International Journal of Control, Automation, and Systems VV(X) (YYYY) 1-10
http://dx.doi.org/10.1007/s12555-xxx-xxxx-x
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Supervised Learning in Model Reference Adaptive Sliding Mode Control

Omar Makke* and **Feng Lin**

Abstract: The well known back-propagation algorithm has revolutionized machine learning and artificial intelligence, particularly in neural network applications. Although gradient descent-based algorithms are utilized in control applications, they are not as prevalent as in neural network applications. This discrepancy can be attributed to the successful development of various adaptation laws which ensure system stability while meeting the required design criteria. Many of these laws can be found in model reference adaptive control (MRAC) and adaptive sliding mode control (ASMC). This paper investigates the applicability of the Brandt-Lin (B-L) learning algorithm, mathematically equivalent to the back-propagation algorithm, in adaptive control applications. We find that combining the B-L learning algorithm with SMC yields a robust controller suitable for model reference adaptive sliding mode control (MRA-SMC). The controller is applicable to linear and a class of nonlinear dynamic systems and is suitable for efficient implementation. We derive the stability criteria for this controller and conduct simulations to study the adaptation's impact on chattering. Our work exemplifies one approach to adopt the back-propagation algorithm in control applications.

Keywords: Adaptive Control, Online Learning Algorithms, Model Reference Adaptive Control (MRAC), adaptive sliding mode control (ASMC), Chattering Reduction, Back-Propagation Algorithm

1. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) are increasingly being adopted in various scientific domains and applications. Specifically, neural networks have been gaining popularity and fame even among non-technical audience due to Generative AI. Central to this increase in popularity is the well known back-propagation algorithm [1]. Supervised deep learning is one of the most widely used methods in neural networks [2] [3]. Although various learning algorithms[1] have been proposed for neural networks, the back-propagation algorithm is probably the most well known and widely used [4] [5] in supervised learning. The back-propagation algorithm gained popularity due to its ease of use, availability of fast computers, and software which performs automatic differentiation such as Autograd [6], which made training large and complex neural networks easier. In comparison, in control applications, the application of the back-propagation algorithm is not trivial. First, the errors have to be back-propagated through a dynamic system, and second, the system stability, including parameters' convergence, must be guaranteed. Over the years, various adaptation algorithms have been successfully developed for wide range of applications in model reference adaptive control (MRAC) and adaptive sliding mode control (ASMC).

The aim of this paper is to (1) investigate the applicability of the back-propagation algorithm in adaptive control due to its large success in supervised learning applications, and (2) design an adaptive controller that is based on the back-propagation algorithm. It is known that sliding mode control (SMC) can result in chattering which can be harmful to physical components if not addressed [7]. It is also known that MRAC requires the controlled plant to match the reference model [8]. We find that combining the back-propagation adaptation algorithm with sliding mode control is applicable to model reference adaptive sliding mode control (MRA-SMC), which benefits from the robustness of SMC and model matching from MRAC. An adaptive controller can match a class of nonlinear plants to a linear reference model where the dynamics of both systems can differ. The controller parameters "learn" the values of the plant's coefficients which is also beneficial in diagnostics and prognostics applications [9]. Furthermore, the parameter adaptation reduces the chattering phenomenon. The controller can be implemented in analog fashion since no dedicated backwards step is required.

The paper is organized as follows. In Section 2, we review recent and relevant work in the literature. In Section 3, we give a brief review of the Brandt-Lin (B-L) learning algorithm for neural networks, chosen for the back-

[1] "Learning" and "adaptation" will be used interchangeably. The word "algorithm" is used here in a generalized sense to mean a model or a mathematical description for updating weights/parameters of neural networks or dynamic systems.

This work is supported in part by the National Science Foundation of USA under grant 2146615.
Omar Makke and Feng Lin are with the Department of Eletrical and Computer Engineering at Wayne State University, Detroit, MI, 48202, USA (emails: omarmakke, flin@wayne.edu)
* Corresponding author.

propagation algorithm implementation. Section 4 extends B-L learning algorithm to general dynamic systems so that it can be used in adaptive control. Section 5 applies the extended B-L algorithm to adaptive PID control and discusses the rationale behind combining sliding mode control with the B-L learning algorithm for adaptive control. In Section 6, the MRA-SMC reaching law is derived using Lyapunov method. Stability of the system and convergence of the adaptation are also proved. Section 7 presents Simulink simulations to demonstrate the controller's effectiveness using both linear and nonlinear plants. Finally, Section 8 concludes the paper.

2. BACKGROUND

In this section we review some of the advancements in learning and adaptation in relation to our work. In [10], Adaptive Neural Networks PID controller structure is proposed for control robot manipulators. Neural networks learn the Fourier series expansion of the signal and the plant. This allows the errors to be back-propagated through the learnt plant at the cost of added computational complexity. In [11] Bayesian optimization is proposed to optimize PID controllers for aircraft maneuvering control. In [12] an adaptive swarm learning process is applied to auto-tune a PID controller. This learning algorithm is based on gradient descent and it updates the weights in discrete fashion. In [13], recursive least square method is used to adapt a PID for DC motor control. In [14], a PID with switching action controller is provided, although this method is not adaptive. It demonstrates the advantages of combining a PID controller with sliding mode control. Other fuzzy adaptive tuning methods are proposed to adapt PID controllers [15] [16]. In [7] [17], non adaptive methods are proposed to reduce chattering. In relation to adaptive sliding mode control (ASMC), several recent methods have been proposed. In [18] [19] [20], the proposed adaptive methods reduce chattering due to the adaptation of parameters. These methods, however, do not focus on model reference control. In [21], a new adaptive method is proposed but it is not based on back-propagation. In [22], the method is direct MRA for Single Input Single Output Systems (SISO) as our work, but is applicable only to linear systems.

