{f}\right) \\ &< -(\sigma + \delta)\|\mathbf{s}\| + \delta\|\mathbf{s}\| \\ &= -\sigma\|\mathbf{s}\| \quad (24)\end{aligned}$$

Therefore, the controller (20) guarantees that the sliding mode $\mathbf{s} = \mathbf{0}$ can be maintained after a limited time.

Because the original SMC law includes a discontinuous function that usually causes an undesired chartering, the following equation can be obtained:

$$\mathbf{u} = -\mathbf{K}\mathbf{x}_p - (\sigma + \delta)\mathbf{sat}(\mathbf{s}, \varepsilon) \quad (25)$$

where

$$\mathbf{sat}(\mathbf{s}, \varepsilon) = \begin{cases}\dfrac{\mathbf{s}}{\|\mathbf{s}\|}, & \text{if } \|\mathbf{s}\| > \varepsilon \\ \dfrac{\mathbf{s}}{\varepsilon}, & \text{otherwise}\end{cases} \quad (26)$$

To reduce the number of controller parameters, we assume that the plant and controller satisfy the following conditions:

(a) The nominal plant model of (1) is known, that is, the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}$ are known in advance.

(b) The weighting matrix $\mathbf{W}$ in (4) has the form $\mathbf{W} = w\mathbf{I}_{m_y}$. By using (8), we can determine the matrices $\mathbf{A}_p$ and $\mathbf{B}_p$.

(c) The eigenvalues of matrix $\bar{\mathbf{A}}_p$ in (11) only change with a single parameter. For example, let $\bar{\mathbf{A}}_p$ has $n + m_y$ multiple roots $-\lambda$ or different roots such as $-\lambda, -1.01\lambda, -1.02\lambda, \ldots$, where $\lambda > 0$. Thus, we can solve the unique positive definite matrix $\mathbf{P}$ satisfying the continuous Lyapunov equation (11).

(d) The scalar parameter ρ is defined as follows:

$$\rho \equiv \sigma + \delta \tag{27}$$

Note that the parameter δ, which is defined in boundary condition (22), is usually unknown in a practical system.

According to the aforementioned assumptions (a)-(d), we define the vector of the controller parameters as follows:

$$\mathbf{p} = \begin{bmatrix} \lambda & w & \rho \end{bmatrix} \tag{28}$$

Next, we can apply the proposed method to search for the optimal controller parameters.

III. N–M SIMPLEX ALGORITHM

Nelder and Mead proposed a simplex method, which is a numerical method for solving multidimensional unconstrained optimization problems [21]. We usually refer to this method as the N–M simplex method or downhill simplex algorithm to distinguish it from Dantz's simplex method, which is commonly used in linear programming.

The N–M simplex algorithm is designed to solve N-dimensional unconstrained optimization problems of the following form:

$$\min_{\mathbf{p} \in \mathbf{R}^N} J(\mathbf{p}) \tag{29}$$

where $J(\mathbf{p})$ is defined as an objective function (also called the performance index, target function, and cost function). Controllers are implemented using microcontrollers or industrial computers. As such, with technological advancements in hardware, controller execution has become increasingly rapid. Regardless of whether the integral or the derivative is used in the control law, even if a controller is realized using the simplest numerical method, the discrete simulation outcome will approach a continuous-type system. Therefore, the continuous performance index was employed in this study, as shown in (30a). The discrete form used in the program implementation is presented in (30c):

$$J(\mathbf{p}) = \frac{1}{(\Delta t)^2} \int_0^{T_M} t \left(\|\mathbf{e}(t)\| + W \|\dot{\mathbf{u}}(t)\| \right) dt \tag{30a}$$

$$\approx \frac{1}{(\Delta t)^2} \sum_{k=1}^{M} k \, \Delta t \left(\|\mathbf{e}(k)\| + W \frac{\|\mathbf{u}(k) - \mathbf{u}(k-1)\|}{\Delta t} \right) \Delta t$$

$$\tag{30b}$$

$$= \sum_{k=1}^{M} k \left(\|\mathbf{e}(k)\| + \frac{W}{\Delta t} \|\mathbf{u}(k) - \mathbf{u}(k-1)\| \right) \tag{30c}$$

where $\mathbf{p}$ is a vector comprising all the controller parameters, such as those in (28); T_M is the index calculation time; Δt is the sampling time; $M = T_M/\Delta t$ is a positive integer; $\mathbf{e}(k)$ and $\mathbf{u}(k)$ represent the output error and control input at the kth instant, respectively; W is the weighting of $\|\dot{\mathbf{u}}(t)\|$ and yields $\|\mathbf{e}(t)\|$ and $\|\dot{\mathbf{u}}(t)\|$ at the closest order of magnitude as possible such that $\|\mathbf{e}(t)\|$ and $\|\dot{\mathbf{u}}(t)\|$ have a similar effect on the performance index. Because the characteristics of the performance index and the results of the optimization search are not affected when the performance index is multiplied by a positive constant, we multiply the continuous performance index by $1/(\Delta t)^2$ to simplify (30a) into the discrete performance index (30c). To prevent chattering, the practical control law is adopted, as presented in (25). Moreover, because actual control inputs have a saturation limit, even if the system trajectory is maintained within the sliding layer, the system may still converge to a limit cycle. To prevent chattering and limit cycles, which are both undesirable, we add a penalty function $W \|\dot{\mathbf{u}}(t)\|$ to the performance index (30a).

