

Article

# Data-Driven Adaptive Controller Based on Hyperbolic Cost Function for Non-Affine Discrete-Time Systems with Variant Control Direction

Miriam Flores-Padilla \* and Chidentree Treesatayapun 

Department of Robotic and Advanced Manufacturing, Cinvestav, Av. Industrial Metalurgia, Ramos Arizpe 25900, México; chidentree@cinvestav.edu.mx

\* Correspondence: miriam.flores@cinvestav.mx; Tel.: +52-(844)-438-9600

**Abstract:** As technology evolves, more complex non-affine systems are created. These complex systems are hard to model, whereas most controllers require information on systems to be designed. This information is hard to obtain for systems with varying control directions. Therefore, this study introduces a novel data-driven estimator and controller tailored for single-input single-output non-affine discrete-time systems. This approach focuses on cases when the control direction varies over time and the mathematical model of the system is completely unknown. The estimator and controller are constructed using a Multiple-input Fuzzy Rules Emulated Network framework. The weight vectors are updated through the gradient descent optimization method, which employs a unique cost function that multiplies the error by a hyperbolic tangent. The stability analyses demonstrate that both the estimator and controller converge to uniformly ultimately bounded (UUB) functions of Lyapunov. To validate the results, we show experimental tests of force control that were executed on the z-axis of a drive-controlled 3D scanning robot. This system has a varying control direction, and we also provide comparison results with a state-of-the-art controller. The results show a mean absolute percentage tracking error smaller than one percent on the steady state and the expected variation in the system's control direction.

**Keywords:** model-free adaptive control; discrete-time systems; non-affine systems; fuzzy neural networks; experimental validation; data-driven control; data-driven model estimation



**Citation:** Flores-Padilla, M.; Treesatayapun, C. Data-Driven Adaptive Controller Based on Hyperbolic Cost Function for Non-Affine Discrete-Time Systems with Variant Control Direction. *Appl. Syst. Innov.* **2024**, *7*, 38. <https://doi.org/10.3390/asi7030038>

Academic Editor: Geng Yang

Received: 15 February 2024

Revised: 3 April 2024

Accepted: 18 April 2024

Published: 28 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the continually growing landscape of control systems, adaptive controllers have garnered significant attention owing to their adeptness in handling the intricate dynamics of modern systems, which are often unknown and highly non-linear [1,2]. With these sophisticated systems, there has been a surge in the availability of system status information, enabling adaptive controllers to rely less on system knowledge. This shift has led to the emergence of data-driven controllers (DDCs) and model estimators [3,4]. Researchers typically categorize the adaptation of DDCs into online learning, offline learning, and hybrid approaches that combine both online and offline learning strategies.

Along the DDCs' most popular online methods, model-free adaptive control (MFAC) [5–8] has the advantages of a low computational cost for other online methods and no information required about the system besides the control direction. This method has been mainly used for non-linear systems. Regarding the offline methods, iterative feedback tuning [9,10] (IFT) and virtual reference feedback tuning [11,12] (VRFT) focus on parameter adaptation/identification; IFT adapts with iterations according to the gradient descent method whereas VRFT looks for the global minimum of the data available. Both require some controlled experiments on the system to have data on its behavior previous to controlling the system. Offline methods usually have lower computational costs than online methods but require more information on the system. No offline methods have reported

results for systems with varying control directions. Hybrid methods have the popular iterative learning control (ILC) [13,14] that is commonly used for systems with repetitive tasks. This method has the advantage of decreasing error in each cycle but requires a bit of prior knowledge of the system. These types of controllers have a common disadvantage: they need to know the control direction of the system.

In 1983, Roger D. Nussbaum [15] proposed a solution of adaptive control for systems where the control direction is unknown. His proposed controller could deal with the unknown control direction by adapting a control gain. The proposed function slowly adapts to the unknown control directions. It has been used to solve different problems such as those pertaining to non-affine systems [16,17], non-linear systems [18], switched systems [19], reducing the computational cost [20,21], industrial applications [22,23], and time-varying control gain with no change in the sign [24]. Unfortunately, to the authors' knowledge, only a very few articles address the problem of systems with varying control directions (control gain with sign change) [25,26]. Both articles use a Nussbaum-type function alongside fuzzy observers and controllers. Their focus is on affine non-linear systems; they only report simulation results to validate their theoretical analysis, and one of them does not present a graph of the control-related parameter estimation.