In [23] [24] [25] [26], Brandt and Lin developed a learning algorithm (abbreviated as B-L algorithm below) that is mathematically equivalent to the back-propagation algorithm in neural networks. The algorithm does not require a dedicated feedback step for error back-propagation, which makes it suitable for online learning and for analog control applications.

In comparison to the reviewed work, the intention of our work is to show how the back-propagation algorithm (using the B-L learning algorithm) can be utilized in adaptive control. We focus on model reference control and aim to "learn" the plant's coefficients.

3. B-L LEARNING ALGORITHM

Because the new learning algorithm to be proposed is an extension of the B-L learning algorithm from neural networks to general systems, let us briefly review the B-L algorithm.

To describe a neural network (either hierarchical or non-hierarchical), we enumerate all neurons in a neural network as $\mathcal{N} = \{1, 2, ..., N\}$. We do not put any restrictions on connections among neurons. The weights of the connection from the i-th neuron to the j-th neuron is denoted by w_{ij}. The set of all connections is denoted by

$$\Psi = \{w_{ij} : i, j \in \mathcal{N} \wedge i \text{ is connected to } j\}.$$

Not all neurons have preceding neurons. If a neuron does not have preceding neutrons, then we consider it as an input neuron. The set of input neurons is denoted by

$$\mathcal{I} = \{n \in \mathcal{N} : (\forall j \in \mathcal{N}) w_{jn} \notin \Psi\}.$$

The firing rates of input neurons r_n, $n \in \mathcal{I}$, are considered as the inputs to the neural network.

The dynamics of non-input neuron $n \in \mathcal{N} - \mathcal{I}$ are described by its membrane potential p_n and firing rate r_n, given by

$$p_n = \sum_{w_{mn} \in \Psi} w_{mn}\, r_m, \quad r_n = \sigma(p_n),$$

where $\sigma(p_n) = 1/(1 + e^{-p_n})$ is the sigmoidal function.

The weights w_{ij} can be adapted to minimize the following least square error

$$E = \frac{1}{2} \sum_{m \in \mathcal{O}} (r_m - \tilde{r}_m)^2,$$

where $\mathcal{O}$ is the set of output neurons and $\tilde{r}_m$ is the desired/target firing rate of the output neuron $m \in \mathcal{O}$.

The following learning algorithm is proposed by Brandt and Lin in [23, 26] to adapt the weights $w_{ij} \in \Psi$.

$$\dot{w}_{ij} = \sigma'(p_j) \frac{r_i}{r_j} (-\gamma r_j (r_j - \tilde{r}_j) + \sum_{w_{jm} \in \Psi} w_{jm}\, \dot{w}_{jm}), \tag{1}$$

where $\sigma'(p_j)$ is the derivative of $\sigma(p_j)$.

It is proved in [23, 26] that the following is true for the B-L algorithm of Equation (1).

$$\dot{w}_{ij} = -\gamma \frac{dE}{dw_{ij}}, \tag{2}$$

where γ is the adaptation/learning rate, which is a design parameter. The above equation shows that the gradient-decent-based learning is achieved. Note that the significance of the B-L algorithm is that the adaptation of the weights is described as a function of time, which makes it suitable for on-line learning.

The B-L algorithm is mathematically equivalent to the back-propagation algorithm for neural networks, but has several advantages over the back-propagation algorithm that allows it to be generalized to other systems [23, 26, 27]. In the next section, we extend the B-L algorithm to general dynamic systems.

4. SUPERVISED LEARNING IN DYNAMIC SYSTEMS

We extend the B-L algorithm to general dynamic systems by replacing neurons in a neuron network by subsystems described by either an algebraic equation or a differential equation. We model a general dynamic system by a generalized signal-flow graph (GSFG), which has all elements of a conventional signal-flow graph (CSFG) [28]. In addition, some nodes in GSFG are super nodes as to be discussed below.

Assume that a GSFG has N nodes. Denote a node by

$$n \in \mathcal{N} = \{1, 2, ..., N\}.$$

Denote the branch (if exists) and its gain from node i to node j by ω_{ij}. The set of branches/gains is denoted by

$$\Omega = \{\omega_{ij} : i, j \in \mathcal{N} \wedge \text{ node } i \text{ is connected to node } j\}.$$

The set Ω is partitioned into two sets:

$$\Omega = \Omega_a \cup \Omega_{na},$$

where Ω_a is the set of adaptable branches/gains and Ω_{na} is the set of non-adaptable branches/gains. Non-adaptable branches have gains which are constants, that is,

$$\omega_{ij} \in \Omega_{na} \Leftrightarrow \omega_{ij} = \bar{\omega}_{ij},$$

where $\bar{\omega}_{ij}$ are constants.

As mentioned above, some nodes in $\mathcal{N}$ are super nodes. A super node consists of a pair of input and output, denoted by

$$(u_n, y_n),$$

where u_n is the input to node n and y_n is the output from node n. Let $U = \{u_n : \mathcal{R} \to \mathcal{R}\}$ be a set of all inputs to a super node n. The relationship between u_n and y_n is described by

$$y_n = \mathcal{G}_n[u_n]. \tag{3}$$

where $\mathcal{G}_n$[2] is a functional which maps every input function of time to an output function of time. If the super node is linear and time-invariant, then $\mathcal{G}_n$ is the convolution of the input with the impulse response of the super node. We assume that the Fréchet derivative of $\mathcal{G}_n$, denoted by $\mathcal{G}'_n$, exists[3].

If a node $n \in \mathcal{N}$ is not a super node, then $y_n = u_n$, that is, $\mathcal{G}_n$ is an identity mapping: $\mathcal{G}_n[u_n(t)] = u_n(t)$.