After the form of the objective function is determined, the N–M simplex method generates a sequence of simplices, where each simplex is defined by $N + 1$ distinct vertices, namely $\mathbf{p}_0, \ldots, \mathbf{p}_N$, whose corresponding function values are $J_0, \ldots, J_N$. The points $\mathbf{p}_0, \ldots, \mathbf{p}_N$ are assumed to be sorted such that $J_0 \leq \cdots \leq J_{N-1} < J_N$, and $\bar{\mathbf{p}}$ represents the centroid of points $\mathbf{p}_0, \ldots, \mathbf{p}_{N-1}$. In each iteration, simplex transformations in the N–M simplex method are controlled by the parameters α, β, and γ. These parameters should satisfy the following conditions:

$$0 < \beta < 1, \quad 0 < \alpha < \gamma \tag{31}$$

These parameters have typical values of $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$. The values of α, γ, β, and $-\beta$ yield the reflection point $\mathbf{p}_r$, expansion point $\mathbf{p}_e$, outer contraction point $\mathbf{p}_c$, and inner contraction point $\mathbf{p}_{cc}$, respectively. The objection functions at these four points are denoted as J_r, J_e, J_c, and J_{cc}, respectively. If none of the four points represents an improvement in the current worst point $\mathbf{p}_N$, the algorithm shrinks the points $\mathbf{p}_1, \ldots, \mathbf{p}_N$ toward the lowest $\mathbf{p}_0$, thereby producing a new simplex. In the shrinking process, each $\mathbf{p}_i$ is replaced by $\mathbf{p}_0 + 0.5 (\mathbf{p}_i - \mathbf{p}_0)$ for $i = 1, \ldots, N$. A new iteration is automatically triggered after completing the shrinking process. The iterative process is continued until the specified termination criteria are satisfied (e.g., when the iterations reach the allowed maximum number or when the accuracy of the function value J_0 is lower than the default value).

We now explain the geometric phenomenon of the N–M simplex method for a 2D parameter (i.e., $N = 2$) as an example. A typical N–M simplex algorithm can generate a series of simplices. Each simplex comprises three vertices, namely
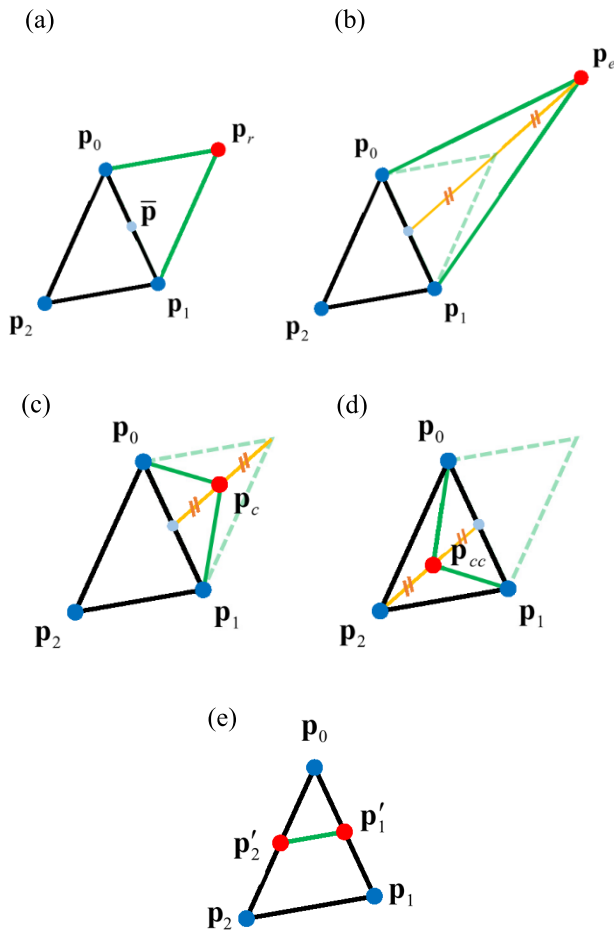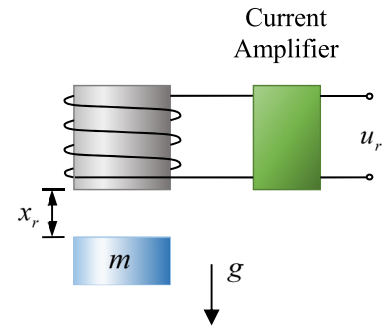
FIGURE 1. (a) Reflection point $\mathbf{p}_r$; (b) expansion point $\mathbf{p}_e$; (c) outer contraction point $\mathbf{p}_c$; (d) inner contraction point $\mathbf{p}_{cc}$; and (e) shrinking the points. $\mathbf{p}_1, \dots, \mathbf{p}_N$.