Unlike affine systems, non-affine systems have a non-linear relation with the system output regarding the control input [27–29]. This property of the non-affine systems makes it very difficult to find an exact solution for them. Non-affine systems have countless applications such as active magnetic bearings, aircraft dynamics, biochemical processes, dynamic models in pendulum control, underwater vehicles, and so on [30,31]. The common approach to controlling non-affine systems is adaptive control, which mostly involves neural networks where the control direction is either known or estimated with the Nussbaum gain [32–34]. Some of these systems also show time-varying sign behavior when we talk about the correlation between the output of the system and the control input. Therefore, our focus is the development of an adaptive controller for non-affine systems with time-varying control direction behavior.

This work presents two significant contributions. First, we introduce a novel controller capable of effectively managing non-affine, non-linear discrete-time systems with varying control directions. This adaptive controller showcases remarkable versatility in handling such systems, enabling precise control even amidst changing control directions. Then, we propose a novel cost function, employing a hyperbolic tangent of the error multiplied by the error, diverging from traditional quadratic or absolute functions. Investigations reveal that this innovative cost function facilitates faster responses to aggressive system changes while ensuring smoother control laws and estimated function responses. These enhancements significantly improve the overall system performance and may find valuable applications in redundancy scenarios for future works.

The rest of this work is organized as follows: Section 2 outlines the requirements and assumptions of the systems to control. Section 3 introduces the model estimator and its upgrade law according to the gradient descent method. We propose an unusual cost function: a hyperbolic tangent of the error multiplied by the error. The stability proof of the estimator is provided at the end of Section 3. Section 4 develops a model-free adaptive controller, where the weight vector is also upgraded according to the gradient descent method with a similar cost function as the estimator. The stability proof of the closed-loop system is provided at the end of Section 4. Section 5 shows the performance of the controller and estimator for a highly non-linear system with a changing control direction. This section also provides a comparison of the experimental results with a state-of-the-art controller. Finally, Section 6 offers suggestions for future work and concluding remarks.

## 2. Problem Formulation

A single-input single-output non-affine non-linear discrete-time system described as

$$y(k+1) = F(y(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u)), \quad (1)$$

with unknown indices  $n_y$  and  $n_u$ , it has the following affine representation:

$$\begin{aligned} y(k+1) &= f(y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \\ &\quad + g(y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u))u(k) \\ &\quad + \varepsilon_h(ky(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u)), \\ &= f(k) + g(k)u(k) + \varepsilon_h(k), \end{aligned} \tag{2}$$

where  $f(k)$  and  $g(k)$  are unknown functions and  $\varepsilon_h(\cdot)$  is a bounded residual error when the non-affine system (1) meets the following assumptions:

**Assumption 1.** The non-linear function  $F(\cdot)$  in (1) needs to be continuous regarding the control law  $u(k)$ . This implies that

$$\frac{\partial y(k+1)}{\partial u(k)} \approx g(k), \tag{3}$$

where  $0 < |g(k)| \leq g^M$  and  $g^M$  is an unknown positive constant. Therefore, the system (1) is controllable with unknown and varying control directions.

According to Assumption 1, the control direction is determined by utilizing the sign function of  $g(k)$  in (3) as

$$\text{sign}\left\{\frac{\Delta y(k+1)}{\Delta u(k)}\right\} = \text{sign}\{g(k)\}, \tag{4}$$

where  $\Delta u(k) \neq 0$ .

Therefore, the equivalent model based on MiFREN [35] is developed in the following section.

### 3. Model Estimator

In this work, a class of non-linear discrete-time systems described in (1) and (2), where  $\delta(k) \triangleq \{y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)\}$ ,  $f(k) \triangleq f(\delta_f(k))$ ,  $g(k) \triangleq g(\delta_g(k))$  and  $\varepsilon_h(k) \triangleq \varepsilon_h(\delta_\varepsilon(k))$ , is rewritten as

$$y(k+1) = f(k) + g(k)u(k) + \varepsilon_h(k), \tag{5}$$

where  $\varepsilon_h(k)$  is a bounded residual error such that  $|\varepsilon_h(k)| \leq \varepsilon_M$ . It is noticeable that  $f(k)$  and  $g(k)$  are unknown functions. Therefore, the adaptive network MiFREN is used to estimate those functions as

$$f(k) = \varphi^T(k)\beta_f^*, \tag{6}$$

$$g(k) = \varphi^T(k)\beta_g^*, \tag{7}$$

where  $\varphi(k)$  is the multidimensional vector of the membership functions and  $\beta_f^*$  and  $\beta_g^*$  are the unknown ideal weight vectors.

