Design of Sliding Mode Control for Complex Nonlinear Models: A Numerical Approach

AHMED REHAN AND AHMED AL-DURRA (Senior Member, IEEE)

1Department of Robotics Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
2Electrical Engineering and Computer Science Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Ahmed Al-Durra (ahmed.aldurra@ku.ac.ae)

ABSTRACT In this work, a numerical approach of formulating Sliding Model Control (SMC) is proposed to deal with complex nonlinear models of certain physical systems. Analytical formulation of SMC for such models is not possible as SMCs require algebraic manipulation of system equations, which is not achievable when the system model has strong nonlinearities. For such models, it is proposed to solve the SMC design using a numerical method that deals with numerical values instead of variables or functions in the equations. Mostly, SMC control law is dependent on the bounds of variables that are present in the mathematical expression of time derivatives of the sliding variable. To compute these bounds, a numerical approach is carried out using uncertainty bounds of the system’s parameters. Also, a numerical approach is used to compute time derivatives of nonlinear functions that are required in the formulation of SMC. Various forms of SMCs are investigated in this respect, including the basic first-order SMC and a second-order SMC. The proposed framework is generalized as well, making it compatible with a wide range of nonlinear models whose algebraic manipulation is not possible. A prototype example of a nonlinear model of a boiler is used as a proof-of-concept. The simulation results are promising and prove the efficacy of the proposed approach.

INDEX TERMS Sliding mode control, second order sliding mode control, boiler, nonlinear control, super-twisting control.

I. INTRODUCTION

Sliding mode controllers (SMC) belong to the class of controllers, which are robust and immune to parametric uncertainties of the model. The design of SMC control constitutes two phases; design of a stable sliding manifold that characterizes the system’s dynamics in reduced dimensional form; design of a discontinuous control law that switches across the manifold in its neighborhood while maintaining the global reachability of sliding manifold [1]. The basic objective of the discontinuous control is to direct velocities of error trajectories towards the sliding manifold. The manifold itself is designed according to certain criteria, mostly using the Lyapunov stability theory, which enforces the reachability of manifold. Such reachability is to be strictly satisfied in order to achieve asymptotic or terminal stability of global error space.

Various implementations show the compatibility of SMCs with real-world systems [2], [3]. This is because, for practical systems, many parameters have to be measured, and the uncertainties in the measurements can be troublesome for modeling and control purposes. To accomplish a robust controlled response of the system, it is always desired to use robust controllers like SMCs, where SMCs can show a high degree of robustness in the presence of uncertainties in the model.

The basic idea of sliding mode control emerged in the 1950s. The SMC in the original form constituted some weaknesses. For instance, the control input exhibits chattering in the vicinity of the sliding manifold, and although the manifold is reached asymptotically, it can sometimes lead to longer convergence times. Many modifications of SMC were done later in an attempt to tackle these shortcomings. Some of the works developed new kinds of manifolds and reaching laws so as to improve regulation and tracking performance. Others used Higher Order Sliding Modes (HOSMs) in sliding planes, where higher derivatives of the sliding variable are used to constitute sliding dynamics [4]. Under this category of HOSMs, Second Order SMC (2-SMC) controllers are the most researched and widely used in various practical
applications [5]. Many works have been reported that focus specifically on designing of unique forms of 2-SMCs [6], 2-SMCs with time delay [7], 2-SMCs with adaptive fuzzy control [8], etc. Super Twisting Controllers (STCs) under this category are particularly famous due to two reasons. One is, they are computationally efficient as they don’t require the calculation of the derivative of the sliding variable, which is a necessity in other 2-SMCs formulations. Other is, they can work with the systems whose relative degree is one with respect to the sliding variable. Many articles prove the applicability of STCs with various kinds of physical systems. They have been analyzed, modified, and tested in conjunction with Neural networks [9], PDIs [10], backstepping controllers [11], and many others. There are many other forms of 2-SMC besides STC. However, a careful review of these SMC contributions determines that all the SMC literature is focused only on new designs of SMCs while disregarding the issue of the design complexity. Like other model-based control schemes, SMC uses algebraic manipulation of modeling equations to derive the control law, where the design process becomes very much cumbersome if the system models involve too many nonlinearities, lengthy equations, etc. Through the years, the complexity of new SMC techniques has grown, making them a tough candidate for control of highly nonlinear and complex models.

One form of SMC, known as integral SMC, is used to improve the tracking performance of SMC to enhance robustness. Although integral SMC increases the order of sliding phase dynamics, which becomes equal to the system’s dimension, the trajectory is made to start exactly at the sliding manifold, thereby eliminating the reaching phase [12]. Another line of SMC is dedicated to modifying the reaching dynamics by using hyperbolic terms [13], non-switching law [14], power terms [15], and many others [16] in control formulation. The target of such works is to develop unique kinds of sliding variable dynamic equations to improve the dynamics of reaching and sliding phases. Some methods attempt to improve the performance using different forms of sliding manifolds, for example, nonlinear manifolds [17], time-varying manifolds [18], moving manifolds [19], fuzzy integral sliding manifolds [20], etc. In the same line, PID manifolds have also been designed to ensure improved convergence to origin [21] or enhanced robustness for time-varying uncertain systems [22]. However, such manifolds increase the complexity of the SMC design process. This is because the analytical formulation of SMC requires lengthy algebraic computations, which cannot be carried out easily for complex models, which is further aggravated by increasing the complexity of the manifolds.

Another branch of SMCs is the terminal sliding mode, which tackles the shortcoming of asymptotic convergence of regular SMCs, by bringing error trajectories to manifold in a finite and shorter time. The terminal SMCs have a shortcoming that the control law may face the problem of singularity, which causes unbound growth of control inputs. This singularity issue has been dealt with different approaches [23]–[25]. However, all nonlinear manifold SMCs and especially terminal SMCs have the same problem, i.e., they use complex formulations of manifolds, which does not solve the problem of the design process for complex nonlinear models.

Artificial Neural Networks (ANNs) are also used in the SMC design process. The Radial Basis Function Neural Networks (RBFNNs) have been extensively used in this regard. In [26], Terminal SMC was improved by using an adaptive RBFN to estimate unknown function in the model of Permanent Magnet Linear Synchronous Motor, and the adaptive law was used to estimate the upper bound of the model’s uncertainty. In another work [27], the authors used RBF in the design process of terminal SMC. With the proposed approach, the switching gain was adaptively adjusted using RBFNN to make the SMC robust and eliminate the chattering problem. In [28], RBFNN was used to model the dynamics of the robot manipulator, and the model was used in conjunction with SMC to execute a controlled response. All such works use ANN-based SMCs for modeling the unknown/unmodeled dynamics of the system or the sliding variables. However, they rarely touch the issue of complexity, and hence the proposed approaches are only compatible with dynamics models of limited complexity.

Regardless of all forms of SMCs, the mathematical designing of SMCs is considered as the most challenging part of it. The basic mathematical tools to design SMC are well developed and mature, but they are tedious or impossible to apply for complex nonlinear models. The complication increases with the increase in the number of nonlinearities and length of modeling equations. Particularly, the bounds and time derivatives of various functions become very hard to compute, as the complexity in the model increases. Such complexity issue is encountered in many models such as of a boiler, octarotor, quadrotor with tilting rotors, etc. Considering the case of the quadrotor, for instance, SMC has been used extensively with a simplified model of the quadrotor, but hardly has it been designed for the quadrotor with tilting rotors [29], [30]. This is because the additional four degrees of freedom afforded by the tilting mechanism increases the complexity of the model to a great extent (see [31] for example). The same is the case with robotic manipulators of sophisticated design or the octarotors [32], where the accurate model is too complex to design SMC unless a simplified model is used.

In this work, we aim to target this shortcoming when the design of SMC is not possible for complicated nonlinear models. In particular, we choose the nonlinear model of the boiler as the basis of our design framework. We use Astrom and Bell’s model of boiler [33], which in fact, is the impetus of this work since all the previous works have focused on using a linearized model or simplified model of boilers for control design. Many practical system models pose the same complexity problem in the SMC design, due to which the proposed framework is also generalized in this work. Our target model [33] was derived using fundamental laws of conservation of energy and momentum. It is due to the model
complexity that the previously reported works have used either a linearized model or a simplified model of boiler for the SMC control [34]–[36] etc. For instance, in [34], the SMC design was carried out using a PID sliding manifold for a linearized model of the boiler. In [36], the linear extended state observer (LESO) was implemented with the notion of compiling the total disturbances of the system as one state. The estimated disturbance was used in the formulation of SMC. Again, a simple model was used to design SMC as the level dynamics take extremely complicated mathematical form, which cannot be dealt with model-based control like SMC.

As a usual practice, the control law of the SMC sliding variable is mathematically dependant on the bounds and time derivatives of the variables involved in the expression of the sliding variable and its derivatives. For complicated models like boilers, the problem of analytical computation of the time derivatives and the variable bounds is either very complex or unsolvable. In this work, we propose a solution to this problem which is summarized as follows:

1) A numerical approach is proposed to solve the SMC design problem for complex nonlinear models with the aim of making the SMC design easily realizable without performing complex analytical computations.
2) The SMC design process constitutes the determination of the bounds of variables and the time derivatives of variables that are involved in the expression of the sliding variable. It is proposed that all such bounds and derivatives can be simply computed numerically. A deviation factor that depends on the parametric uncertainties is incorporated in these bounds, making them globally applicable around an operating point.
3) With the numerical computation of bounds and time derivatives, the SMC equations acquire a regular form for which any of the existing SMC algorithms can be applied with great ease.