$\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$, and their corresponding object function values J_0, J_1, and J_2, respectively, where the vertices $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are ordered as $J_0 \leq J_1 < J_2$ and $\bar{\mathbf{p}}$ denotes the centroid of $\mathbf{p}_0$ and $\mathbf{p}_1$. In each iteration, the N–M simplex method examines one or more of the four different ζ values along the line $\bar{\mathbf{p}} + \zeta (\bar{\mathbf{p}} - \mathbf{p}_N)$; in this example, $N = 2$. These four values yield the reflection point $\mathbf{p}_r$, expansion point $\mathbf{p}_e$, outer contraction point $\mathbf{p}_c$, and inner contraction point $\mathbf{p}_{cc}$. The object function values at these points are denoted as J_r, J_e, J_c, and J_{cc}, respectively. If none of the four points represent an improvement in the current worst point $\mathbf{p}_2$, the algorithm shrinks points $\mathbf{p}_1$ and $\mathbf{p}_2$ toward the best point $\mathbf{p}_0$, thereby producing a new simplex. In the shrinking process, each $\mathbf{p}_i$ is replaced by $\mathbf{p}_0 + 0.5 (\mathbf{p}_i - \mathbf{p}_0)$ for $i = 1, 2$. Upon producing a new simplex, a new iteration is automatically triggered. This iterative process is continued until the specified termination criteria are satisfied. Fig. 1 illustrates the different simplices described in the aforementioned text.

1. Fig. 1(a) displays the simplex for the reflection point $\mathbf{p}_r$, which is expressed as follows:

$$\mathbf{p}_r = \bar{\mathbf{p}} + \alpha (\bar{\mathbf{p}} - \mathbf{p}_2) \tag{32}$$

FIGURE 2. Schematic of the magnetic levitation system.

2. Fig. 1(b) depicts the simplex for the expansion point $\mathbf{p}_e$, which is expressed as follows:

$$\mathbf{p}_e = \bar{\mathbf{p}} + \gamma (\bar{\mathbf{p}} - \mathbf{p}_2) \tag{33}$$

3. Fig. 1(c) illustrates the simplex for the outer contraction point $\mathbf{p}_c$, which is expressed as follows:

$$\mathbf{p}_c = \bar{\mathbf{p}} + \beta (\bar{\mathbf{p}} - \mathbf{p}_2) \tag{34}$$

4. Fig. 1(d) shows the simplex for the inner contraction point $\mathbf{p}_{cc}$, which is expressed as follows:

$$\mathbf{p}_{cc} = \bar{\mathbf{p}} - \beta (\bar{\mathbf{p}} - \mathbf{p}_2) \tag{35}$$

5. Fig. 1(e) presents the simplex for shrinking the points $\mathbf{p}_1$ and $\mathbf{p}_2$. The following equation is obtained:

$$\mathbf{p}'_i = \mathbf{p}_0 + 0.5(\mathbf{p}_i - \mathbf{p}_0), \ \ i = 1, 2 \tag{36}$$

The flowchart and pseudocode of the N–M simplex algorithm are shown in Appendixes A and B, respectively.

IV. SIMULATION RESULTS
Let us consider a magnetic levitation system, whose schematic is depicted in Fig. 2.

The symbol definition and simulation conditions for this system are defined as follows:

x_r (mm): real air gap
$x_d = 0.5$ mm (desired gap)
u_r (A): real control input
$m = 0.5$ kg (mass)
$g = 9800$ mm/s^2 (acceleration of gravity)
$k_c = 54000$ Nmm^2/A^2 (constant)

where u_r is subjected to the following saturation condition:

$$u_r = \begin{cases} 5, & \text{if } u_r > u_{max} = 5 \\ u_r, & \text{if } u_{min} \leq u_r \leq u_{max} \\ 0, & \text{if } u_r < u_{min} = 0 \end{cases} \tag{37}$$

The dynamical equation of the magnetic levitation system can be represented as follows [31]–[33]:

$$m\ddot{x}_r = mg - k_c \left(\frac{u_r}{x_r}\right)^2 \tag{38}$$

or

$$\ddot{x}_r = g - \frac{k_c}{m}\left(\frac{u_r}{x_r}\right)^2 \tag{39}$$

The following equations are defined: $x_r = x_d + x$ and $u_r = u_d + u$. Expanding the right-hand side of (39) into a Taylor series about the operation point (x_d, u_d), where $g = \frac{k_c}{m}\left(\frac{u_d}{x_d}\right)^2$, the following equation is obtained:

$$\ddot{x} = \frac{2k_c u_d^2}{mx_d^3}x - \frac{2k_c u_d}{mx_d^2}u + d \tag{40}$$

where d denotes the higher-order terms. The state and output equations of the linearized system are defined as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{d} \tag{41a}$$

$$y = \mathbf{g}\mathbf{x} \tag{41b}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{42a}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ \dfrac{2k_c u_d^2}{mx_d^3} & 0 \end{bmatrix} \tag{42b}$$

$$\mathbf{d} = \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{42c}$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ -\dfrac{2k_c u_d}{mx_d^2} \end{bmatrix} \tag{42d}$$

$$\mathbf{g} = [1, 0] \tag{42e}$$

Then, we define a new state as follows:

$$z = w\int_0^t e\,d\tau = w\int_0^t x\,d\tau, \quad z(0) = 0 \tag{43}$$

The expanded linearized system is represented as follows:

$$\dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{b}_p u + \mathbf{f} \tag{44a}$$

$$y = \mathbf{g}_p\mathbf{x}_p \tag{44b}$$

where

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{x} \\ z \end{bmatrix}, \quad \mathbf{A}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ w & 0 \end{bmatrix}, \quad \mathbf{b}_p = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix},$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{0} \\ -wx_d \end{bmatrix}, \quad \mathbf{g}_p = \begin{bmatrix} \mathbf{g} & 0 \end{bmatrix} \tag{45}$$