By utilizing the equivalent model based on MiFREN, the dynamics in (5)–(7) are estimated as

$$\hat{y}(k+1) = \hat{f}(k) + \hat{g}(k)u(k). \tag{8}$$

Therefore, the MiFREN implementation leads to

$$\hat{f}(k) = \varphi^T(k)\beta_f(k), \tag{9}$$

and

$$\hat{g}(k) = \varphi^T(k)\beta_g(k), \tag{10}$$

where  $\beta_f(k)$  and  $\beta_g(k)$  are iterative weight vectors used to estimate the functions  $\hat{f}(k)$  and  $\hat{g}(k)$ , respectively. This implementation is illustrated in the diagram of Figure 1. This diagram shows that the inputs of the estimator are the system output and the estimation error at the  $k$ th iteration to avoid causality issues. Both inputs enter simple fuzzy membership functions  $\mu$  and are later combined in the multidimensional membership function  $\varphi(k)$ . Then, the estimation proceeds as explained in Equations (8)–(10).

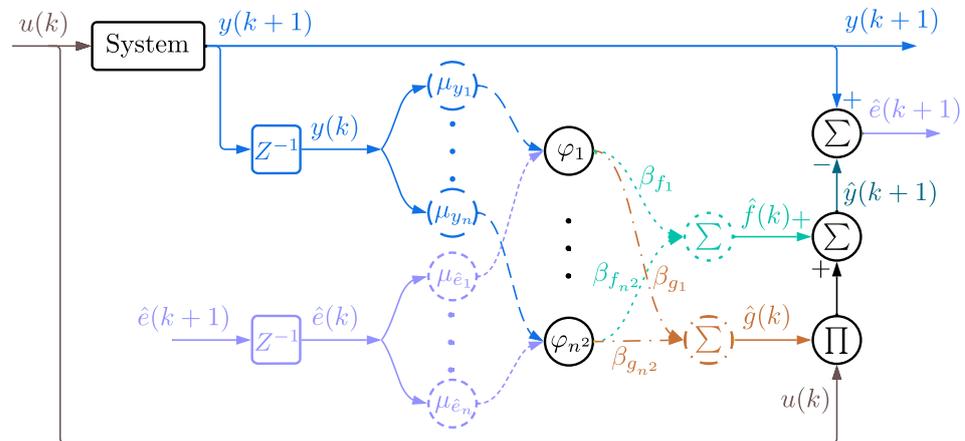


Figure 1. Estimator diagram.

The weight vectors  $\beta_f(k)$  and  $\beta_g(k)$  are updated by the gradient descent method. Thus, the estimation error  $\hat{e}(k + 1)$  is introduced as

$$\hat{e}(k + 1) = y(k + 1) - \hat{y}(k + 1). \tag{11}$$

Recalling (8)–(10) with (11), it yields

$$\begin{aligned} \hat{e}(k + 1) &= \varphi^T(k)\beta_f^* + \varphi^T(k)\beta_g^*u(k) + \varepsilon_h(k) - \varphi^T(k)\beta_f(k) - \varphi^T(k)\beta_g(k)u(k), \\ &= \varphi^T(k)\tilde{\beta}_f(k) + \varphi^T(k)\tilde{\beta}_g(k)u(k) + \varepsilon_h(k), \end{aligned} \tag{12}$$

where  $\tilde{\beta}_f(k) = \beta_f^* - \beta_f(k)$  and  $\tilde{\beta}_g(k) = \beta_g^* - \beta_g(k)$ .

The cost function  $\hat{E}(k + 1)$  is secondly selected as a semi-definite positive function:

$$\hat{E}(k + 1) = \tanh(\hat{e}(k + 1))\hat{e}(k + 1). \tag{13}$$

It is worth observing that the proposed cost function (13) is developed here to reduce the high-frequency behavior, which will be discussed next in the experimental results.