As in CSFG, the input signal of node n is the sum of all signals flowing to n:

$$u_n = \sum_{m=1}^{N} \omega_{mn}\, y_m. \tag{4}$$

Our goal is to use on-line learning to learn/adapt the gains $\omega_{ij} \in \Omega_a$ so that some error is minimized. We assume that the error is a function of outputs:

$$E = E(y_1, y_2, ..., y_N).$$

Theorem 1: Consider an adaptive system described by a generalized signal-flow graph with nodes $n \in \mathcal{N}$ and branches $\omega_{ij} \in \Omega$. Using the following new learning algorithm

$$\begin{aligned} \dot{\omega}_{ij} = \mathcal{G}'_j[u_j]\frac{y_i}{y_j}(-\gamma y_j \frac{\partial E}{\partial y_j} \\ + \sum_{\omega_{jm}\in\Omega_a} \omega_{jm}\, \dot{\omega}_{jm} + \sum_{\omega_{jm}\in\Omega_{na}} \bar{\omega}_{jm}\, \dot{\omega}_{jm}), \end{aligned} \tag{5}$$

where γ is the adaptation/learning rate, the gradient-decent-based on-line learning is achieved as

$$\dot{\omega}_{ij} = -\gamma \frac{dE}{d\omega_{ij}}. \tag{6}$$

Proof

The proof can be found in [27]. In comparison with equation (1), equation (5) replaces σ' with the Fréchet derivative $\mathcal{G}'$, and considers branches which have fixed (non-adaptable) parameters $\bar{\omega}_{jm}$.

Note that if node j is an output node, Equation (5) reduces to

$$\dot{\omega}_{ij} = -\gamma y_i \mathcal{G}'_j[u_j]\frac{\partial E}{\partial y_j}. \tag{7}$$

5. ADAPTIVE CONTROL BY SUPERVISED LEARNING

In this section, we investigate adaptive control using the B-L algorithm from the previous section. We consider the PID control of Figure 1, where the gains K_p, K_i, and K_d are adapted to minimize the square of error $e = y - \tilde{y}$, that is,

$$E = \frac{1}{2}e^2 = \frac{1}{2}(y - \tilde{y})^2.$$

[2] $\mathcal{G}_n$ can be viewed as the model of a single-input-single-output system starting at $-\infty$.

[3] The Fréchet derivative [29] of $\mathcal{G}_n$ is defined as a functional such that

$$\lim_{||\varepsilon||\to 0} \frac{||\mathcal{G}_n[u+\varepsilon] - \mathcal{G}_n[u] - \mathcal{G}'_n[u]\varepsilon||}{||\varepsilon||} = 0.$$

Using Theorem 1 and Equation (7), we have

$$\dot{K}_p = -\gamma z_p \mathcal{G}'_j \frac{\partial E}{\partial y} = -\gamma z_p \mathcal{G}'_j e.$$

Approximating the Fréchet derivative $\mathcal{G}'$ by a constant (see [27]), absorbing it into γ, and doing the same for K_i, and K_d, we obtain the following adaptation law

$$\dot{K}_p = -\gamma z_p e \qquad \dot{K}_i = -\gamma z_i e \qquad \dot{K}_d = -\gamma z_d e \qquad (8)$$

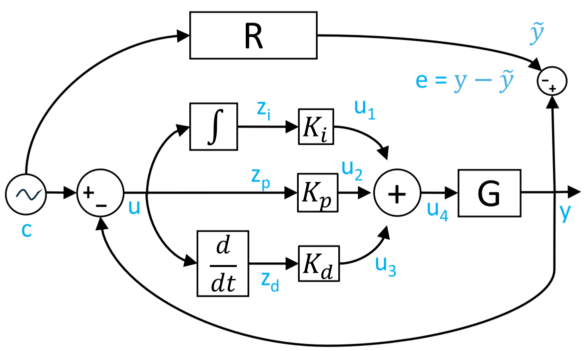


Fig. 1. Model reference adaptive PID control.

A simulation study is conducted to test the adapted control for a linear stable plant with the following transfer function

$$G(s) = \frac{5}{s^2 + 2.4s + 2.25}.$$

The reference model is given by

$$R(s) = \frac{25s^2 + 50s + 100}{s^3 + 27.4s^2 + 52.25s + 100}.$$

A perfect match of the controlled system with the reference model is possible when

$$K_p = 10, \quad K_i = 20, \quad K_d = 5 \qquad (9)$$

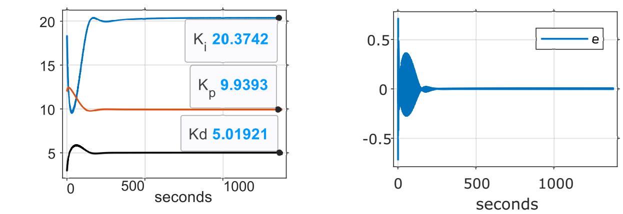
In the simulation, the initial gains are given by

$$K_p = 12, \quad K_i = 18, \quad K_d = 3 \qquad (10)$$

To provide adequate excitation [30] for gains to converge, we first use input $c(t) = sin(2\pi t) + sin(\pi t)$. The results are shown in Figure 2, where the error goes to 0 and K_p, K_i, K_d converge to the true values.

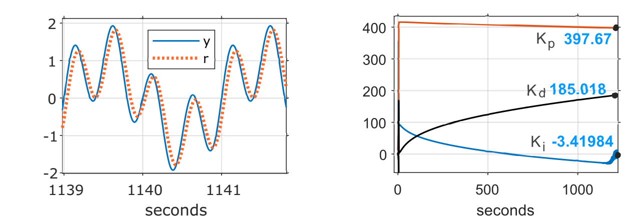
However, if the frequency of the input is increased to $c(t) = sin(4\pi t) + sin(\pi t)$, the error does not go to 0. Figure 3a shows a phase shift between y and $\tilde{y}$. This leads to a constant error depending on the initial values of the gains and causes the gains to not converge to a final value as shown in Figure 3b.