We now design a sliding mode controller for the magnetic levitation system according to the aforementioned description. First, by applying the pole assignment method, the feedback gain $\mathbf{k}$ can be found such that the eigenvalue of $(\mathbf{A}_p - \mathbf{b}_p\mathbf{k})$ is a triple root $-\lambda$, where $\lambda > 0$. Thus, the Lyapunov equation (11) has a unique positive definite solution $\mathbf{P}$. Then, let the sliding function can possess the following form:

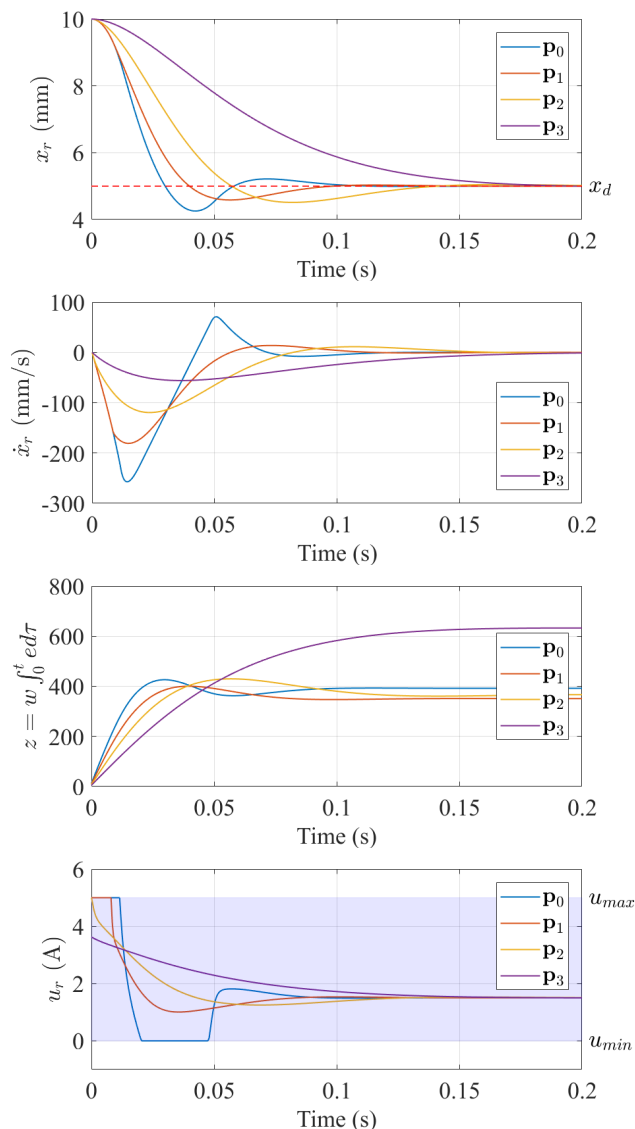$$s = \mathbf{c}\mathbf{x}_p - \mathbf{c}_1\mathbf{x}(0) \tag{46}$$

FIGURE 3. Time responses of the magnetic levitation system controlled by the four sets of controller parameters listed in Table 1.

where

$$\mathbf{c} = (\mathbf{c}_0\mathbf{b}_p)^{-1}\mathbf{c}_0 = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix}, \quad \mathbf{c}_1 \in \mathbf{R}^{1\times n}, \ \mathbf{c}_2 \in \mathbf{R} \tag{47}$$

The control law is designed as follows:

$$u = -\mathbf{k}\mathbf{x}_p - \rho \cdot sat(s, \varepsilon) = -\mathbf{c}\mathbf{A}_p\mathbf{x}_p - \rho \cdot sat(s, \varepsilon) \tag{48}$$

where the scalar parameter ρ is defined as in (28). The controller must determine three parameter values, which are represented by the vector $\mathbf{p} = \begin{bmatrix} \lambda & w & \rho \end{bmatrix}$. To avoid chattering, we introduce a sliding layer in the actual control law. The parameter ε of the sliding layer needs to only select a constant so that all initial controllers do not produce chattering. Therefore, the parameters to be searched in the simplex method do not include ε.