The proposed method is applied and tested with various forms of SMCs on the complex model of a boiler system. It is proven that using a stable sliding manifold, the SMC control can regulate the level and pressure dynamics of a boiler under harsh action of steam disturbance, hence verifying that the proposed control technique works well in a simulation environment and has a potential of performing effectively in a real-time scenario as well.

The paper is organized as follows. Section II is used to highlight issues behind SMC formulation of complex models and present the motivation of proposed work. Section III explains the model of the boiler and the associated complexities of the model. In Section IV, the boiler model is used as a build-up for numerical SMC formulation. Section V deals with few 2-SMCs control techniques. In Section VI, we propose numerical algorithms to compute time derivatives and bounds of variables involved in various SMC formulations. In Section VII, the general formulation of the SMC numerical framework is presented. Simulations and conclusions are presented in section VIII and section IX, respectively.

II. PROBLEM FORMULATION

Many control techniques involve strict use of system models to carry out the design phase of the control problem. SMCs are such controllers that highly rely on algebraic manipulation of modeling equations to formulate the control. In the SMC design procedure, the main aim is to make sliding manifolds an invariant set of global error space. The approach of the Lyapunov theory is used for the design process, which requires the time derivatives and the bounds of various variables involved in the SMC equations. It is observed that the major complexity lies in the analytical computation of the time derivatives, the variables bounds, and the typical form of the modeling equation, which is comfortably handled using available tools of SMC formulations.

In the SMC design, the complexity of various analytical computations can be avoided if the SMC design is carried out using a numerical approach. For the time derivatives in SMC equations, we can consider a numerical approach of differentiation using forward or backward difference method. Such numerical differentiation can convert complex analytical functions $y = h(x)$ into a linearized form $y = C(x, t)x$, a form that can be used easily for the SMC design using the available SMC tools.

The other problem is the bounds of complicated functions that can be addressed in a similar way. The idea is to use a numerical approach here to convert the uncertainty bounds of system parameters into the bounds of SMC variables. Such conversion can be carried out by discretizing the uncertainty bounds and using each discrete point to compute the numerical value of concerning functions in the sliding variable equation. The bounds of the functions would correspond to the maximum and minimum of the numerical values of the concerning functions, which are otherwise impossible to derive or compute using the analytical approach.

We implement the above procedure for a boiler system of the model [33]. For completeness, we present the boiler model first, and then we describe how a typical first-order SMC (1-SMC) can be used to control the state of pressure using Utkin’s approach of SMC formulation. The level dynamics, in contrast to the pressure, are a complicated nonlinear function of boiler variables so we develop a numerical approach in an algorithmic form to design the SMC for such a complicated case of nonlinear output. In a similar fashion, we formulate a numerical approach for designing 2-SMC for the pressure control of the boiler system. Particularly, we numerically derive the time derivatives and bounds of SMC variables, which are used to formulate the 2-SMC for the pressure control system.

III. BOILER WORKING PRINCIPLE AND MODEL

A boiler is a device that is used to convert water into steam for various industrial applications. It is considered to be dangerous equipment since it operates at extremely high pressures and temperatures, which can be life-threatening if not adequately controlled.
Many complications are associated with controlling a boiler system. Among them, the most intricate challenge is model-based control. It is because the model of the boiler system has many nonlinearities; analytical nonlinearities (variable multiplications, logarithmic and high power terms, etc.) and the interpolation nonlinearities of the steam table. Very effective control is required to be implemented in order to keep the states of the boiler under safety level. The controller should be versatile enough to perform all the essential tasks of stabilizing the unstable dynamics, performing the setpoint tracking for all outputs, and controlling the outputs without affecting the MIMO couplings that exist between all the inputs and the outputs.

In this work, we use Astrom and Bell [33] dynamic model of boiler, which uses four states of the boiler to model all the dynamics of the boiler. These states are Drum Pressure, Total Volume of Water, Steam Quality, and Steam Volume in Drum. The state of drum pressure is also an output. The other output is the drum level, which is modeled as an independent nonlinear function of all the states. The overall model [33] was derived using laws of conservation of energy, mass, and momentum for individual components as well as for the whole system globally. The governing equations of the system take the following nonlinear state-space form:

\[
\begin{align*}
& a_{11} \frac{dV_{wt}}{dt} + a_{12} \frac{dp}{dt} = \dot{m}_{f_w} - \dot{m}_s, \\
& a_{21} \frac{dV_{wt}}{dt} + a_{22} \frac{dp}{dt} = Q + \dot{m}_{f_w} h_{fw} - \dot{m}_s h_s, \\
& a_{32} \frac{dp}{dt} + a_{33} \frac{d\alpha_{st}}{dt} = Q - \alpha_{st} h_s \dot{m}_{dc}, \\
& a_{42} \frac{dp}{dt} + a_{43} \frac{d\alpha_{st}}{dt} + a_{44} \frac{dV_{sd}}{dt} = \rho_s \frac{V_{sd}^2}{T_d} (V_{st}^\alpha - V_{sd}) + \frac{h_{fw} - h_w}{h_{fw}} \dot{m}_{f_w}
\end{align*}
\]

where the coefficients \(a_{ij}\)'s are

\[
\begin{align*}
& a_{11} = \rho_w - \rho_s, \\
& a_{12} = V_{wt} \frac{\partial \rho_w}{\partial p} + V_st \frac{\partial \rho_s}{\partial p}, \\
& a_{21} = \rho_w h_w - \rho_s h_s, \\
& a_{22} = V_{wt} \left( \frac{\partial h_w}{\partial p} + \rho_w \frac{\partial h_w}{\partial p} \right) + V_st \left( \frac{\partial h_s}{\partial p} + \rho_s \frac{\partial h_s}{\partial p} \right) - V_t + m_m C_p \frac{\partial T_m}{\partial p}, \\
& a_{32} = (\rho_w \frac{\partial h_w}{\partial p} - \alpha_{st} h_s \frac{\partial h_w}{\partial p}) (1 - \alpha_{sv}) V_r + ((1 - \alpha_{sv}) h_s \frac{\partial \rho_s}{\partial p} + \rho_s \frac{\partial h_s}{\partial p} ) V_r \\
& + (\rho_s + (\rho_w - \rho_s) \alpha_{st}) h_s V_r \frac{\partial \alpha_{sv}}{\partial p} - V_r + m_m C_p \frac{\partial T_p}{\partial p}, \\
& a_{33} = ((1 - \alpha_{sv}) \rho_s + \alpha_{st} \rho_w) h_s V_r \frac{\partial \alpha_{sv}}{\partial \alpha_{st}}, \\
& a_{42} = V_{sd} \frac{\partial \rho_s}{\partial p} + \frac{1}{h_{fw}}
\end{align*}
\]

\[
\begin{align*}
& + \alpha_{st} (1 + \beta) V_r (\alpha_{sv}) \frac{\partial \rho_s}{\partial p} + ((1 - \alpha_{sv}) (\partial \rho_w)/\partial p) \\
& + (\rho_s - \rho_w) \frac{\partial \alpha_{sv}}{\partial p}, \\
& a_{43} = \alpha_{st} (1 + \beta) (\rho_s - \rho_w) V_r \frac{\partial \alpha_{sv}}{\partial \alpha_{st}}, a_{44} = \rho_s
\end{align*}
\]

where \(P, V_{wt}, \alpha_{st}\) and \(V_{sd}\) are the state variables. Other variables that are present in the model are computed using the following relations:

\[
\begin{align*}
& \frac{\partial \alpha_{sv}}{\partial p} = \frac{1}{(\rho_{w} - \rho_{s}) \Gamma (\frac{\partial \rho_{s}}{\partial p} - \frac{\partial \rho_{w}}{\partial p})} \times (1 + \frac{\rho_{w}}{\rho_{s}} 1 - \frac{\rho_{s} + \rho_{w}}{\eta_{s}} \ln(1 + \eta)) \quad (3) \\
& \frac{\partial \alpha_{sv}}{\partial \alpha_{t}} = \frac{\alpha_{s} (\frac{\partial \alpha_{sv}}{\partial \alpha_{t}})}{\alpha_{s} \eta_{s} \eta} \quad (4) \\
& \frac{q_{dc}}{k} = 2 \alpha_{w} A_{dc} (\alpha_{w} - \alpha_{s}) \frac{\partial \alpha_{sv}}{\partial \alpha_{t}} V_r \quad (5) \\
& \alpha_{t} = \frac{\alpha_{s} (\frac{\partial \alpha_{sv}}{\partial \alpha_{t}})}{\alpha_{s} \eta_{s}} \quad (6) \\
& \frac{q_{ct}}{k} = \frac{h_{w} - h_{t}}{h_{c}} \frac{1}{h_{c}} (\frac{\partial \alpha_{sv}}{\partial \alpha_{t}}) V_r + \frac{V_{st}}{\eta_{s}} \frac{\partial h_{s}}{\partial \alpha_{t}} V_r - V_t + m_m C_p \frac{\partial T_m}{\partial \alpha_{t}} \frac{dp}{dt} + (\rho_{w} - \rho_{s}) V_r \frac{\partial \alpha_{sv}}{\partial \alpha_{t}} \frac{d \alpha_{t}}{dt} \quad (7)
\end{align*}
\]

In model (1), the output level is not explicitly defined as a state. One way would be to use Lie derivatives approach and represent level as one of the states in a new model, but it is not doable given the complexity of nonlinear equations. To make it easier, the level is defined as a separate output variable which is computed using the following set of equations:

\[
\begin{align*}
& l = V_{wd} + V_{sd} \quad (8) \\
& V_{wd} = V_{wt} - V_{dc} - (1 - \alpha_{sv}) V_r \quad (9) \\
& \alpha_{v} = \frac{\rho_{w}}{\rho_{w} - \rho_{s}} \frac{(1 + \frac{\alpha_{s}}{\alpha_{v}})}{\alpha_{s}} (1 - \frac{\alpha_{w} - \alpha_{s}}{\alpha_{v}}) \ln \left( 1 + \frac{\alpha_{w} - \alpha_{s}}{\alpha_{v}} - \alpha_{t} \right) \quad (10) \\
& \rho_{s} = \text{LUT}_1 (P) \\
& \rho_{w} = \text{LUT}_2 (P) \\
& h_{s} = \text{LUT}_3 (P) \\
& h_{w} = \text{LUT}_4 (P) \quad (11)
\end{align*}
\]

where \(\text{LUT}_i\)'s are lookup tables, which for boilers, are referred to as steam tables that provide values of \(\rho_i\)'s and \(h_i\)'s as a function of the sate of steam pressure \(P\). A polynomial function of a second or third-order is sufficient to interpolate the values of the steam table.