Intuitively, adaptation using $e(t)$ at time t is slow with respect to the plant dynamics. To improve the adaptation, we consider the "predicted" error $e(t+h)$ for (small) $h >$



(a) Gains are adapting. (b) Error is minimized.

Fig. 2. The error e is minimized and the gains in Equation (10) approach those in Equation (9).



(a) y is leading r (b) Parameters keep drifting.

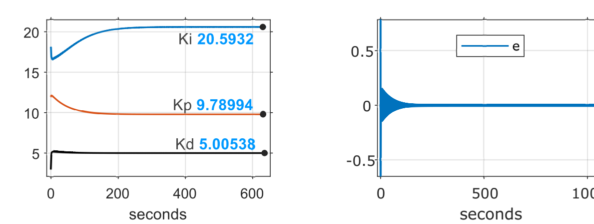
Fig. 3. Performance deteriorates at higher frequencies.

0 and approximate $e(t+h)$ by $q(t) = e(t) + h\dot{e}(t)$. The adaptation laws are then modified to

$$\dot{K}_p = -\gamma z_p q \qquad \dot{K}_i = -\gamma z_i q \qquad \dot{K}_d = -\gamma z_d q \qquad (11)$$

which can be obtained from Equation (7) with $E = \frac{1}{2}q^2$.

With these modifications, the gains converge again as shown in Figure 4. The performance is also better than the one shown in Figure 2. Using q, the error is now smaller by an order of magnitude. This suggests that combining sliding mode control with the B-L learning algorithm is beneficial, which will be done the following section.



(a) Parameters adapting. (b) Error is minimized.

Fig. 4. Parameters converge to their desired values when e is replaced with q.

One may notice that $e(t+h)$ can be expanded to include as many derivatives of $e(t)$ as needed to match the plant's order, that is, $(e(t+h))^2 = (e(t) + h\dot{e}(t) + \frac{h^2}{2}\ddot{e}(t) + \ldots)^2$. This is the motivation to combine sliding mode control with with the B-L learning algorithm.

6. THE PROPOSED MRA-SMC METHOD

Consider a second order dynamic system described by the following differential equation:

$$\ddot{y}+\sum_{i=1}^{I} a_i g_i(\dot{y},y) = u \tag{12}$$

Similarly, consider the following second order reference model driven by an input c.

$$\ddot{\tilde{y}}+\sum_{i=1}^{I} \tilde{a}_i r_i(\dot{\tilde{y}},\tilde{y}) = \sum_{j=1}^{J} b_j f_j(c,\dot{c}) \tag{13}$$

We assume that the reference model is stable, which is reasonable for any practical application. To simplify the notation, in the rest of the paper, we will write $\sum_{i=1}^{I}, \sum_{j=1}^{J}, g_i(\dot{y},y), r_i(\dot{\tilde{y}},\tilde{y})$ and $f_j(c,\dot{c})$ as $\sum_i, \sum_j, g_i, r_i$ and f_j, respectively, if appropriate. Note that our method can be extended to high-order systems, but with more complex notations. Note further that $f(.)$, $g(.)$, and $r(.)$ can differ, and can be nonlinear. We assume that they are continuous and bounded if their input is continuous and bounded. The goal of adaptive control is to adapt the gains ω_i and v_j so that the following error is minimized

$$E = \frac{1}{2}q^2 = \frac{1}{2}(\dot{e}+\lambda e)^2,$$

where $e = y - \tilde{y}$ and $q = \dot{e}+\lambda e$, and $\lambda > 0$. We use q instead of s to avoid confusing s with the complex variable s used in transfer functions. Define

$$\begin{aligned} x_i &= \omega_i + a_i - \tilde{a}_i \\ z_j &= v_j - b_j, \end{aligned} \tag{14}$$

and $\mathbf{x} = [x_1,\ldots,x_I]$ and $\mathbf{z} = [z_1,\ldots,z_J]$.

When $\mathbf{x} = \mathbf{z} = 0$, the adapted parameters reach their desired values and $e = 0$, $q = 0$.

By Theorem 1, the adaptation/learning is given by

$$\dot{\omega}_i = \gamma g_i q, \quad \dot{v}_j = -\gamma f_j q \tag{15}$$

We treat equation (15) as part of the system dynamics and design a model reference adaptive sliding mode control

$$u = \sum_j v_j f_j - \sum_i \omega_i g_i - \lambda \dot{e} - \sum_i \tilde{a} r_i + \sum_i \tilde{a} g_i + u_2 \tag{16}$$

where u_2 will be given later. The control architecture is shown in Figure 5. We prove that under this control, $q \to 0$, $\mathbf{x} \to \mathbf{x_f}$ and $\mathbf{z} \to \mathbf{z_f}$ as $t \to \infty$ for some constant values $\mathbf{x_f}$ and $\mathbf{z_f}$ using Lyapunov Theorem. Consider the following candidate Lyapunov function

$$V(q,\mathbf{x},\mathbf{z}) = \frac{1}{2}(q^2 + \sum_i x_i^2 + \sum_j z_j^2) \tag{17}$$

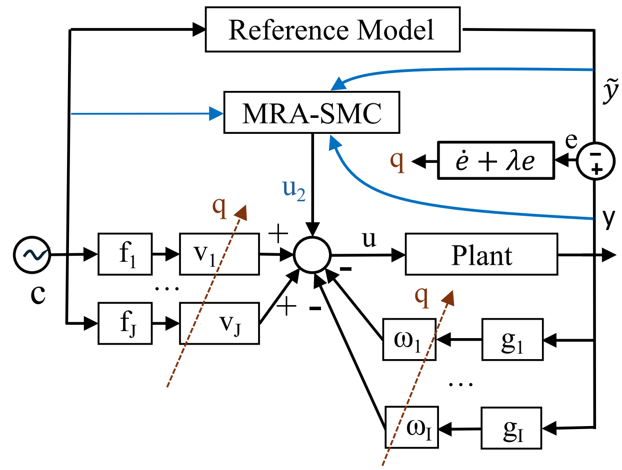


Fig. 5. Controller architecture. ω_1, ω_2, v_1 and v_2 adapt via the B-L Algorithm.