Therefore, the update laws for the weight parameters are formulated by the gradient descent method as follows

$$\beta_f(k + 1) = \beta_f(k) - \eta_f \frac{\partial \hat{E}(k + 1)}{\partial \beta_f(k)}, \tag{14}$$

and

$$\beta_g(k + 1) = \beta_g(k) - \eta_g \frac{\partial \hat{E}(k + 1)}{\partial \beta_g(k)}, \tag{15}$$

where  $\eta_f$  and  $\eta_g$  are the learning rates for  $\beta_f$  and  $\beta_g$ , respectively.

Using the chain rule, the cost function (13) derivation is obtained as

$$\begin{aligned} \frac{\partial \hat{E}(k + 1)}{\partial \beta_f(k)} &= \frac{\partial \hat{E}(k + 1)}{\partial \hat{e}(k + 1)} \frac{\partial \hat{e}(k + 1)}{\partial \hat{y}(k + 1)} \frac{\partial \hat{y}(k + 1)}{\partial \beta_f(k)}, \\ &= -h_{\hat{e}}(k + 1)\varphi^T(k), \end{aligned} \tag{16}$$

where

$$h_{\hat{e}}(k+1) \triangleq \operatorname{sech}^2(\hat{e}(k+1))\hat{e}(k+1) + \tanh(\hat{e}(k+1)). \tag{17}$$

Then, the cost function (13) derivation is also calculated as

$$\begin{aligned} \frac{\partial \hat{E}(k+1)}{\partial \beta_g(k)} &= \frac{\partial \hat{E}(k+1)}{\partial \hat{e}(k+1)} \frac{\partial \hat{e}(k+1)}{\partial \hat{y}(k+1)} \frac{\partial \hat{y}(k+1)}{\partial \beta_g(k)}, \\ &= -h_{\hat{e}}(k+1)\varphi^T(k)u(k). \end{aligned} \tag{18}$$

Thus, substituting (16) into (14) and (15) into (18), the update laws for the weight parameters can be formulated as follows

$$\beta_f(k+1) = \beta_f(k) + \eta_f h_{\hat{e}}(k+1)\varphi(k), \tag{19}$$

and

$$\beta_g(k+1) = \beta_g(k) + \eta_g h_{\hat{e}}(k+1)\varphi(k)u(k). \tag{20}$$

It is seen that  $\eta_f$  and  $\eta_g$  can play an important role in the model performance. Thus,  $\eta_f$  and  $\eta_g$  are determined as in the following theorem.

**Theorem 1.** *A class of non-affine non-linear discrete-time systems (1) that can be represented in an affine way (5)—meeting the three assumptions made in Section 2—can also be estimated by (8) based on MiFREN. The estimation error, along with the internal signals, is convergent when the estimator parameters are designed following the conditions*

$$\eta_f = \eta_g = \eta_T, \tag{21}$$

$$0 < \eta_T \leq \frac{2\zeta_{\hat{e}}(k+1)}{h_{\max}^2 \varphi_{\max}^2 (1 + u^2(k))}. \tag{22}$$

**Proof.** To verify the convergence of the estimation error along and the internal signals of the estimator, let us select the Lyapunov function as

$$L_{\hat{y}}(k+1) = \tilde{\beta}_f^2(k+1) + \tilde{\beta}_g^2(k+1). \tag{23}$$

Therefore, the Lyapunov function differentiation is calculated as

$$\Delta L_{\hat{y}}(k+1) = \tilde{\beta}_f^2(k+1) - \tilde{\beta}_f^2(k) + \tilde{\beta}_g^2(k+1) - \tilde{\beta}_g^2(k). \tag{24}$$