The given model (1) assumes a typical state-space form, but the real complexity lies in the algebraic manipulation of
nonlinear coefficients of (2). One issue in the MIMO system like the boiler is that there exists a mathematical coupling between all the inputs and outputs of the system. So it is necessary to determine which input-output pair has to be formed for constructing the control loops. This can be done by analyzing the mathematical model or doing hit-and-trial simulation to analyze the impact of each input on all the outputs. For the case of boiler, it is observed that the following input-output pairs are most efficient in executing a controlled response of boilers:

- Feedwater rate – level control.
- Heating rate – pressure control.

The level control is implemented under the name of three-term control, in which the feedforward element of the steam flow rate is included to nullify the variations in the steam variable control. Figure 1 shows the complete control loop structure. The steam flow rate is used in a feedback control loop to stabilize the level output against dynamic disturbances of feedwater rate.

Consider model (1) of our system in the following form:

\[
\begin{align*}
\dot{e}_L(t) &= L_d - L(t) \quad (12) \\
\dot{m}_{fw}(t) &= \dot{m}_s(t) + f_L(e_L(t)) \quad (13)
\end{align*}
\]

where \(f_L\) is the control function to be formulated in the next section.

Similarly, the pressure control loop is designed using the control input of the heating rate:

\[
\begin{align*}
\dot{e}_P(t) &= P_d - P(t) \quad (14) \\
\dot{Q}(t) &= f_Q(e_P(t)) \quad (15)
\end{align*}
\]

where \(f_Q\) is the control function to be formulated in the next section.

Given the complex nature of the model, as evident from (1)-(15), analytical derivation of any kind of control technique is impossible. Typical analysis of Lipschitz properties of the functions is also not doable, and deriving an analytical solution of the system response is barely possible due to such nonlinear structure of the model.

### IV. NUMERICAL FORMULATION OF SMC FOR BOILER

The basic principle of SMC design is to transform an ‘n’ dimensional tracking problem into a one-dimensional stabilization problem. This achieves a controlled response in two phases, commonly known as the reaching phase and the sliding phase. Various methods are available that proposes different designing schemes for both the manifold design and the control law. For this work, we use Utkin’s approach for the basic derivation of 1-SMC. However, it is to be noted that most of SMC design methods are based on the models having strict feedback integrating form (normal form or canonical form). For such models, the relative degree of output variables is at least two, hence designing a stable sliding manifold is simple. In the case of the boiler model, all the outputs and states have a relative degree of one, and hence we use a trivial formulation of sliding manifold using one variable. Later, we will upgrade the sliding manifold to include the two output variables. At this point, we avoid designing manifold as a linear combination of all states, because for highly complicated systems, it is not practice to design manifold as a linear combination of all states, because for highly complicated nonlinear models, it is unclear whether the linear combination of all the states would produce a stable sliding manifold or not. That’s why we restrict the design process for the two variables, which are in fact the outputs of our system.

Consider model (1) of our system in the following form:

\[
\begin{align*}
\begin{bmatrix}
V_{wt} \\
p \\
\alpha_{st} \\
V_{sd}
\end{bmatrix}
&= \begin{bmatrix}
V_{wt} \\
p \\
\alpha_{st} \\
V_{sd}
\end{bmatrix} \\
&= \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -h_g m_{dc} & 0 \\
0 & 0 & 0 & -\rho_s/T_d
\end{bmatrix}
+ \begin{bmatrix}
h_f - h_s \\
h_f \\
h_f - h_w \\
h_c
\end{bmatrix}
\begin{bmatrix}
\dot{m}_s \\
\dot{m}_{df}
\end{bmatrix} \\
\begin{bmatrix}
0 & 1 & 0 & 0 \\
h_f - h_s & h_f & 1 & 0 \\
h_f - h_w & h_f - h_w & h_c & 0
\end{bmatrix}
\begin{bmatrix}
\dot{m}_s \\
\dot{m}_{df}
\end{bmatrix}
\end{align*}
\]

where \(\dot{m}_{df} = \dot{m}_{fw} - \dot{m}_s\) is differential feedwater rate, and \(a_{ij}\) are functions of all states as given in (2). \(V_{sd}\) is taken 0 for simplicity, however in real operation it has some value.

The above model takes the following notational form:

\[
E(x)\dot{x} = A(x)x + B(x)u \quad (17)
\]

where,

\[
\begin{align*}
E(x) &= \begin{bmatrix}
a_{11} & a_{12} & 0 & 0 \\
a_{21} & a_{22} & 0 & 0 \\
a_{31} & a_{32} & a_{33} & 0 \\
a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}, x &= \begin{bmatrix}
V_{wt} \\
p \\
\alpha_{st} \\
V_{sd}
\end{bmatrix}, u = \begin{bmatrix}
\dot{m}_s \\
\dot{m}_{df}
\end{bmatrix}, \\
A(x) &= \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -h_g m_{dc} & 0 \\
0 & 0 & 0 & -\rho_s/T_d
\end{bmatrix}, \\
B(x) &= \begin{bmatrix}
0 & 1 & 0 & 0 \\
h_f - h_s & h_f & 1 & 0 \\
h_f - h_w & h_f - h_w & h_c & 0
\end{bmatrix}
\end{align*}
\]
For ease of notation, we omit the index ‘x’, and use the following form:

\[ E\dot{x} = Ax + Bu \] (19)

The state (19) can be rewritten as:

\[ x = E^{-1}Ax + E^{-1}Bu \] (20)

The equation (20) uses the inverse of matrix E, which is only calculable in the numerical form given the nonlinear structure of matrix E. Such modeling form of boiler (20) is of great essence in the formulation of numerical SMC. Since it is impossible to have \( E^{-1} \) and (20) in analytical form, the only way to implement this equation is in the numerical way, i.e., by computing (20) numerically at each instant of a simulation or sampling instant of a real-time operation. So, at each instant, we produce the numerical form of the equation (20) and do the subsequent computations of numerical SMC design.

There are two outputs to be controlled, the drum level ‘L’, and the steam pressure ‘P’. For now, we focus on control of the pressure. The level would be controlled by using a PID with nominally tuned parameters.

The pressure is the second state in the state vector, and so is given as:

\[ P = C_p x \] (21)

where, \( C_p = [0100] \). For our system, the input relative degree of all the states and outputs is one. In this respect, we proceed with a trivial form of the sliding manifold, where the sliding variable ‘s’ is simply a function of the error between output and setpoint. This form of the sliding manifold is also used in [6], [37], [38], and can be considered as a sliding point instead of a manifold and is commonly used in Lyapunov control design. We begin with this sliding point as a bottom-up approach for more complex manifold design later in this article.

Consider the sliding variable as a function of output pressure.

\[ s = \hat{G}\hat{P} \] (22)

where \( \hat{P} = P - P_d \), and \( P_d = 8500 \) is the desired state value of steam pressure.

We break the input vector into two inputs as follows:

\[ u = \begin{bmatrix} \dot{m}_s \\ \dot{m}_{df} \\ Q \end{bmatrix} \rightarrow u_Q = Q \text{ and } u_{SF} = \begin{bmatrix} \dot{m}_s \\ \dot{m}_{df} \end{bmatrix} \] (23)

In this way, the ‘B’ matrix is dissociated as:

\[ B = \begin{bmatrix} 0 & 1 & 0 \\ h_f - h_s & h_f & 1 \\ 0 & 0 & 1 \\ h_f - h_w & h_f - h_w & 0 \end{bmatrix} \rightarrow B_Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } B_{SF} = \begin{bmatrix} 0 & 1 & 0 \\ h_f - h_s & h_f & 1 \\ 0 & 0 & 1 \\ h_f - h_w & h_f - h_w & 0 \end{bmatrix} \] (24)

For now, we consider the case \( \dot{P}_d = 0 \), i.e. constant desired pressure and so, the sliding variable derivative is given as:

\[ \dot{s} = \hat{G}\dot{\hat{P}} = GCP\dot{X} = GC_p(E^{-1}Ax + E^{-1}Bu) \] (25)

Computing \( u_{eq} \) using Utkin’s approach [1],

\[ \dot{s} = 0 \]

\[ GC_p(E^{-1}Ax + E^{-1}Bu) = 0 \] (26)

Decomposing input ‘u’ into \( u_Q \) and \( u_{SF} \).

\[ GC_p(E^{-1}Ax + E^{-1}B_{SF} u_{SF} + E^{-1}B_Q u_Q) = 0 \] (27)

\[ u_{Qeq} = -\left(GC_pE^{-1}B_Q\right)^{-1}GCP_x(E^{-1}Ax + E^{-1}B_{SF} u_{SF}) \] (28)

\[ u = u_{Qeq} - ksign(s) \] (29)

Since the modeling inaccuracies cannot realize an ideal equivalent control of (29), the dynamics of the system get partially canceled out and remain there in the form of a perturbation. This perturbation factor is taken into account while designing the switching gain ‘k’.