It is clear that

$$\begin{cases} V(q,\mathbf{x},\mathbf{z}) & = 0 \qquad q = 0 \wedge \mathbf{x} = \mathbf{0} \wedge \mathbf{z} = \mathbf{0} \\ V(q,\mathbf{x},\mathbf{z}) & > 0 \qquad Otherwise \end{cases} \tag{18}$$

Suppose at $t = 0$ we know the worst case the initial values of q, $\mathbf{x}$ and $\mathbf{z}$ which maximize $V(t=0)$. Let V_{max} be the largest possible value of $V(0)$. Furthermore, suppose we know the worst case initial values of y, $\dot{y}$, c, and $\dot{c}$ which maximize $|g(t=0)|$ and $|f(t=0)|$. Let f^* and g^* be the maximum possible values of $|f(t=0)|$ and $|g(t=0)|$. At $t > 0$, $f(.)$ and $g(.)$ are known and can be computed. Define

$$|f|_{max}(t) = \begin{cases} |f^*| & t = 0 \\ |f(t)| & t > 0 \end{cases} \qquad |g|_{max}(t) = \begin{cases} |g^*| & t = 0 \\ |g(t)| & t > 0 \end{cases} \tag{19}$$

*** Since f and g are functions of y, $\dot{y}$, c, and $\dot{c}$, they are known online. So, we can simply let $|f|_{max}(t) = |f(t)|$ and $|g|_{max}(t) = |g(t)|$. Therefore, I use $|f(t)|$ and $|g(t)|$ directly and revised the above and u_{max} as follows.

Suppose at $t = 0$ we know the worst case the initial values of q, $\mathbf{x}$ and $\mathbf{z}$ which maximize V at $t = 0$. Denote the resulting maximum by V_{max}. Define

$$\text{sign}(q) = \begin{cases} 1 & q > 0 \\ 0 & q = 0 \\ -1 & q < 0 \end{cases} \tag{20}$$

*** I changes 1+V to 1+2V in u_{max} so that Lemma 1 in your new version is no longer needed, because by Equation (17), $x_i^2 \leq 2V$.

We have the following lemma.

Lemma 1: Let $u_2 = (u_{max} - |\varepsilon|)\text{sign}(q)$, where ε is a (small) constant and

$$u_{max} = (1-\gamma)(\sum_i (1+2V_{max})|g_i| + \sum_j (1+2V_{max})|f_j|) \tag{21}$$

with $\gamma > 1$. Then $\dot{V}(q,\mathbf{x},\mathbf{z}) = \frac{d}{dt}V(q,\mathbf{x},\mathbf{z}) \leq -|q||\varepsilon| \leq 0$.

Proof

We first prove the following.

$$\begin{aligned}
&|\sum_j z_j f_j - \sum_i x_i g_i| \\
\leq &\sum_i |x_i|\ |g_i| + \sum_j |z_j|\ |f_j| \\
\leq &\sum_i (1+x_i^2)|g_i| + \sum_j (1+z_j^2)|f_j| \\
&\text{(because } x_i^2 + 1 - |x_i| = (|x_i|-0.5)^2 + 0.75 \geq 0) \\
\leq &\sum_i (1+2V)|g_i| + \sum_j (1+2V)|f_j| \\
&\text{(by Equation (17), } x_i^2 \leq 2V) \\
\leq &\sum_i (1+2V_{max})|g_i| + \sum_j (1+2V_{max})|f_j|.
\end{aligned}$$

Hence, for $\gamma > 1$, $(1-\gamma) < 0$ and

$$\begin{aligned}
&u_{max} - |\varepsilon| \\
=&(1-\gamma)(\sum_i (1+2V_{max})|g_i| + \sum_j (1+2V_{max})|f_j|) - |\varepsilon| \\
\leq&(1-\gamma)|\sum_j z_j f_j - \sum_i x_i g_i| - |\varepsilon|
\end{aligned} \tag{22}$$