Utilizing the learning laws developed in (19) and (20), we obtain

$$\begin{aligned} \tilde{\beta}_f(k+1) &= \beta_f^* - \beta_f(k+1), \\ &= \tilde{\beta}_f(k) - \eta_f h_{\hat{e}}(k+1)\varphi(k), \end{aligned} \tag{25}$$

and

$$\begin{aligned} \tilde{\beta}_g(k+1) &= \beta_g^* - \beta_g(k+1), \\ &= \tilde{\beta}_g(k) - \eta_g h_{\hat{e}}(k+1)\varphi(k)u(k), \end{aligned} \tag{26}$$

respectively. By substituting (25) and (26) into (24), we have

$$\begin{aligned} \Delta L_{\hat{y}}(k+1) &= \left( \tilde{\beta}_f(k) - \eta_f h_{\hat{e}}(k+1)\varphi(k) \right)^2 - \tilde{\beta}_f^2(k) \\ &\quad + \left( \tilde{\beta}_g(k) - \eta_g h_{\hat{e}}(k+1)\varphi(k)u(k) \right)^2 - \tilde{\beta}_g^2(k), \\ &= -2\eta_f \tilde{\beta}_f^T(k)\varphi(k)h_{\hat{e}}(k+1) + \eta_f^2 h_{\hat{e}}^2(k+1)\varphi^T(k)\varphi(k) \\ &\quad - 2\eta_g \tilde{\beta}_g^T(k)\varphi(k)h_{\hat{e}}(k+1)u(k) \end{aligned}$$

$$+\eta_g^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) u^2(k). \tag{27}$$

According to (21) where  $\eta_f = \eta_g = \eta_T$ , the relation in (27) can be rewritten as

$$\begin{aligned} \Delta L_{\hat{y}}(k+1) &= -2\eta_T \left( \tilde{\beta}_f^T(k) \varphi(k) + \tilde{\beta}_g^T(k) \varphi(k) u(k) \right) h_{\hat{e}}(k+1) \\ &\quad + \eta_T^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) \left( 1 + u^2(k) \right). \end{aligned} \tag{28}$$

From the definition of the estimation error (12), we have

$$\hat{e}(k+1) - \varepsilon_h(k) = \varphi^T(k) \tilde{\beta}_f(k) + \varphi^T(k) \tilde{\beta}_g(k) u(k). \tag{29}$$

By employing (28), it yields

$$\begin{aligned} \Delta L_{\hat{y}}(k+1) &= -2\eta_T \hat{e}(k+1) h(k+1) + 2\eta_T \varepsilon_h(k) h_{\hat{e}}(k+1) \\ &\quad + \eta_T^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) \left( 1 + u^2(k) \right). \end{aligned}$$

Let us define  $\zeta_{\hat{e}}(k+1) \triangleq \hat{e}(k+1) h_{\hat{e}}(k+1)$ ; thus, from the last equation, we have

$$\begin{aligned} \Delta L_{\hat{y}}(k+1) &= -2\eta_T \zeta_{\hat{e}}(k+1) + 2\eta_T \varepsilon_h(k) h_{\hat{e}}(k+1) \\ &\quad + \eta_T^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) \left( 1 + u^2(k) \right). \end{aligned} \tag{30}$$

Figure 2 shows the cost function  $E(\hat{e}(k))$ , a semi-definite positive function dependent on the estimation error  $\hat{e}(k)$ . Furthermore the function  $h_{\hat{e}}(k+1)$  in Figure 3, shows that  $h_{\hat{e}}(k+1)$  is a bounded function regardless of the value of  $\hat{e}(k+1)$  with the limits  $-1.2 \leq h_{\hat{e}}(k+1) \leq 1.2$ .

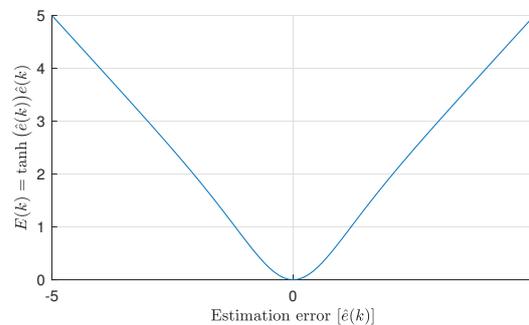


Figure 2. Cost function  $E(\hat{e}(k))$ .

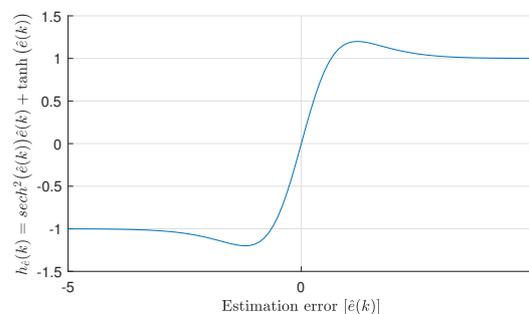


Figure 3. Function  $h_{\hat{e}}(\hat{e}(k))$ .