For equivalent control, we use the nominal form of the model (19):

\[ E_o(x)\dot{x} = A_o(x)x + B_o(x)u \] (30)

\[ u_{Qeq} = -\left(GC_pE^{-1}B_Q\right)^{-1}GCP_x(E^{-1}Ax + E^{-1}B_{SF} u_{SF}) \] (31)

where, \( E_o, A_o \) and \( B_o \) are nominal matrices that are dependent on the model parameters selected, for example, by using average values of the uncertainty bounds. Since the actual parameters of the system cannot be known, the differences between the nominal matrices and actual matrices will always be present, but the good point is that such differences are bounded and numerically calculable. We compute these bounds in Section VI.

In order to use Utkin’s method of SMC [1], we need to determine the switching gain ‘k’ of (29). We can do so by using the nominal form of (29) and (31) in the derivative of sliding variable \( \dot{s} \) of (27).

\[ \dot{s} = GC_p(E^{-1}Ax + E^{-1}B_{SF} u_{SF} + E^{-1}B_Q u_Q) \] (32)

\[ = GC_p(E^{-1}Ax + E^{-1}B_{SF} u_{SF} + E^{-1}B_Q u_Q) \]

\[ \times B_Q \left(-\left(GC_pE^{-1}B_Q\right)^{-1}\right) \]

\[ \times GC_p\left(E_o^{-1}A_o x + E_o^{-1}B_{oSF} u_{SF} - ksign(s)\right) \] (33)

\[ = GC_pE^{-1}(Ax + B_{SF} u_{SF}) + GC_pE^{-1} \]
In this work, the gain ‘k’ is determined numerically at each instant, using the following equation.

\[ k = k_0 + \eta \]  

(39)

where \( k_0 = |\gamma| \) and \( \eta \) is arbitrary that determines the reaching time of error trajectories and can be manipulated at the expense of control energy.

Equation (39) constitutes a numerical form of SMC for pressure output, where all the terms of expressions are available in numerical form and so is \( k_o \). The parameter \( k_0 \) is in fact a function of the states i.e. \( k_0 = k_o (x) \), and change dynamically with all the states. Hence the parameter \( k = k_0 + \eta \) can be regarded as an adaptive gain of the switching control of the SMC.

A. SMC FOR LEVEL CONTROL

The numerical design of SMC for the level output uses a modified procedure of the SMC design for the pressure output. This is because the output level is not explicitly defined as a state as the output pressure is. The level is modeled using the nonlinear equation (8), which further depends on expressions (2) and (9). Combining all the expressions analytically to produce a single equation is not possible, so we use a general expression to denote level output as:

\[ l = h_L(x) \]  

(40)

where ‘x’ is a state vector, and ‘h’ is a composite function that includes all the functional dependence of level on all the states.

Here, we refer to the same sequence of SMC derivation of the previous section where equations (28), (29) and (38) are the key components of the numerical formulation of SMC. The derivation of SMC, according to these equations, is contingent on the format of the pressure output, which is in linear form i.e. \( y = P = C_p x \). Hence it is required to convert (40) into a similar linearized form, so that an equivalent of equations (28), (29) and (38) can be derived for output level as well. For that, we first examine (25) and write it in a general form:

\[ \dot{\gamma} = G \dot{\bar{\gamma}} = G \bar{C} \dot{x} = G \bar{C} (\dot{A}x + Bu) \]  

(41)

where \( \bar{A}, \bar{B}, \bar{C} \) can be any general matrices, but for our case \( \bar{A} = E^{-1} A, \bar{B} = E^{-1} B, \bar{C} = C_p \) or \( C_L \). The variable ‘y’ is the concerned output, which is either level or pressure or both as a vector.

From (40), the time derivative of output level takes the following form.

\[ \dot{L} = \frac{\partial h_L}{\partial x} \dot{x} \]  

(42)

where \( \partial h_L / \partial x \) is not available in an analytical form due to the complexity of the nonlinear function \( h_L \). Here we use the approach of numerical differentiation using forward difference method, where \( \partial h_L / \partial x \) is numerically represented in the following form.

\[ \frac{\partial h_L}{\partial x} \simeq \frac{\Delta h_L}{\Delta x} = \left[ \frac{\Delta h_1}{\Delta x_1} \frac{\Delta h_2}{\Delta x_2} \frac{\Delta h_3}{\Delta x_3} \frac{\Delta h_4}{\Delta x_4} \right] \]  

(42)

Essentially (42) represents the gradient of \( h_L \) using the numerical approach. The gradient is computed in the form of an algorithm rather than a mathematical formulation. We use the Ab Initio method (or known as the First Principle) of differentiation to form this algorithm, which can be generalized into vector form or even a matrix form for computing vector and matrix time derivatives.

The algorithm is, now onwards, termed as Ab Initio algorithm and is as follows:

- Slightly increment each state, i.e. \( x_i = x_i + \Delta x \)
- Measure the corresponding change in \( L \), i.e. \( \Delta L \).
- Compute \( \Delta L / \Delta x_i \)
- Repeat the above steps for all states and create the gradient vector: \( \nabla L \simeq \frac{\Delta h_L}{\Delta x} = \left[ \frac{\Delta h_1}{\Delta x_1} \frac{\Delta h_2}{\Delta x_2} \frac{\Delta h_3}{\Delta x_3} \frac{\Delta h_4}{\Delta x_4} \right] = \hat{C} \nabla L \)
- The numerical value of \( \nabla L \), when multiplied with \( \dot{x} \) gives an estimate of \( \dot{L} \).
Using the above algorithm gives us a linear form \( \dot{L} = \hat{C}_{VL} \dot{x} \) similar to \( \dot{P} = C_p \dot{x} \), which can be analytically used to represent the level in the form of (41):

\[
\dot{L} = \nabla L \dot{x} = \hat{C}_{VL} \dot{x} = \hat{C}_{VL} \left( E^{-1} A x + E^{-1} B u \right) \tag{43}
\]

Since we use \( \dot{m}_{df} \) to control level, we dissociate input vector in the following form to have an independent representation of differential feedwater rate \( U_{df} = \dot{m}_{df} \).

\[
\dot{s} = G \dot{L} = G \hat{C}_{VL} \left( E^{-1} A x + E^{-1} B_{SQ} u_{SQ} + E^{-1} B_{df} u_{df} \right) \tag{44}
\]

The equation (44) is applied onwards in the SMC design for the level output. A simple test was conducted to prove the applicability of this method. A disturbance signal of the sinusoid is applied using the input of steam flow rate at time \( t = 200 \), and derivatives of level are recorded using the derivatives of level are recorded using the derivative of the input instead of regular input. Since our case system (boiler) involves output variables of relative degree one, we focus on the 2-SMC formulation for the sliding variable of degree 1.

Consider the following expression of the sliding variable:

\[
s = s(t, x) \tag{49}
\]

where \( \dot{x}(t) = f(x) + g(x) u \).

Taking first and second-order derivatives of \( s \),

\[
\dot{s} = \frac{\partial}{\partial t} s(t, x) + \frac{\partial}{\partial x} [s(t, x)] f(x) + g(x) u \tag{50}
\]

\[
\ddot{s} = \frac{\partial}{\partial t} \dot{s}(t, x, u) + \frac{\partial}{\partial x} \dot{s}(t, x, u) f(x) + g(x) u + \frac{\partial}{\partial u} \dot{s}(t, x, u) \dot{u}(t) \tag{51}
\]

which can be rewritten in notational form as:

\[
\ddot{s} = \alpha(t, x, u) + \beta(t, x, u) \dot{u}(t) \tag{52}
\]

where the following bounds hold:

\[
0 < \Gamma_m \leq \beta(t, x, u) \leq \Gamma_M \quad |\alpha(t, x, u)| \leq \Phi \tag{53}
\]

For the case of the boiler, we incorporate the system equations into the above equations to determine the time derivatives of \( \dot{s} \). Consider a sliding variable \( \dot{s} \) defined as a linear combination of outputs:

\[
s = G y \tag{54}
\]

For now, we derive 2-SMC for output pressure \( y = P \). The level will be controlled using a secondary controller (for example, PID).
Taking 1st and 2nd order time derivatives and incorporating (20):

\[
\ddot{x} = G\ddot{y} = GC_p \ddot{x} = GC_p \left( E^{-1}Ax + E^{-1}Bu \right)
\]

\[
= GC_pE^{-1}Ax + GC_pE^{-1}Bu
\]

\[
= GC_pE^{-1}Ax + GC_pE^{-1}B_{SFUSF} + GC_pE^{-1}B_{Q\dot{u}Q} \tag{55}
\]

\[
\ddot{x} = GC_pE^{-1}Ax + GC_pE^{-1}Ax + GC_pE^{-1}Ax
\]

\[
+GC_pE^{-1}B_{SFUSF} + GC_pE^{-1}B_{SFUSF}
\]

\[
+GC_pE^{-1}B_{Q\dot{u}Q} + GC_pE^{-1}B_{Q\dot{u}Q} + GC_pE^{-1}B_{Q\dot{u}Q} \tag{56}
\]

where \( u_{SF} \) should be zero, as the concerned input is \( u_Q \), while other inputs are considered constants.