Therefore,

$$\begin{aligned}
&\dot{V}(q,\mathbf{x},\mathbf{z}) \\
=&q\dot{q} + \sum_i x_i \dot{x}_i + \sum_j z_j \dot{z}_j \\
=&q\dot{q} + \sum_i x_i \dot{\omega}_i + \sum_j z_j \dot{v}_j \\
=&q(\dot{q} + \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
&\text{(by Equation (15))} \\
=&q(\ddot{e} + \lambda\dot{e} + \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
=&q(\ddot{y} - \ddot{\bar{y}} + \lambda\dot{e} + \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
=&q(u - \sum_i a_i g_i + \sum_i \tilde{a}_i r_i - \sum_j b_j f_j + \lambda\dot{e} \\
&+ \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
&\text{(by Equations (12) and (13))} \\
=&q(\sum_j v_j f_j - \sum_i \omega_i g_i - \lambda\dot{e} - \sum_i \tilde{a} r_i + \sum_i \tilde{a} g_i + u_2 \\
&- \sum_i a_i g_i + \sum_i \tilde{a}_i r_i - \sum_j b_j f_j \\
&+ \lambda\dot{e} + \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
&\text{(by Equation (16))} \\
=&q(\sum_j (v_j - b_j) f_j - \sum_i (\omega_i + a_i - \tilde{a}) g_i + u_2 \\
&+ \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
=&q(\sum_j z_j f_j - \sum_i x_i g_i + u_2 + \sum_i x_i \gamma g_i - \sum_j z_j \gamma f_j) \\
&\text{(by Equation (14))} \\
=&q(1-\gamma)(\sum_j z_j f_j - \sum_i x_i g_i) + qu_2 \\
=&q(1-\gamma)(\sum_j z_j f_j - \sum_i x_i g_i) + q(u_{max} - |\varepsilon|)sign(q) \\
=&q(1-\gamma)(\sum_j z_j f_j - \sum_i x_i g_i) + |q|(u_{max} - |\varepsilon|) \\
\leq&q(1-\gamma)(\sum_j z_j f_j - \sum_i x_i g_i) \\
&+ |q|(1-\gamma)|\sum_j z_j f_j - \sum_i x_i g_i| - |q||\varepsilon| \\
&\text{(by Equation (22))} \\
\leq&-|q||\varepsilon| \\
&\text{(because } (1-\gamma) < 0 \text{ implies } (1-\gamma)A \\
&+ (1-\gamma)|A| \leq 0 \text{ for } A = q(\sum_j z_j f_j - \sum_i x_i g_i)).
\end{aligned}$$

Therefore

$$\begin{cases} \frac{d}{dt}V(q,\mathbf{x},\mathbf{z}) < 0, & \text{if } |q| > 0 \\ \frac{d}{dt}V(q,\mathbf{x},\mathbf{z}) = 0, & \text{if } q = 0 \end{cases} \tag{23}$$

Q.E.D.

Lemma 2: Suppose q is bounded. Then e and $\dot{e}$ are bounded.

Proof

Consider the linear system described by $q = e + \lambda\dot{e}$ with the input q and outputs e and $\dot{e}$. The system is stable because $\lambda > 0$. It is well-known that for any stable linear system, bounded input cannot generate unbounded outputs.

Q.E.D.

Theorem 2: The parameters ω_i and v_j converge to some final values. Furthermore, $q(t)$ is asymptotically stable and converges to 0.

Proof

We first prove that $q(t)$ is asymptotically stable. Consider the candidate Lyapunov function $V_q(q) = V(q,0,0) = \frac{1}{2}q^2(t)$. Clearly, $V_q(0) = 0$ if $q = 0$ and $V_q(t) > 0$ otherwise. Also, from equation (23), $\dot{V}_q(t) = 0$ if $q = 0$ and $\dot{V}_q(t) < 0$ otherwise. Hence, $V_q(t)$ is a Lyapunov function. Therefore, $q(t)$ is asymptotically stable and converges to 0.

Now, q is asymptotically stable $\Rightarrow q$ is bounded $\Rightarrow e$ and $\dot{e}$ are bounded (by Lemma 2) $\Rightarrow y$ and $\dot{y}$ are bounded (because the reference model is stable implies $\tilde{y}$ and $\dot{\tilde{y}}$ are bounded) $\Rightarrow g_i$ and f_j are bounded $\Rightarrow \lim_{t\to\infty} \dot{\omega}_i = 0$ and $\lim_{t\to\infty} \dot{v}_j = 0$ (by Equation (15) and $\lim_{t\to\infty} q = 0$). Hence, the parameters ω_i and v_j converge to some final values.

Q.E.D.

Remark 1: The convergence of the parameters to their desired values (not just to some final values) highly depends on the input characteristics and plant dynamics. Simulations below show that with persistent excitation, parameters do converge to their desired values.

Clearly, the worst case conditions give direction on how to set the adaptation rate γ. If there is certainty that the initial values are all small, then γ can be practically large. If γ is made large, then there is increased chattering initially. However, as the parameters adapt, $|q|$ gets smaller. We can take advantage of q and the parameters convergence to reduce chattering. When $|q|$ is permanently small, the parameters approach their final values, and therefore, the worst case u_{max} can be reduced. We accept a small error and set u_2 to

$$u_2 = (ku_{max}|q| - \frac{1}{k})\text{sign}(q) \quad k \gg 0 \tag{24}$$

Note that $ku_{max}|q| > u_{max}$ and $\dot{V}(t) \leq 0$ when $|q| > \frac{1}{k}$. Equation (24) may cause $\dot{V}(t) > 0$ when $|q| < \frac{1}{k}$. We set $\dot{\omega}_i = 0$ and $\dot{v}_j = 0$ when $|q| < \frac{1}{k}$ to avoid parameter drift. This is not a problem because $|q| \approx 0$ anyway. If $\dot{V}(t) > 0$ remains true, then $|q| > \frac{1}{k}$ will occur and then $\dot{V}(t) < 0$. Therefore, to reduce chattering, we made q stable (not asymptotically stable) and bounded by $\frac{1}{k}$. We reduced chattering but acquired a very small error, which is a very practical approach for many applications.