Given that  $\varepsilon_h(k) h_{\hat{e}}(k+1)$  is a bounded function with an unknown sign, the Lyapunov function differentiation (30) can be rewritten as

$$\Delta L_{\hat{y}}(k+1) \leq -2\eta_T \zeta_{\hat{e}}(k+1) + \eta_T^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) \left( 1 + u^2(k) \right)$$

$$+2|\eta_T \varepsilon_h(k) h_{\hat{e}}(k+1)| < 0 \tag{31}$$

since, for definite positive values  $a$  and  $b$  with an unknown value  $c$ , we can say that

$$-a + b + c \leq -a + b + |c|.$$

Limits for the learning rate  $\eta_T$  are derived according to the known functions

$$-2\eta_T \xi_{\hat{e}}(k+1) + \eta_T^2 h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) (1 + u^2(k)) < 0,$$

as

$$0 < \eta_T \leq \frac{2\xi_{\hat{e}}(k+1)}{h_{\hat{e}max}^2 \varphi_{max}^2 (1 + u_{max}^2)} \leq \frac{2\xi_{\hat{e}}(k+1)}{h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) (1 + u^2(k))},$$

where  $h_{max} \geq |h_{\hat{e}}(k+1)|$ ,  $u_{max} \geq |u(k)|$ , and  $\varphi_{max}^2 \geq \varphi^T(k) \varphi(k) \forall t \geq 0$ . It is suggested to set the learning rate as

$$\eta_T \approx \frac{2\xi_{\hat{e}}(k+1)}{h_{\hat{e}max}^2 \varphi_{max}^2 (1 + u^2(k))} \tag{32}$$

when no offline learning has taken place to accelerate the error convergence to a bonded compact set. In this case, a zero-initial-weight vector is recommended if there is no human knowledge of the system behavior to pass into the neural network. If the initial parameters of the estimator are obtained by offline learning or previous behavioral knowledge of the system, the learning rate can be set as

$$\eta_T \approx \frac{2\xi_{\hat{e}}(k+1)}{h_{\hat{e}max}^2 \varphi_{max}^2 (1 + u_{max}^2)} \tag{33}$$

Recalling the differentiation of the Lyapunov (31), and with the previous statements, it can be seen that the differentiation is semi-definite negative when the estimation error co-related parameter  $\xi_{\hat{e}}(k+1)$  is bounded as

$$\xi_{\hat{e}}(k+1) > \Omega_{\xi}, \tag{34}$$

where  $\Omega_{\xi} \triangleq \frac{1}{2} \eta_T h_{\hat{e}}^2(k+1) \varphi^T(k) \varphi(k) (1 + u^2(k)) + |\varepsilon_h(k) h_{\hat{e}}(k+1)|$ . This boundary concludes the stability proof, where the estimation error and estimator's internal signals are established as UUB according to the proposed Lyapunov function (for more information, see the Lyapunov extension Theorem 2.5.7 [36]). □

The following section proposes an adaptive controller that can deal with varying and unknown control directions, based on information provided by the estimator.

#### 4. Data-Based Model-Free Adaptive Controller

An adaptive data-driven controller is proposed with a direct control scheme (Figure 4). The adaptive controller depends on the closed-loop system tracking error and the desired trajectory. The upgrade of the weight vector  $\beta(k)$  depends on the model estimator as described in this section.

Defining the system's tracking error as

$$e(k+1) \triangleq r(k+1) - y(k+1), \tag{35}$$

where  $r(k+1)$  is the desired trajectory, we propose a MiFREN-based adaptive controller as

$$u(k) = \varphi_c^T(k) \beta_c(k), \tag{36}$$

where  $\varphi_c(k)$  is a multidimensional membership-function vector and  $\beta_c(k)$  is the weight vector for the controller. The fundamental difference of the proposed controller is found

in the weight vector actualization method. The actualization will be performed with the gradient descent method and a novel cost function: the hyperbolic tangent of the tracking error multiplied by the tracking error. It is worth noticing that this is a model-free adaptive controller, and no further information on the system is required at this point. The actualization method for the weight vector  $\beta_c(k)$  will be discussed in more detail in this section.