The above equation is in the following form:

\[
\ddot{x} = \alpha(t, x, u) + \beta(t, x, u) \dot{u}(t) \tag{58}
\]

where,

\[
\alpha(t, x, u) = GC_pE^{-1}Ax + GC_pE^{-1}Ax
\]

\[
+GC_pE^{-1}B_{SFUSF} + GC_pE^{-1}B_{SFUSF}
\]

\[
+GC_pE^{-1}B_{Q\dot{u}Q} + GC_pE^{-1}B_{Q\dot{u}Q} + GC_pE^{-1}B_{Q\dot{u}Q} \tag{59}
\]

\[
\beta(t, x, u) = GC_pE^{-1}B_{Q} \tag{60}
\]

Equations (59) and (60) are sufficient for the formulation of 2-SMC. Particularly, these equations can be used to numerically compute the bounds of variables \( \alpha(t, x, u) \) and \( \beta(t, x, u) \). Once the bounds are available, any form of 2-SMC can be designed and implemented. In this work, we use the STC algorithm to prove the compatibility of our numerical approach with the 2-SMC.

The STC law is formulated as:

\[
u(t) = u_1(t) + u_2(t)
\]

\[
u_1(t) = \begin{cases} 
-u & \text{if } |u| > 1 \\
-W\text{sign}(s) & \text{if } |u| \leq 1
\end{cases}
\]

\[
u_2(t) = \begin{cases} 
-\lambda [s_0]^p \text{sign}(s) & \text{if } |s| > s_0 \\
-\lambda [s]^p \text{sign}(s) & \text{if } |s| \leq s_0
\end{cases} \tag{61}
\]

where the following constraints should be satisfied:

\[
W > \frac{\Phi}{\Gamma_m}
\]

\[
\lambda^2 \geq \frac{4\Phi}{\Gamma_m^2}(W + \Phi)
\]

\[
0 < \rho \leq 0.5 \tag{62}
\]

VI. NUMERICAL COMPUTATION OF THE BOUNDS AND THE TIME DERIVATIVES

This section discusses a very important aspect of this work, which is the numerical computation of bounds and time derivatives that are involved in all SMC formulations. From the above equations, especially (36), (59) and (60), it is evident that we may not need the actual values of \( \gamma(x, t), \alpha(x, t) \) and \( \beta(x, t) \), however the bounds of these variables are important in the SMC design process. Once the bounds are available, the constraints (38) can be easily fulfilled to implement first-order SMC or HOSMC algorithms. The bounds are not known apriori and cannot be computed analytically for complicated models like that of boiler. Hence, we use a numerical approach to compute these bounds. For our method, we rely on the assumption that the parametric uncertainties in the system’s original parameters are all known. These uncertainties are also bounded with known constraints in the following form:

\[
k_i \leq \kappa \leq k_m \tag{63}
\]

where ‘\( \kappa \)’ is the vector of systems static parameters that totally depend on the particular structure and configuration of the system. For boilers, each element \( k_i \) of \( \kappa \) belongs to the set \( \Lambda \), i.e.

\[
k_i \in \Lambda
\]

\[= \{ C_p, m_r, A_d, V_d, \eta, l, m_{md}, V_r, m_{mt}, V_{st}, V_1, V^0_{sd}, m_{1} \} \tag{64}\]

The parameter’s nomenclature can be referred to in the Appendix IX-A. The parametric uncertainty (63) can be written for all the individual parameters as:

\[
C_{p1} < C_p < C_{p2}
\]

\[
m_{r1} < m_r < m_{r2}
\]

\[
A_{d1} < A_d < A_{d2}
\]

\[
\vdots
\]

\[
m_{1s1} < m_{st} < m_{1s2} \tag{65}
\]

These uncertainties are transferred into the bounds (38) and (53). The transformation is not straightforward and cannot be done analytically, and hence a numerical computation of bounds is carried out. For this, we evaluate \( \gamma(x, t), \alpha(x, t) \) and \( \beta(x, t) \) at a given operating point using a range of system models chosen from a discretized set satisfying the given domain (65). We keep track of how the maximum and minimum values of \( \gamma(x, t), \alpha(x, t) \) and \( \beta(x, t) \) vary with respect to all extreme values of the uncertainties. The maximum of these variations is termed as deviation factors (\( \Delta\gamma, \Delta\alpha, \Delta\beta \)) that are also used to make a robust estimate of the bounds. This approach can be stated using the following algorithm:

1) Simulate the system in ‘N’ number of iterations, where ‘N’ is the maximum number of combinations of \( k_1 \) and \( k_m \) of (65). Mathematically \( N = 2^{N_p} \), \( N_p \) being total number of uncertain parameters. We may discretize the set (65) into more points at the expense of computational effort to produce more accurate bounds.

2) In each simulation, iteratively set \( k_i \) to one combination of maximum and minimum bounds (\( k_i \) and \( k_m \)) of (65).
3) Compute numerical values of $\gamma(x, t)$ in the case of 1-SMC, or $\alpha(x, t)$ and $\beta(x, t)$ in the case of 2-SMC at each instant.
4) During each $i^{th}$ instant, all $\gamma^i(x, t)$, $\alpha^i(x, t)$ and $\beta^i(x, t)$ would achieve a maximum ($\gamma^i_{m}$, $\alpha^i_{m}$, $\beta^i_{m}$) and minimum value ($\gamma^i_{l}$, $\alpha^i_{l}$, $\beta^i_{l}$). Keep a record of these extreme values.
5) After ‘N’ simulations, compute the maximum deviation in all the extreme values i.e. ($\Delta \gamma^i_{m}$, $\Delta \alpha^i_{m}$, $\Delta \beta^i_{m}$) and ($\Delta \gamma^i_{l}$, $\Delta \alpha^i_{l}$, $\Delta \beta^i_{l}$).
6) The bounds (38) and (53) are transformed by incorporating the deviation factors as follows.

a. For 1-SMC, use:

$$
k = \left( \frac{G_p E^{-1} B_Q}{G_p E^{-1} (A x + B_{SF} u_{SF})} \right)^{-1} \frac{G_p E^{-1} (A x + B_{SF} u_{SF})}{G_p E^{-1} (A x + B_{SF} u_{SF})} \times \left[ - \left( \frac{G_p m E^{-1} B_m}{G_p m E^{-1} A m} \right)^{-1} G_p m E^{-1} A m x + E^{-1} B_{SF m} u_{SF} \right] + \eta + \Delta \gamma_m$$

b. For 2-SMC, use:

$$
\Gamma_M = \max \left[ \beta(t, x, u) - \Delta \beta_m, 0 \right] + \Delta \beta_m
$$

$$
\Gamma_m = \max \left[ \beta(t, x, u) - \Delta \beta_l, 0 \right] - \Delta \beta_l
$$

$$
\Phi = \max \left[ \left| \alpha(t, x, u) \right| + \Delta \alpha_m, 0 \right] + \Delta \alpha_m
$$

$$
\Delta \gamma_m = \max \left[ (G_p E^{-1} B_m - G_p E^{-1} A x) + \eta + \Delta \gamma_m, 0 \right]$$

During regular operation, we adaptively compute the bounds $k$ for 1-SMC, and $\Gamma_m$, $\Gamma_M$, $\Phi$ for 2-SMC. For example, at each time instant, we compute the numerical value of $\beta(t, x, u)$. We adaptively set $\Gamma_m$ as the lowest value of $\beta(t, x, u)$, and $\Gamma_M$, as the highest value of $\beta(t, x, u)$ plus/minus the deviation factors $\Delta \beta_{l or m}$ that arises due to parametric uncertainties of the model. An example Matlab implementation would be in the following way:

If $\beta(t) = \gamma_m$

$\gamma_m = \beta(t) + \Delta \beta_m$;

else

$\gamma_m = \beta(t) - \Delta \beta_l$;

end

In a similar fashion, we numerically compute the bounds $k$ and $\Phi$ of $\gamma(x, t)$ and $\beta(x, t)$, respectively.

A. COMPUTING TIME DERIVATIVES OF MATRICES

The expressions, (59) and (60), involve time derivatives of matrices in the form $E^{-1} A$, $B_{SF}$, $B_Q$. We compute these derivatives numerically, at each instant, using the same Ab Initio algorithm that we used for $\dot{L}$. Since, $\dot{L}$ is a scalar, the approach needs to be applied for all the individual elements of the matrices, or we can do a matrix implementation as well with a little increase in the complexity of the programming code. However, using a matrix approach results in a three-dimensional cube comprising four matrices due to the four states of the boiler system. The cube is multiplied with $\dot{L}$, which gives time derivative of the matrices $E^{-1} A$, $B_{SF}$, $B_Q$.

VII. GENERAL FORMULATION OF NUMERICAL SMC

After developing SMC for the boiler nonlinear model, we can generalize the approach for nonlinear models of any form.

Consider a general form of a nonlinear model:

$$
\dot{x} = f(x, t) + g(x, t) u + \phi(x, t) u_d
$$

$$
y = h(x, t) x
$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $u_d \in \mathbb{R}^d$ and $y \in \mathbb{R}^p$. The input $u_d$ is assumed to be a measurable disturbance.