Now, let us apply the N–M simplex method to search the optimal parameters. In this example, we set the sampling time
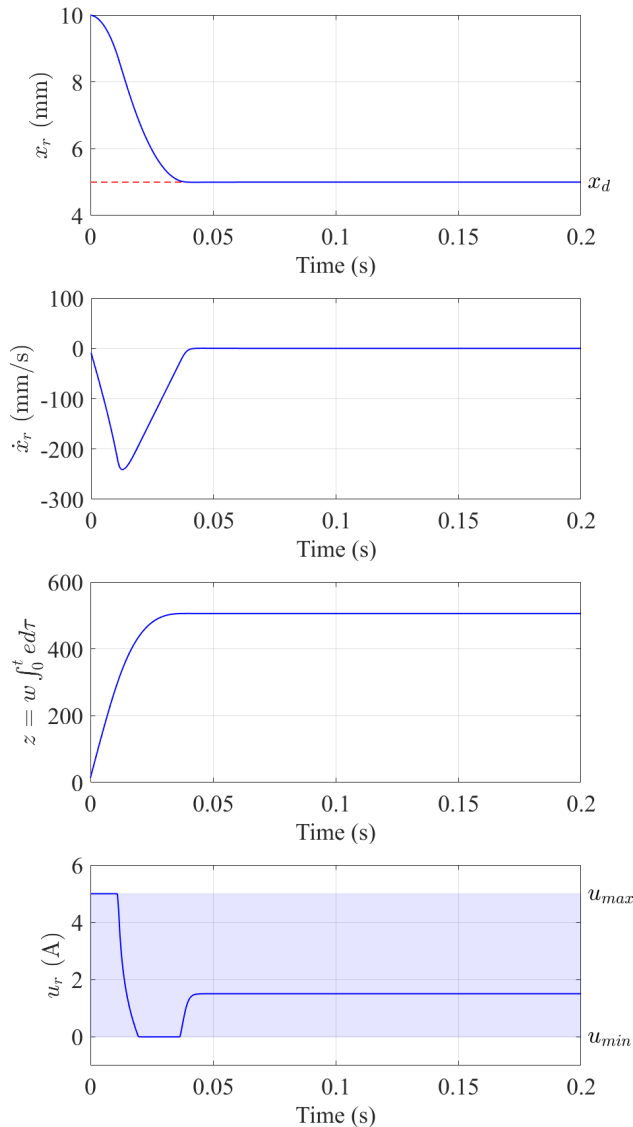
FIGURE 4. Time response of the magnetic levitation system controlled by the sliding mode controller, whose parameters are searched using the N–M simplex method.
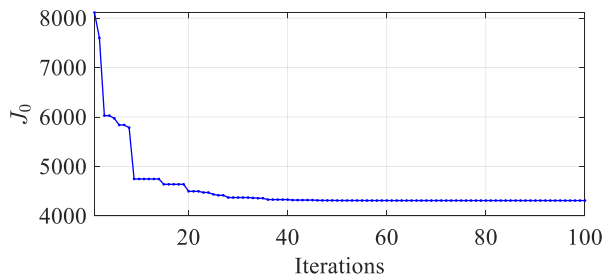
FIGURE 5. Convergence graph of performance index J_0 when the simplex method based on the initial vertices given in Table 1 is used to search for the controller parameters of the magnetic levitation system.

to 0.0005 s and the simulation time to 0–0.2 s. Because the magnitudes of the error and control input of this system are comparable, we consider the weighting W in the performance index (30c) equal to Δt and express the norm as an absolute

TABLE 1. Initial controller parameters for the magnetic levitation system.

i	$\mathbf{p}_i = \begin{bmatrix} \lambda & w & \rho \end{bmatrix}$	$J_i = J(\mathbf{p}_i)$
0	$\mathbf{p}_0 = \begin{bmatrix} 70 & 5000 & 500 \end{bmatrix}$	$J_0 = 8.1138 \times 10^3$
1	$\mathbf{p}_1 = \begin{bmatrix} 80 & 4000 & 2000 \end{bmatrix}$	$J_1 = 9.1231 \times 10^3$
2	$\mathbf{p}_2 = \begin{bmatrix} 100 & 3000 & 1000 \end{bmatrix}$	$J_2 = 20.236 \times 10^3$
3	$\mathbf{p}_3 = \begin{bmatrix} 120 & 2000 & 500 \end{bmatrix}$	$J_3 = 55.882 \times 10^3$

value. The final performance index is as follows:

$$J(\mathbf{p}) = \sum_{k=1}^{400} k \left(|e(k)| + |u(k) - u(k-1)| \right) \qquad (49)$$

Furthermore, because the proposed integral sliding mode controller has three key adjustable parameters, namely λ, w, and ρ, we must obtain four sets of parameter combinations before we use the N–M simplex method to search for the optimization parameters. In the following text, we present the heuristic selection principles for the aforementioned three parameters.

The parameter λ is considerably affected by the characteristics of the open-loop system. For example, without loss of generality, let us assume that the dominant pole of an open-loop system located at the left half of the complex plane is $-\lambda_0$. If we wish to quicken the response of the closed-loop system through a feedback controller, the dominant pole, which is $-\lambda$, has a reference range between approximately $-\lambda_0$ and $-2\lambda_0$ (the actual value still depends on factors such as the power provided by the actuator, nonlinearity, disturbance, and transient response performance).

The parameter w is related to the speed of error convergence. The larger the value of w, the faster is the error convergence. Therefore, we attempt at selecting the initial value of w between $10\lambda_0$ and $100\lambda_0$.