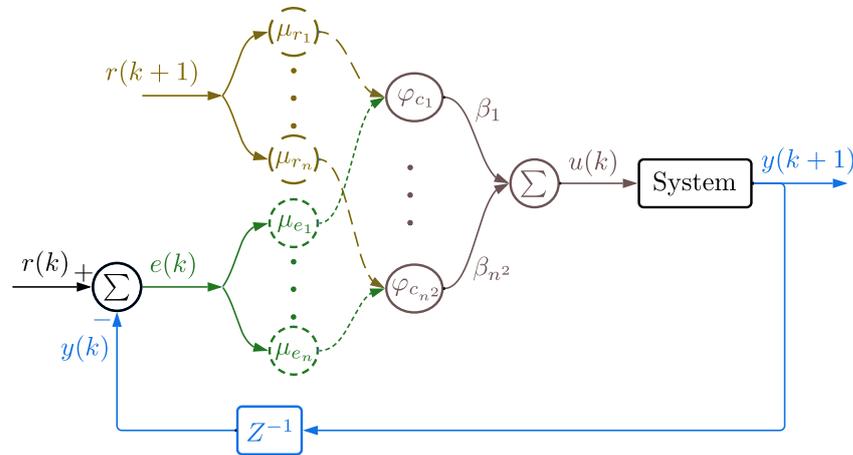


Figure 4. Model-free controller diagram.

The upgrade law of weight vectors is defined according to the gradient descent method as

$$\beta_c(k+1) = \beta_c(k) - \eta \left[ \frac{\partial E(k+1)}{\partial \beta_c(k)} \right], \tag{37}$$

with the cost function

$$E(k+1) = \tanh(e(k+1))e(k+1). \tag{38}$$

As was stated for the estimator cost function, this type of function has the advantage of being smooth near the origin, unlike functions with absolute values.

The partial derivative needed for the upgrade law (37) is obtained with the chain rule

$$\begin{aligned} \frac{\partial E(k+1)}{\partial \beta_c(k)} &= \left[ \frac{\partial E(k+1)}{\partial e(k+1)} \right] \left[ \frac{\partial e(k+1)}{\partial \beta_c(k)} \right], \\ &= \left[ \operatorname{sech}^2(e(k+1))e(k+1) + \tanh(e(k+1)) \right] \left[ \frac{\partial e(k+1)}{\partial \beta_c(k)} \right]. \end{aligned}$$

and the partial derivative of the tracking error regarding the weight vector is obtained as

$$\begin{aligned} \frac{\partial e(k+1)}{\partial \beta_c(k)} &= \left[ \frac{\partial e(k+1)}{\partial y(k+1)} \right] \left[ \frac{\partial y(k+1)}{\partial u(k)} \right] \left[ \frac{\partial u(k)}{\partial \beta_c(k)} \right], \\ &= - \left[ \frac{\partial y(k+1)}{\partial u(k)} \right] \varphi_c(k). \end{aligned}$$

The term  $\frac{\partial y(k+1)}{\partial u(k)}$ , according to Theorem 1, is estimated as

$$\begin{aligned} \frac{\partial y(k+1)}{\partial u(k)} &\approx \frac{\partial [f(k) + g(k)u(k) + \varepsilon_h(k)]}{\partial u(k)} \\ &\approx g(k) \approx \hat{g}(k). \end{aligned} \tag{39}$$

Then, the partial derivative of the cost function is approximated as

$$\frac{\partial E(k+1)}{\partial \beta_c(k)} \approx - \left[ \operatorname{sech}^2(e(k+1))e(k+1) + \tanh(e(k+1)) \right] \hat{g}(k)\varphi_c(k).$$

Substituting the last equation on the upgrade law (37), we obtain a feasible upgrade law

$$\beta_c(k+1) = \beta_c(k) + \eta h_e(k+1)\hat{g}(k)\varphi_c(k), \tag{40}$$

with  $h_e(k+1) \triangleq \operatorname{sech}^2(e(k+1))e(k+1) + \tanh(e(k+1))$ . Per the stability proof for Theorem 2, the upgrade law can also be stated as

$$\beta_c(k+1) = \beta_c(k) + \eta_c h_e(k+1)\operatorname{sign}\{\hat{g}(k)\}\hat{g}_{min}\varphi_c(k), \tag{41}$$

with  $|\hat{g}(k)| \leq \hat{g}_{min} \forall k \geq 0$ .