Let the corresponding nominal model in the following form:

$$
\dot{x} = f_o(x, t) + g_o(x, t) u + \phi_o(x, t) u_d
$$

Consider, a sliding manifold defined as a linear combination of the error states:

$$
s = G \tilde{x}
$$

where we have assumed that the sliding manifold $G \tilde{x}$ is inherently stable. Such stability must be ensured based on the knowledge of the system or by-hit-and-trial simulations. The relative degree of $'s'$ is assumed to be 1. The SMC formulation for the case of the relative degree of 2 is easier and is not discussed in this work. Also, we only consider constant setpoints, i.e. $\dot{x}_d = 0$. However, the approach can be extended to the cases with variable setpoints with little modification.

Differentiating the sliding variable w.r.t time,

$$
\dot{s} = G \ddot{x}
$$

The above sliding manifold can be easily transformed as a linear combination of output errors by using the following relation:

$$
\dot{s} = G \ddot{y} = G \dot{C}_{xy} \ddot{x} = G' \ddot{x}
$$

where $G' = G \dot{C}_{xy}$. The vector $\dot{C}_{xy}$ is computed using the same Ab-initio algorithm that was used to compute the gradient of level $\nabla L_{ij}$ or matrix element $\nabla e_{ij}$ in the previous section.

Using (68) in (71)

$$
\dot{s} = G(f(x, t) + g(x, t) u + \phi(x, t) u_d)
$$

$$
\dot{s} = 0 \rightarrow u_{eq} = (-G g(x, t))^{-1} (G f(x, t) + \phi(x, t) u_d)
$$

Using, $u = u_{eq} - k \text{sign}(s)$. Also, dropping the index $(x, t)$, we can rewrite $\dot{s}$ equation as:

$$
\dot{s} = G f + G g (-G g_o)^{-1} (G (f_o + \phi_o u_d) - k \text{sign}(s)) + G \phi u_d
$$
where the input $u$, has been incorporated from the nominal state \( (69) \).

\[
\dot{s} = \left( Gg (-Gg_o)^{-1} \right) \left( Gg (-Gg_o)^{-1} \right)^{-1} \left( G\dot{f} + Gf_o + \phi_o u_d \right) - k \text{sgn}(s) + \left( Gg (-Gg_o)^{-1} \right)^{-1} \left( G\phi u_d \right)
\]  

(76)

Let,

\[
\gamma = \left( Gg (-Gg_o)^{-1} \right)^{-1} G\dot{f} + Gf_o + \phi_o u_d + \left( Gg (-Gg_o)^{-1} \right)^{-1} G\phi u_d
\]  

(77)

\[
\rightarrow \dot{s} = \left( Gg (-Gg_o)^{-1} \right) (\gamma - k \text{sgn}(s))
\]  

(78)

To maintain reachability condition of the sliding manifold, the following condition must be fulfilled:

\[
k > |\gamma| - \frac{1}{2} \frac{d}{dt} s^2 = \ddot{s} < 0
\]  

(79)

A trivial fulfillment of the above constraint can be carried out using the following:

\[
k = k_0 + \eta
\]  

(80)

where

\[
k_0 = \left| \left( Gg (-Gg_o)^{-1} \right)^{-1} G\dot{f} + Gf_o + \phi_o u_d \right| + \left( Gg (-Gg_o)^{-1} \right)^{-1} G\phi u_d
\]

and $\eta$ is arbitrary.

Similarly, for second-order SMC, we need to compute numerical bounds of variables ($\alpha, \beta$) in the equation of $\ddot{s}$. Suppose that the sliding manifold is a Hurwitz stable linear combination of outputs i.e., $s = G\ddot{y}$, then using (68), we can compute $\ddot{s}$ in numerical form.

\[
\dot{s} = G\ddot{y} = G\dot{Cg}\ddot{x} + G\dot{Cg}\gamma + G\dot{Cg}\phi (x,t) u_d
\]  

(81)

Dropping the indices \((x,t)\):

\[
\dot{s} = G\dot{Cg}\gamma + G\dot{Cg}\gamma u + G\dot{Cg}\phi u_d
\]  

(82)

Let $G' = G\dot{Cg}$. 

\[
\dot{s} = G' f + G' g u + G' \phi u_d = G' (f + g u + \phi u_d)
\]  

(83)

Taking the second-time derivative:

\[
\ddot{s} = G' \ddot{f} + G' \ddot{g} u + G' \ddot{\phi} u_d + G' \phi u_d
\]  

(84)

\[
\rightarrow \ddot{s} = G' f \dddot{x} + G' g \dddot{x} u + G' \dddot{u} + G' \phi u_d
\]  

(85)

where, $\dddot{x}$ is available from (68). The matrices $f, g, \phi$, and $\phi$ are the Jacobians of the functions $f, g$ and $\phi$ respectively with respect to all the states. The Ab-initio algorithm of Section VI-A is used for their computation, which because of brevity, is not discussed again. The numerical computation of $G'_s$ is also carried out using a similar numerical approach that is presented in detail in Section IX-B of the Appendix.

Rearranging (85),

\[
\ddot{s} = G' f \dddot{x} + G' g \dddot{x} u + G' \phi u_d + G' \phi u_d + (G_s \dddot{x})^T (f + g u + \phi u_d) + G'_g u
\]  

(86)

Comparing above with the general form $\ddot{s} = \ddot{a} (x, t, u) + \ddot{b} (t, x, u) \dot{u}(t)$, we have:

\[
\ddot{a} (x, t, u) = G' f \dddot{x} + G' g \dddot{x} u + G' \phi u_d
\]  

(87)

\[
\ddot{b} (t, x, u) = G'_g g (x, t)
\]  

(88)

The algorithm of Section VI can be generalized as follows:

a. For 1-SMC, use

\[
\ddot{k} = |\gamma| + \eta + \Delta \gamma_m
\]  

(89)

where $\gamma$ is numerically available from (77). $\Delta \gamma_m$ is the deviation factor numerically computable using the algorithm of Section VI.

b. For 2-SMC, use:

\[
\Gamma_M = \max \left[ \ddot{b} (t, x, u) + \Delta \ddot{b}_m \right]
\]  

\[
\Gamma_m = \min \left[ \ddot{b} (t, x, u) - \Delta \ddot{b}_l \right]
\]  

\[
\Phi = \max [|\ddot{a} (t, x, u)|] + \Delta \alpha_m
\]  

(90)

where $\ddot{a}$ and $\ddot{b}$ are numerically available using (87) and (88). $\Delta \ddot{b}_m, \Delta \ddot{b}_l$ and $\Delta \alpha_m$ are the deviation factors numerically computable using the algorithm of Section VI.

**VIII. SIMULATION RESULTS**

We simulate all the above-presented controllers while investigating the responses of the two outputs of the boiler. The input of steam ‘$\ddot{m}_i$’ is used as the main disturbance agent in the boiler system as any variation in the steam can cause all the unstable boiler dynamics to go unbound. We change the steam flow in the form of a ramp at $t = 100s$. All control implementations are individually discussed in subsequent subsections. In order to create uncertainties, we randomly change the parameters of the boiler by $\pm 10\%$. For simulation, we run two models of boiler simultaneously; one represents the actual system whose parameters are changing by $\pm 10\%$, the other represents the nominal model with constant parameters whose equations are used to determine SMC equivalent control law. The aim is to stabilize pressure at 8500 and level at 1. The values of all boiler parameters are adopted from [33].

A brief overview of how the controllers are set is presented hereafter.

**A. 1-SMC FOR PRESSURE CONTROL**

We have first used 1-SMC to control one output to observe how the proposed approach is working. In this regard, we take a sliding manifold of (22) to control pressure using the input of the heating rate ‘$Q$’. The level is controlled by a nominal control (PID) via an input of differential feedwater rate ‘$\ddot{m}_d$’.

Figure 3 and Figure 4 shows the simulation results with
B. USING INTEGRAL SLIDING MANIFOLD FOR 1-SMC

The 1-SMC approach can be upgraded to sliding manifolds of various forms. In order to tackle out the disturbances, different operating points, \( (P_d = 8500, L_d = 1.0) \) and \( (P_d = 8000, L_d = 1.5) \) respectively. The deviation factor is found to be \( \Delta \gamma_m = 1.4549 \).

\[ \text{FIGURE 4. Controlling pressure with 1-SMC at a different operating point.} \]

\[ \text{FIGURE 5. Controlling Pressure using Integral SMC.} \]

Integral SMC (ISM) is a widely recognized SMC formulation, in which the order of sliding variable is increased by incorporating an integral function of error. In this respect, the ISMC manifold takes the following form:

\[ s = \tilde{P} + \lambda \int_0^t \tilde{P}(z) \, dz \tag{91} \]

Using this form of sliding variable ‘s’, the mathematics of the numerical design of SMC is not affected and hence the same program is re-simulated with the upgraded sliding manifold. Figure 5 shows the simulation response under the same operating conditions, where the following parameters were considered in order to design the manifold:

\[ \eta = 10, \quad \lambda = 0.1 \]

The simulation again proves that the numerical approach of designing integral SMC is working perfectly while stabilizing all the concerned variables at respective setpoints without violating safety constraints.