Finally, $\rho \equiv \sigma + \delta$. To avoid chattering, the actual application uses the control law presented in (25). Because the second term in (25) is $-\rho \cdot \mathbf{sat}(\mathbf{s}, \varepsilon)$, the high-gain control force is used in the sliding mode controller to fight against the disturbance. Because of the high gain, the aforementioned term reaches saturation if $\|\mathbf{s}\| > \varepsilon$. For the controller to maintain a reasonable high-gain characteristic, when determining the value of ρ, we can select a value between approximately $50u_{\max}$ and $200u_{\max}$. According to the aforementioned systematic empirical method, four sets of parameters for the sliding mode controller of the magnetic levitation system, which construct the initial simplex, are listed in Table 1. In this example, when the width ε of the boundary layer is equal to 1, the four sets of controller parameters in the initial simplex will not cause chattering in the system. Therefore, we set ε as 1 in the optimization search process.

The control results for the aforementioned four sets of controllers are presented in Table 1 and Fig. 3.

Then, we use the aforementioned four parameter sets as the four vertices of the initial simplex. The iteration of the
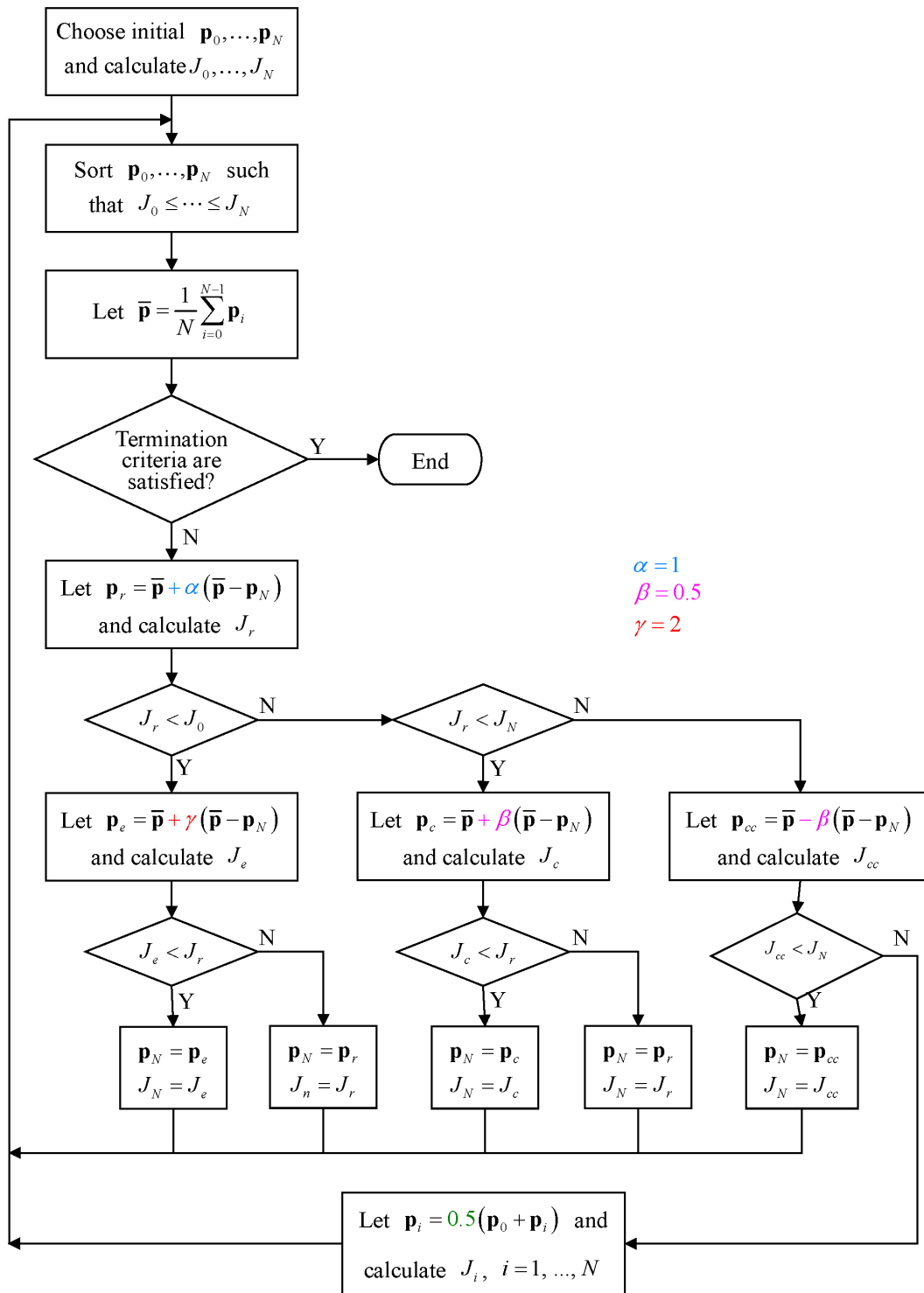
FIGURE 6. Flowchart of the N–M simplex algorithm.

searching loop is terminated if 100 iterations are completed or if the following condition is met:

$$\frac{J_n - J_0}{J_0} < 10^{-4} \tag{50}$$

The result obtained using the N–M simplex optimal search method is displayed in Fig. 4, where the calculation process ends after 56 iterations and the resulting parameters are $\lambda = 59.862$, $w = 5597.5$, and $\rho = 705.23$. The corresponding performance index is $J = 4310.4$, which is considerably

lower than the corresponding values for the initial four parameter sets.