7. SIMULATION RESULTS

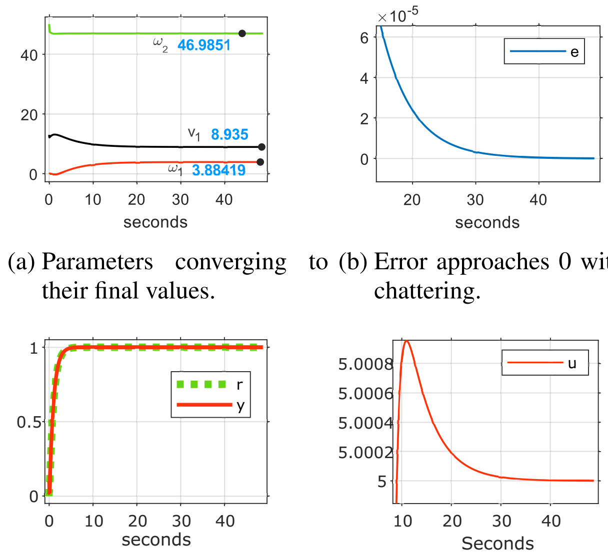
7.1. LINEAR PLANT WITH LINEAR REFERENCE MODEL

Consider the following plant $G(s)$ and reference $R(s)$:

$$R(s) = \frac{20}{s^2+20s+20} \text{ and } G(s) = \frac{1}{s^2-15s+5} \tag{25}$$

*** Please specify g_i, r_i, and f_j.

Assume that the plant's initial condition is $y(0) = 1$. Let $\lambda = 100$, $\gamma = 10$, $k = 100$, and initially, $\omega_2(0) = 2$, $\omega_1(0) = -2$, $v_1(0) = 5$. The desired final values for the parameters are $\omega_1 = 15$, $\omega_2 = 35$ and $v_1 = 20$ to satisfy Equation (14). We first run the simulation using a DC input. The results are shown in Figure 6. In this example no chattering is observed. Even though the parameters did not converge to their desired values in Figure 6a, they did converge to some final values. The same simulation is repeated using three sinusoidal inputs of $10Hz$, $1Hz$ and $0.4Hz$ to provide persistent excitation. The parameters did converge to their desired values as shown in Figure 7a. Also, Figure 7 shows that chattering is present yet is small after the parameters converged on their values, and there are no at the input or the error.



(a) Parameters converging to their final values.

(b) Error approaches 0 with no chattering.

(c) Tracking performance.

(d) Control effort u.

Fig. 6. Controller performance using DC input for unstable linear plant.

7.2. NONLINEAR PLANT WITH LINEAR REFERENCE MODEL

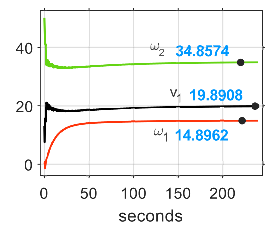
In this example, for the same input, The plant and reference are set to

$$\begin{aligned} \ddot{y} &= u + 10\dot{y}^2 y - 5|sin(y)| \\ R(s) &= \frac{20}{s^2+20s+20} \end{aligned} \tag{26}$$

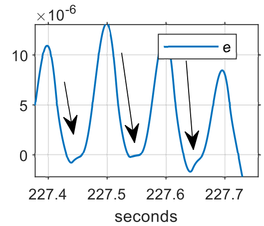
In this case, $g_1 = |sin(y)|$, $g_2 = \dot{y}y^2$, $r_1 = \tilde{y}$, $r_2 = \dot{\tilde{y}}$, and $f_1 = 1$. The desired final values for the parameters are $v_1 = 20$, $\omega_1 = 15$ and $\omega_2 = 30$.

Figure 8d shows that even for nonlinear plant, the parameters converged to their desired values. This convergence gives visibility into the parameters of the system and can help in applications where diagnostics are important, as the plant's coefficients can be monitored.

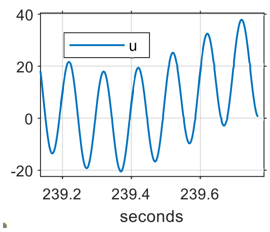
As can be seen through both examples, and similar to [20] and [18], parameter adaptation reduces chattering. This is achieved by reducing the terms $(-a_i - \omega_i + \tilde{a}_i)$ and $(v_j - b_j)$ in equation (14), and therefore, the amplitude of the discontinuous switching action can be reduced.



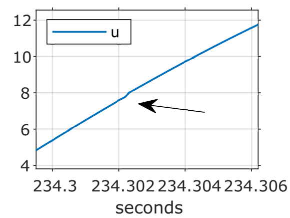
(a) Parameters converging to their desired values.



(b) Chattering is negligible after parameter convergence.

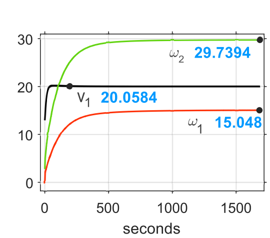


(c) Control u showing small chattering.

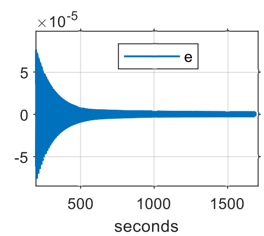


(d) Control u showing small chattering.

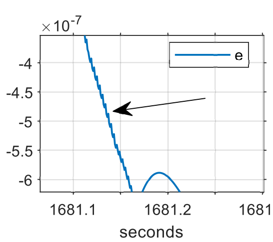
Fig. 7. Controller performance using 3 sinusoidal inputs superimposed for unstable linear plant



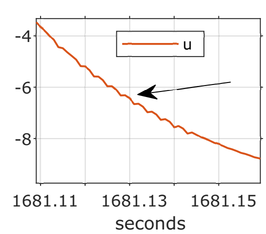
(a) Parameters converging to their desired values.



(b) Error is minimized with no spikes.



(c) Error changes due to chattering.



(d) Chattering is negligible after parameter convergence.

Fig. 8. Controller performance for nonlinear plant with linear rerference model.