C. 1-SMC FOR LEVEL CONTROL

In the next phase, we switch the control loops. We use SMC to control the output level while using a nominal PID for pressure control. For level control, we test another formulation of SMC, which is described in Khalil’s text [39].

\[ u_{df} = \mu(x) \text{sign}(s_L) \tag{92} \]

where, \( \mu(x) = \mu_0 + \rho(x) + \Delta \rho \)

Where \( \Delta \rho \) is the same deviation factor that is determined as \( \Delta \rho = 5.8059 \) using the algorithm of Section VI. \( \mu_0 \) is arbitrary and has the same role as \( \eta \), i.e., to determine the...
convergence time of error trajectories towards the sliding manifold. The parameter $\rho(x)$ is derived from (44) such that the following is satisfied.

$$
\rho(x) \geq \left| \left( G\hat{C}_{VL}E^{-1}B_{DF} \right)^{-1} \left( G\hat{C}_{VL}E^{-1}Ax + G\hat{C}_{VL}E^{-1}B_{SQ}u_{SQ} \right) \right| (93)
$$

This control formulation also works accurately. Figure 6 shows the simulation result where the output level was controlled at the desired setpoint of $L_d = 1$.

**D. COMBINED 1-SMC FOR PRESSURE AND LEVEL**

We implement MIMO control using sliding manifold as a linear combination of output errors. The sliding gains in multi-variable manifold should be selected such that the resultant sliding dynamics remain stable in the sliding phase. This is a challenging task for a complex nonlinear model, where the model is not available in normal or canonical form. Hence, the only way is to randomly select the gains of the manifold and simulate the results to see the stability of manifold for the selected gains.

In MIMO SMC formulation, the output pressure is controlled by heating rate $Q$ in one SMC, and drum level is controlled by differential feedwater rate $\dot{m}_{df}$ using another SMC. To realize this, a 2-dimensional sliding surface is formulated.

$$
G = \begin{bmatrix}
    g_{P1} & g_{P2} \\
g_{L1} & g_{L2}
\end{bmatrix} \quad (94)
$$

$$
\hat{s} = \begin{bmatrix}
    \hat{P} \\
    \hat{L}
\end{bmatrix}
$$

Here we point out that only 2 outputs have been used for the boiler in the manifold design, where a more generalized approach would be to use all the states. This is not possible for the boiler case because, due to the nonlinear nature of the model, the $x'_d$s (the desired setpoints of all the individual states corresponding to output setpoints) need to be determined for every simulation instant, which will be analytically impossible and even computationally complex using the numerical approach.

First, we investigate how the two SMCs are independently designed without any coupling between the controls. This can be done by enforcing the de-coupling condition as $g_{P2} = g_{L1} = 0$. Figure 7 shows the simulation results.

In the next phase, we relax the decoupling condition by choosing $g'_{ij}s = 1, \forall i, j$ which turns out to be a stable manifold. Figure 8 shows the simulation result of this approach. The offset in level output can be trivially neutralized using an integral state $x_{Ld} = \int_0^t \hat{L}(s) \, ds$.

**E. ANALYSIS OF $s$ AND $\hat{s}$ NUMERICAL COMPUTATION**

As a validation step, we test a simple model of second-order in order to analyze how the numerical computations of $s$ and $\hat{s}$ compare with their respective analytical computations. We first derive the expressions of $s$ and $\hat{s}$ analytically, and then validate the accuracy of the numerical approach for the equations (85) and (72). The model selected is from the text [40], which is as
follows.

\[
\dot{x}_1 = x_2 \\
\dot{x}_2(t) = -25x_1 + 133u(t) + 10\sin(\pi t)
\] (96)

The following SMC control of the example 1.3.2 from [40] has been used to track \( r = \sin(t) \).

\[
u = \frac{1}{b}(\epsilon \text{sign}(s) + ks + c(\dot{r} - \dot{x} + \ddot{r} - 25\dot{x} - d_c))
\] (97)

\[
c = 15, \epsilon = 0.5, k = 10
\]

\[
d_c = -(10) \text{sign}(s)
\] (98)

Let’s define the output variable as a nonlinear function of states.

\[
y = x_1 \sin(x_1) + x_2^3 - x_1
\] (99)

We use a sliding variable as a function of output:

\[
s = g(y - y_d)
\] (100)

Taking first and second derivatives while considering \( y_d \) to be constant.

\[
\dot{s} = g\dot{y} = (\sin(x_1) + x_1 \cos(x_1) - 1)\dot{x}_1 + 3x_2^2\dot{x}_2
\]

\[
= (\sin(x_1) + x_1 \cos(x_1) - 1)x_2
\]

\[
+ 3x_2^2(-25x_1 + 133u(t)) + 10\sin(\pi t))
\] (101)

\[
\rightarrow \dot{s} = (\cos(x_1) + \cos(x_1) - x_1 \sin(x_1))x_2
\]

\[
+ 3x_2^2(-25)\dot{x}_1 + ((\sin(x_1) + x_1 \cos(x_1) - 1)
\] (102)

\[
\rightarrow \ddot{s} = ((\cos(x_1) + \cos(x_1) - x_1 \sin(x_1))x_2
\]

\[
+ 3x_2^2(-25)\dot{x}_1 + ((\sin(x_1) + x_1 \cos(x_1) - 1)
\] (103)

The \( \dot{x}_1 \) and \( \dot{x}_2 \) can be substituted from (96). However, we only use the above equations to plot \( \dot{s} \) and \( \ddot{s} \). Equations (102) and (103) are the analytical expressions of \( \dot{s} \) and \( \ddot{s} \). We use these expressions to plot \( \dot{s} \) and \( \ddot{s} \) and compare with the numerically computed \( \dot{s} \) and \( \ddot{s} \) from equations (85) and (72). Figure 9 shows the plots where the two curves of both graphs almost coincide hence proving the accuracy of the numerical approach.

**F. SECOND ORDER SLIDING MODE CONTROL**

We simulate Super Twisting SMC as presented in Section V using the proposed numerical approach. Only the output of
TABLE 1. Variable Symbols and Units.

| Symbol | Variable name                               | Units       |
|--------|---------------------------------------------|-------------|
| $\bar{a}_v$ | Average steam volume fraction              | (dimensionless) |
| $\alpha_r$ | Steam quality at the riser outlet          | (dimensionless) |
| $\rho_w$ | Density of feedwater                       | kg/m$^3$   |
| $\rho_g$ | Density of saturated steam                 | kg/m$^3$   |
| $L$    | Drum level                                 | m           |
| $L_d$  | Desired drum level                         | m           |
| $e_L$  | Error b/w actual and desired drum level    | m           |
| $P$    | Drum pressure                              | kPa         |
| $P_d$  | Desired drum pressure                      | kPa         |
| $V_{dc}$ | Downcomer volume                          | m$^3$       |
| $p$    | Drum pressure                              | kPa         |
| $A_d$  | Drum surface area                          | m$^2$       |
| $V_d$  | Drum volume                                | m$^3$       |
| $\eta$ | Efficiency                                 | (dimension less) |
| $h_w$  | Enthalpy of feedwater                      | kJ/kg       |
| $\dot{m}_{fw}$ | Feedwater flow rate                      | kg/s       |
| $\dot{m}_{df}$ | Differential feedwater flow                | kg/s       |
| $q_{fs}$ | Feedwater steady-state value               | kg/s       |
| $\dot{m}_s$ | Steam flow rate                          | kg/s       |
| $g$    | Gravity                                    | m/s$^2$     |
| $Q$    | Heating rate                               | MW          |
| $h_{fg}$ | Latent heat of vaporization of water      | kJ/kg       |
| $l$    | Level of the water                         | m           |
| $m_{md}$ | Mass of drum                              | kg          |
| $\tau_p$ | Metal temperature                         |              |
| $\dot{m}_{dc}$ | Riser & downcomer mass flow rate        | kg/s       |

Table 1 (Continued.)

| Symbol | Variable name                               | Units       |
|--------|---------------------------------------------|-------------|
| $m_r$  | Riser metal mass                           | kg          |
| $V_r$  | Riser volume                               | m$^3$       |
| $\dot{m}_r$ | Risers flow rate                        | kg/s       |
| $h_f$  | Specific enthalpy of saturated liquid water|kJ/kg       |
| $h_g$  | Specific enthalpy of saturated liquid water|kJ/kg       |
| $C_p$  | Specific heat of metal                     | J/(K kg)    |
| $\dot{m}_{ct}$ | Steam condensation rate                    | kg/s       |
| $\dot{m}_s$ | Steam flow rate                           | kg/s       |
| $\dot{m}_{sd}$ | Steam flow rate through liquid surface of drum | kg/s |
| $\alpha_{st}$ | Steam quality (dimension less)         |             |
| $T_{FG}$ | Temperature of flue gas                   | °C          |
| $m_{mt}$ | Total mass of metal tube and drum tube      | kg          |
| $m_{st}$ | Total metal mass                           | kg          |
| $V_{st}$ | Total volume of steam in the system       | m$^3$       |
| $V_t$  | Total volume of the drum, downcomer, and risers; | m$^3$ |
| $V_{wt}$ | Total volume of water                      | m$^3$       |
| $V_{sd}$ | Volume of steam in drum under liquid level | m$^3$   |
| $\nu_{sd}^a$ | Volume of steam in the drum in the hypothetical situation, when there is no condensation of steam in the drum | m$^3$ |
| $q_{dc}$ | Total mass flow rate into the risers      | kg/s       |
| $q_{ct}$ | Condensation flow of steam                 | kg/s       |
| $k$    | Friction coefficient in (dimensionless)    |             |
| $T_d$  | Residence time                             | s           |

To satisfy $\lambda^2 \geq \frac{4\Phi^2}{\Gamma_m(W+\Phi)}$, we select $\lambda = \sqrt{\frac{4\Phi^2}{\Gamma_m(W+\Phi)}} + 1$.