V. CONCLUSION

Sliding mode controllers have been widely used in the control field due to their simple algorithm and robustness to disturbance. This paper proposes a design method for an error-integral-type sliding mode controller. The main characteristics of the proposed controller are that it needs to determine only three key parameters regardless of the system order. Most previous studies have only discussed the inequalities or boundary conditions that the controller parameters must satisfy to ensure system stability. However, in practical applications, even if the closed-loop system is stable, its response performance or state trajectory may not be suitable. Theoretically, each controller parameter contains an infinite number of choices, and all parameters affect the system response. Therefore, the determination of the optimal parameters of the controller has become an important step in designing a superior controller. To address this problem, we used the N–M simplex method to assist the designer in determining the optimal controller parameters more efficiently. Finally, we performed a numerical simulation on a magnetic levitation system to illustrate the feasibility and effectiveness of the proposed method.

APPENDIX A
FLOWCHART OF THE N–M SIMPLEX ALGORITHM
See Figure 6.

APPENDIX B
PSEUDOCODE OF THE N–M SIMPLEX ALGORITHM
Define $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$
Choose the initial $\mathbf{p}_0, \ldots, \mathbf{p}_N$ and calculate $J_0, \ldots, J_N$
while true
    Sort $\mathbf{p}_0, \ldots, \mathbf{p}_N$ such that $J_0 \le \cdots \le J_N$
    $\bar{\mathbf{p}} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{p}_i$
if termination criteria are satisfied
    Terminate execution of while loop
end
    $\mathbf{p}_r = \bar{\mathbf{p}} + \alpha (\bar{\mathbf{p}} - \mathbf{p}_N)$ (Reflection)
    Calculate J_r
if $J_r < J_0$
    $\mathbf{p}_e = \bar{\mathbf{p}} + \gamma (\bar{\mathbf{p}} - \mathbf{p}_N)$ (Expansion)
    Calculate J_e
    if $J_e < J_r$
       $\mathbf{p}_N = \mathbf{p}_e$
       $J_N = J_e$
    else
       $\mathbf{p}_N = \mathbf{p}_r$
       $J_N = J_r$
    end
else if $J_r < J_n$
    $\mathbf{p}_c = \bar{\mathbf{p}} + \beta (\bar{\mathbf{p}} - \mathbf{p}_N)$ (Outer contraction)
    Calculate J_c

if $J_c < J_r$
    $\mathbf{p}_N = \mathbf{p}_c$
    $J_N = J_c$
else
    $\mathbf{p}_N = \mathbf{p}_r$
    $J_N = J_r$
end
else
    $\mathbf{p}_{cc} = \bar{\mathbf{p}} - \beta (\bar{\mathbf{p}} - \mathbf{p}_N)$ (Inner contraction)
    Calculate J_{cc}
    if $J_{cc} < J_N$
       $\mathbf{p}_N = \mathbf{p}_{cc}$
       $J_N = J_{cc}$
    else
       for $i = 1, \ldots, N$
          $\mathbf{p}_i = 0.5 (\mathbf{p}_0 + \mathbf{p}_i)$ (Shrink)
   Calculate J_i
       end
    end
    end
end
Print out $\mathbf{p}_0$ and J_0

REFERENCES

[1] S. V. Emelyanov, "Control of ?rst order delay systems by means of an astatic controller and nonlinear corrections," *Autom. Remote Control*, vol. 8, pp. 983–991, 1959.

[2] S. V. Emelyanov, (in Russian), *Variable Structure Control Systems*. Moscow, Russia: Nauka, 1967.

[3] Y. Itkis, *Control Systems of Variable Structure*. New York, NY, USA: Wiley, 1976.

[4] V. A. Utkin, (in Russian), *Sliding Modes and Their Application in Variable Structure Systems*. Moscow, Russia: Nauka Moscow: Mir, 1978.

[5] V. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Autom. Control*, vol. AC-22, no. 2, pp. 212–222, Apr. 1977.

[6] J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: A survey," *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 2–22, Feb. 1993.

[7] K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 3, pp. 328–342, May 1999.

[8] A. Šabanovic, "Variable structure systems with sliding modes in motion control—A survey," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 212–223, May 2011.

[9] [Online]. Available: https://en.wikipedia.org/wiki/Sliding_mode_control

[10] R. A. DeCarlo, S. H. Zak, and G. P. Matthews, "Variable structure control of nonlinear multivariable systems: A tutorial," *Proc. IEEE*, vol. 76, no. 3, pp. 212–232, Mar. 1988.

[11] J. J. Slotine and S. Sastry, "Tracking control of nonlinear systems using sliding surfaces with application to robot manipulator," *Int. J. Control*, vol. 38, no. 2, pp. 465–492, 1983.