8. CONCLUSION

In this paper, we investigated the applicability of the B-L learning algorithm to adaptive control. We found that combining the dynamics of the B-L learning algorithm with sliding mode control improves the performance of the controller. We derived the control law which guarantee the convergence of the parameters to a final value, which reduces chattering. We also showed that given enough persistent excitation at the input, the parameters converge to their desired values. We showed that it is possible to adapt a class of nonlinear plants to linear reference models, and gain information about the plant's parameters. In summary, the B-L learning algorithm is applicable in adaptive control when combined with sliding mode control. For future work, we plan to further investigate which chatter reduction techniques in the literature fits well with our method while ensuring the parameters don't drift. We will also investigate adapting to the uncertainties related to the actuator dynamics. Furthermore, the work in Section 5 will be further investigated to see if predictive AI can be used to replace $e(t+h)$, and how a controller may be designed considering the prediction errors.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] D. G. Stork, "Is backpropagation biologically plausible," in *International Joint Conference on Neural Networks*, vol. 2, pp. 241–246, IEEE Washington, DC, 1989.

[2] D. P. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc., 2001.

[3] K. Gurney, *An introduction to neural networks*. CRC press, 2014.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[5] Y. Chauvin and D. E. Rumelhart, *Backpropagation: theory, architectures, and applications*. Psychology Press, 2013.

[6] A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 04 2018.

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

[8] A. T. Nguyen, M. S. Rafaq, H. H. Choi, and J.-W. Jung, "A model reference adaptive control based speed controller for a surface-mounted permanent magnet synchronous motor drive," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9399–9409, 2018.

[9] O. Makke and O. Gusikhin, "Connected vehicle prognostics framework for dynamic systems," *Advances in Intelligent Systems and Computing*, 2018.

[10] B. Mohamed, K. Kara, O. Ait sahed, and M. Hadjili, "Adaptive fourier series neural network pid controller," *International Journal of Control, Automation and Systems*, vol. 19, 07 2021.

[11] D. Kim, H.-S. Oh, and I.-C. Moon, "Black-box modeling for aircraft maneuver control with bayesian optimization," *International Journal of Control, Automation and Systems*, vol. 17, 05 2019.

[12] J. Pongfai, W. Assawinchaichote, C. Angeli, P. Shi, and X. Su, "Optimal pid controller autotuning design for mimo nonlinear systems based on the adaptive slp algorithm," *International Journal of Control Automation and Systems*, 09 2020.

[13] S. D. Sahputro, F. Fadilah, N. A. Wicaksono, and F. Yusivar, "Design and implementation of adaptive pid controller for speed control of dc motor," in *2017 15th International Conference on Quality in Research (QiR) : International Symposium on Electrical and Computer Engineering*, pp. 179–183, 2017.

[14] M. Kim and S. Lee, "Pid with a switching action controller for nonlinear systems of second-order controller canonical form," *International Journal of Control, Automation and Systems*, vol. 19, 03 2021.

[15] A. Ramírez-Mendoza, L. Amezquita-Brooks, O. Garcia Salazar, and W. Ramírez-Mendoza, "Fuzzy adaptive neurons applied to the identification of parameters and trajectory tracking control of a multi-rotor unmanned aerial vehicle based on experimental aerodynamic data," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 380–393, 06 2020.

[16] S. Tian and X.-f. Fu, "Fuzzy adaptive pid strategy for asynchronous machines direct torque control," in *Fuzzy Information and Engineering Volume 2* (B. Cao, T.-F. Li, and C.-Y. Zhang, eds.), (Berlin, Heidelberg), pp. 1249–1255, Springer Berlin Heidelberg, 2009.

[17] C. J. Fallaha, M. Saad, H. Y. Kanaan, and K. Al-Haddad, "Sliding-mode robot control with exponential reaching law," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 600–610, 2011.

[18] S. Wang, C. Jiang, Q. Tu, and C. Zhu, "Sliding mode control with an adaptive switching power reaching law," *Scientific Reports*, vol. 13, 09 2023.

[19] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628–3637, 2016.

[20] A. K. Junejo, W. Xu, C. Mu, M. M. Ismail, and Y. Liu, "Adaptive speed control of pmsm drive system based a new sliding-mode reaching law," *IEEE Transactions on Power Electronics*, vol. 35, no. 11, pp. 12110–12121, 2020.

[21] R. Franco, H. Ríos, and A. F. de Loza, "A model reference adaptive continuous sliding-mode control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 96–101, 2019.

[22] E. Guzman and J. A. Moreno, "A new finite-time convergent and robust direct model reference adaptive control for siso linear time invariant systems," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7027–7032, 2011.

[23] R. D. Brandt and F. Lin, "Supervised learning in neural networks without feedback network," in *Intelligent Control, 1996., Proceedings of the 1996 IEEE International Symposium on*, pp. 86–90, IEEE, 1996.

[24] F. Lin, R. D. Brandt, and G. Saikalis, "Self-tuning of pid controllers by adaptive interaction," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 5, pp. 3676–3681, IEEE, 2000.

[25] R. D. Brandt and F. Lin, "Adaptive interaction and its application to neural networks," *Information Sciences*, vol. 121, no. 3-4, pp. 201–215, 1999.

[26] F. Lin, "Supervised learning in neural networks: Feedback-network-free implementation and biological plausibility," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[27] O. Makke and F. Lin, "Learning in dynamic systems and its application to adaptive pid control." https://arxiv.org/abs/2308.10851, 2023.

[28] S. J. Mason, "Feedback theory-further properties of signal flow graphs," *Proceedings of the IRE*, vol. 44, no. 7, pp. 920–926, 1956.

[29] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1997.

[30] S. Boyd and S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

Omar Makke is a PhD student at Wayne State University. He currently holds the position of Technical Expert In Connected Vehicle Analytics at Ford Motor Company, and has received his B.S. and M.S. in Electrical Engineering from University of Michigan - Dearborn in 2006 and 2008 resepctively.

Feng Lin received his B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1984 and 1988, respectively. He was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA, from 1987 to 1988. Since 1988, he has been with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, where he is currently a Professor. He is a fellow of IEEE.