To satisfy $0 < \rho \leq 0.5$, we select $\rho = 0.5$. Deviation factors are set as $\Delta \beta_t = 6.1 \times 10^{-6}$, $\Delta \beta_m = 8.7 \times 10^{-6}$, and $\Delta \alpha_m = 1.045 \times 10^{-5}$. Figure 10 shows the simulation results. It is evident that the Super Twisting SMC response...
shows a diminished chattering as compared to all other forms of SMCs.

G. BRIEF DISCUSSION
The simulations (Figure 3–Figure 10) show that the introduction of change in steam dynamically affects all the variables; however, they remain within safety limits without violating any physical constraint. Also, it is evident that the SMC successfully stabilizes both outputs, the pressure, and the level, which after undergoing temporary dynamics, maintain their respective setpoints. All the simulations show that the inputs are able to regulate the concerned outputs at their respective setpoints. All the simulations show a diminished chattering as compared to all other forms of SMCs.

The further gradient of the above vector \( \hat{C}_{V_y} \), i.e. \( \frac{\partial (\hat{C}_{V_y})}{\partial x} \) generates a matrix.

\[
\frac{\partial (\hat{C}_{V_y})}{\partial x} = \begin{bmatrix}
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_1^2} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_1 \partial x_n} \\
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_2 \partial x_1} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_2^2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_n \partial x_1} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_n^2}
\end{bmatrix}_{n \times n}
\]

APPENDIX

A. NOMENCLATURE OF SYMBOLS AND VARIABLES
See Table 1.

B. DOUBLE DERIVATIVE OF FUNCTION \( h(x, t) \)
Consider a case in which the aim is to numerically compute the gradient of a vector \( \hat{C}_{V_y} \).

\[
G_x = \frac{\partial (G \hat{C}_{V_y})}{\partial x} = G \frac{\partial (\hat{C}_{V_y})}{\partial x}
\]

where \( G \) is a constant matrix. \( \hat{C}_{V_y} \) itself is regarded as the gradient of \( h \) along the state vector \( x \).

\[
\hat{C}_{V_y} = \frac{\partial h}{\partial x} = \Delta h = \begin{bmatrix}
\Delta h & \Delta h & \Delta h & \cdots & \Delta h
\end{bmatrix}
\]

The further gradient of the above vector \( \hat{C}_{V_y} \), i.e. \( \frac{\partial (\hat{C}_{V_y})}{\partial x} \) generates a matrix.

\[
\frac{\partial (\hat{C}_{V_y})}{\partial x} = \begin{bmatrix}
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_1^2} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_1 \partial x_n} \\
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_2 \partial x_1} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_2^2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 \hat{C}_{V_y}}{\partial x_n \partial x_1} & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \hat{C}_{V_y}}{\partial x_n^2}
\end{bmatrix}_{n \times n}
\]

\[
\frac{\partial^2 f}{\partial x_j \partial x_k}(a, b) \equiv \frac{\partial f}{\partial x_j}(a + \Delta a, b) - \frac{\partial f}{\partial x_j}(a - \Delta a, b) \\
\frac{\partial f}{\partial x_j}(a + \Delta a, b) \equiv \frac{f(a + \Delta a, b + \Delta b) - f(a + \Delta a, b - \Delta b)}{2\Delta b} \\
\frac{\partial f}{\partial x_j}(a - \Delta a, b) \equiv \frac{f(a - \Delta a, b + \Delta b) - f(a - \Delta a, b - \Delta b)}{2\Delta b} \\
\frac{\partial^2 f}{\partial x_j \partial x_k}(a, b) \equiv \frac{f(a + \Delta a, b + \Delta b) - f(a + \Delta a, b - \Delta b) - f(a - \Delta a, b + \Delta b) + f(a - \Delta a, b - \Delta b)}{4\Delta a \Delta b}
\]
which is evaluated using the Ab-initio algorithm by using
minute increments to numerically compute the individual el-
ments of the above matrix. The double derivatives of elements
are computed as follows.
Consider an \((i, j)\) element of the above matrix.
\[
\frac{\partial^2 f}{\partial x_i \partial x_j}(a, b)
\]
The point \((a, b)\) represents the current value of \((x_i, x_j)\) at
a certain instant. To compute the double derivative of \(f\), we
apply very small increments, \(\Delta a \to 0, \Delta b \to 0\), in \((x_i, x_j)\)
while keeping the rest of the states constant, shown at the
bottom of the page.
In the case of \((i = j)\), i.e., for the diagonal elements, the
above gets simplified.
\[
\frac{\partial^2 f}{\partial x_i^2}(a, b) \approx f(a - h) - 2f(a) + f(a + h)
\]
where \(h \to 0\), and is assigned a very small numerical value
in the simulation. The rest of the states are kept constants.
Using the above two equations, all the elements of
\(\frac{\partial^2 \mathbf{Cy}}{\partial t^2}\)
are numerically computed and used thereafter.

REFERENCES
[1] V. I. Utkin, Sliding Modes in Control and Optimization, 1st ed. Berlin,
Germany: Springer, 1992.
[2] J. Huang, J. Yang, D. Xie, and D. Wu, “Optimal sliding mode chaos
control of direct-drive wave power converter,” IEEE Access, vol. 7,
p. 90922–90930, 2019.
[3] L. Zhang, Y. Wang, Y. Hou, and H. Li, “Fixed-time sliding mode
control for uncertain robot manipulators,” IEEE Access, vol. 7,
p. 149750–149763, 2019.
[4] W. Perruquetti and J. P. Barbot, Sliding Mode Control in Engineering,
Boca Raton, FL, USA: CRC Press, 2002.
[5] G. Bartolini, A. Pisano, E. Punta, and E. Usai, “A survey of applications
of second-order sliding mode control to mechanical systems,” Int. J. Control,
vol. 76, nos. 9–10, pp. 875–892, Jan. 2003.
[6] A. Levant, “Principles of 2-sliding mode control,” Automatica, vol. 43,
n. 4, pp. 576–586, Apr. 2007.
[7] Y. Kafi, M. Saad, K. Benjelloun, and A. Fatemi, “Discrete-time second
order sliding mode with time delay control for uncertain robot manipula-
tors,” Robot. Auton. Syst., vol. 94, pp. 53–60, Aug. 2017.
[8] S. Li and J. Fei, “Adaptive second-order sliding mode fuzzy control based
on linear feedback strategy for three-phase active power filter,” IEEE
Access, vol. 6, pp. 72992–73000, 2018.
[9] Z. Feng and J. Fei, “Super-twisting sliding mode control for micro
gyroscopic based on RBF neural network,” IEEE Access, vol. 6,
p. 64993–65001, 2018.
[10] M. Rahmani, H. Komijani, A. Ghanbari, and M. M. Ettedgh, “Optimal
novel super-twisting PID sliding mode control of a MEMS gyroscope
based on multi-objective bat algorithm,” Microsyst. Technol., vol. 24,
n. 6, pp. 2835–2846, Jun. 2018.
[11] M. Horch, A. Boumediene, and L. Baghli, “Backstepping approach for
nonlinear super twisting sliding mode control of an induction motor;” in
Proc. 3rd Int. Conf. Control, Eng. Inf. Technol. (CEIT), May 2015, pp. 1–6.
[12] C. P. Vo, X. D. To, and K. K. Auh, “A novel adaptive gain integral terminal
sliding mode control scheme of a pneumatic artificial muscle system with
time-delay estimation,” IEEE Access, vol. 7, pp. 141133–141143, 2019.
[13] L. Tao, Q. Chen, Y. Nan, and C. Wu, “Double hyperbolic reaching
law with chattering-free and fast convergence,” IEEE Access, vol. 6,
p. 27717–27725, 2018.
[14] A. Bartoszewicz and P. Lesniewski, “New switching and nonswitching
type reaching laws for SMC of discrete time systems;” IEEE Trans.
Control Syst. Technol., vol. 24, no. 2, pp. 670–677, Mar. 2016.
[15] H. Ma and Y. Li, “Multi-power reaching law based discrete-time sliding-
mode control,” IEEE Access, vol. 7, pp. 49822–49829, 2019.
AHMED REHAN received the B.S. degree in electrical engineering from the National University of Sciences and Technology (NUST), Pakistan, and the M.S. degree in systems and control engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He is currently pursuing the Ph.D. degree in robotics engineering with the Khalifa University of Science and Technology (KUSTAR), Abu Dhabi, United Arab Emirates. His research interests include process modeling, advanced control, and optimization.

AHMED AL-DURRA (Senior Member, IEEE) received the Ph.D. degree in ECE from The Ohio State University, in 2010. He is currently a Professor with the EECS Department Khalifa University of Science and Technology, United Arab Emirates. He is leading the Energy Systems Control and Optimization Laboratory under the Advanced Power and Energy Center. His research interests include the applications of control and estimation theory on power systems stability, micro and smart grids, renewable energy systems and integration, and process control. He has one U.S. patent, one edited book, 12 book chapters, and over 200 scientific papers in top-tier journals and refereed international conference proceedings. He has supervised/co-supervised over 25 Ph.D./master’s students. He is an Editor of the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY and the IEEE POWER ENGINEERING LETTERS and an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS and Frontiers in Energy Research.