[12] M.-S. Chen, Y.-R. Hwang, and M. Tomizuka, "A state-dependent boundary layer design for sliding mode control," *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1677–1681, Oct. 2002.

[13] C.-C. Fuh, "Variable-thickness boundary layers for sliding mode control," *J. Mar. Sci. Technol.*, vol. 16, no. 4, pp. 286–292, Dec. 2008.

[14] C. J. Fallaha, M. Saad, H. Y. Kanaan, and K. Al-Haddad, "Sliding-mode robot control with exponential reaching law," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 600–610, May 2011.

[15] A. Bartoszewicz, "A new reaching law for sliding mode control of continuous time systems with constraints," *Trans. Inst. Meas. Control*, vol. 37, no. 4, pp. 515–521, Apr. 2015.

[16] P. Li, L. Xiong, Z. Wang, M. Ma, and J. Wang, "Fractional-order sliding mode control for damping of subsynchronous control interaction in DFIG-based wind farms," *Wind Energy*, vol. 23, no. 3, pp. 749–762, Mar. 2020.

[17] M. L. Corradini and G. Orlando, "Linear unstable plants with saturating actuators: Robust stabilization by a time varying sliding surface," *Automatica*, vol. 43, no. 1, pp. 88–94, Jan. 2007.

[18] A. Ferrara and M. Rubagotti, "A sub-optimal second order sliding mode controller for systems with saturating actuators," *IEEE Trans. Autom. Control*, vol. 54, no. 5, pp. 1082–1087, May 2009.

[19] A. Bartoszewicz and A. Nowacka-Leverton, "ITAE optimal sliding modes for third-order systems with input signal and state constraints," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1928–1932, Aug. 2010.

[20] W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimisation and evolutionary operation," *Technometrics*, vol. 4, no. 4, pp. 441–461, Nov. 1962.

[21] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[22] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.

[23] K. I. M. McKinnon, "Convergence of the Nelder–Mead simplex method to a nonstationary point," *SIAM J. Optim.*, vol. 9, no. 1, pp. 148–158, Jan. 1998.

[24] D. Byatt, "Convergent variants of the Nelder–Mead algorithm," M.S. thesis, Degree Master of Sci. Math., Univ. Canterbury, Christchurch, New Zealand, 2000.

[25] C. J. Price, I. D. Coope, and D. Byatt, "A convergent variant of the Nelder–Mead algorithm," *J. Optim. Theory Appl.*, vol. 113, no. 1, pp. 5–19, Apr. 2002.

[26] C.-C. Fuh, H.-H. Tsai, and H.-C. Lin, "Parameter identification of linear time-invariant systems with large measurement noises," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 2874–2878.

[27] V. I. Utkin, *Sliding Modes in Control and Optimization*. New York, NY, USA: Springer, 1992.

[28] W.-C. Su, S. V. Drakunov, and Ü. Özgüner, "Constructing discontinuity surfaces for variable structure systems: A Lyapunov approach," *Automatica*, vol. 32, no. 6, pp. 925–928, Jun. 1996.

[29] C. E. Edwards and S. K. Spurgeon, *Sliding Mode Control: Theory and Application*. London, U.K.: Taylor and Francis, 1998.

[30] A. F. Filippov, *Differential Equation with Discontinuous Rightsides*. Boston, MA, USA: Kluwer, 1988.

[31] D. L. Trumper, S. M. Olson, and P. K. Subrahmanyan, "Linearizing control of magnetic suspension systems," *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 4, pp. 427–438, Jul. 1997.

[32] J. Zhang, X. Wang, and X. Shao, "Design and real-time implementation of Takagi–Sugeno fuzzy controller for magnetic levitation ball system," *IEEE Access*, vol. 8, pp. 38221–38228, 2020.

[33] Y. Sun, J. Xu, G. Lin, and S. M. Mumtaz, "Deep learning based semi-supervised control for vertical security of maglev vehicle with guaranteed bounded airgap," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4431–4442, Jul. 2021.
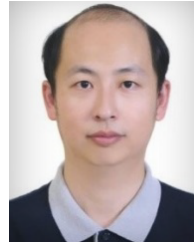
CHYUN-CHAU FUH was born in Taiwan, in 1968. He received the B.S. and Ph.D. degrees in mechanical engineering from the National Central University, Taoyuan, Taiwan, in 1991 and 1996, respectively. He is currently a Professor with the Department of Mechanical and Mechatronic Engineering, National Taiwan Ocean University. His research interests include automatic control, automation, artificial intelligence, fuzzy theory, machine vision, mathematical optimization, nonlinear dynamics, robotics, signal processing, system identification, and mechatronics.

HSUN-HENG TSAI was born in Taiwan, in 1967. He received the B.E. degree in mechanical engineering from Chen Kung University, in 1991, and the Ph.D. degree from the Department of Mechanical Engineering, National Central University, Taoyuan, Taiwan, in 2001. He has been with the Department of Biosystems Engineering, Pintung University of Science and Technology, Taiwan. He is currently a Professor with the Institute of Biomechatronics Engineering, where he teaches and conducts research in the area of mechatronic, automatic control, and measurement. His research interests include nonlinear control, biomechatronics systems, and biomedical engineering.

• • •