**Theorem 2.** A class of non-affine non-linear discrete-time systems (1) represented in an affine way (2) is estimated as (8) if the original system (1) follows the assumptions described in Section 2. The tracking error along with the internal signals are convergent with the system estimator according to Theorem 1, the controller (36), and the upgrade law (40) or (41), if parameters are designed following the next conditions:

- $g(k) \approx \hat{g}(k)$ ;
- $\operatorname{sign}\{g(k)\} = \operatorname{sign}\{\hat{g}(k)\}$ ;
- $|\varepsilon_h(k)| \ll 1$ ;
- $0 < \eta < \frac{1}{\hat{g}_{max}^2 \|\varphi_{cmax}\|_2^2} \leq \frac{1}{\hat{g}^2(k) \|\varphi_c(k)\|_2^2}$ .

**Proof.** To verify the convergence of the closed-loop tracking error and the convergence of the system's internal signals, the Lyapunov function is selected:

$$L(k+1) = \tilde{\beta}_c^T(k+1)\tilde{\beta}_c(k+1) + e^2(k+1). \tag{42}$$

Therefore, the Lyapunov function differentiation is calculated as

$$\Delta L(k+1) = \tilde{\beta}_c^T(k+1)\tilde{\beta}_c(k+1) - \tilde{\beta}_c^T(k)\tilde{\beta}_c(k) + e^2(k+1) - e^2(k). \tag{43}$$

Defining  $\tilde{\beta}_c(k+1)$  as the difference in the ideal weight vector  $\beta^*$  and the current iteration  $\beta(k+1)$ , it is also established that

$$\tilde{\beta}_c(k+1) = \beta_c^* - \beta_c(k+1),$$

Substituting the update law (40) in the last equation

$$\tilde{\beta}_c(k+1) = \beta_c^* - \beta_c(k) - \eta h_e(k+1)\hat{g}(k)\varphi_c(k),$$

the weight vector error is also described as

$$\tilde{\beta}_c(k+1) = \tilde{\beta}_c(k) - \eta h_e(k+1)\hat{g}(k)\varphi_c(k). \tag{44}$$

In a similar sense, considering that the ideal weight produces the ideal controller and no tracking error, it is inferred that

$$e^*(k+1) = 0 = r(k+1) - y^*(k+1) = f(k) + g(k)\beta_c^{*T}\varphi_c(k),$$

hence,

$$r(k+1) = f(k) + g(k)u^*(k).$$

Substituting the last equation in the tracking error (35),

$$e(k+1) = f(k) + g(k)\beta_c^{*T}\varphi_c(k) - f(k) - g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k),$$

is rearranged as

$$e(k+1) = g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k). \tag{45}$$

With Equations (44) and (45), the Lyapunov function differentiation (43) is rewritten as

$$\begin{aligned} \Delta L(k+1) &= \left(g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k)\right)^2 - e^2(k) \\ &\quad + \left(\tilde{\beta}_c(k) - \eta h_e(k+1)g(k)\varphi_c^T(k)\right)^2 - \tilde{\beta}_c^2(k), \end{aligned}$$

and with some mathematical processes takes the form

$$\begin{aligned} \Delta L(k+1) &= \left(g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k)\right)^2 - e^2(k) \\ &\quad - 2\eta\varphi_c^T(k)\tilde{\beta}_c(k)h_e(k+1)g(k) + \eta^2h_e^2(k+1)g^2(k)\varphi_c^2(k), \end{aligned} \tag{46}$$

and finally,

$$\begin{aligned} \Delta L(k+1) &= \left(g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k)\right)^2 - e^2(k) \\ &\quad - \eta \left[ 2\varphi_c^T(k)\tilde{\beta}_c(k)h_e(k+1)g(k) - \eta h_e^2(k+1)g^2(k)\varphi_c^2(k) \right]. \end{aligned} \tag{47}$$

From the last equation, the learning rate  $\eta$  needs to be positive definite. The term multiplied by  $\eta$  also needs to be definite positive as

$$2\varphi_c^T(k)\tilde{\beta}_c(k)h_e(k+1)g(k) - \eta h_e^2(k+1)g^2(k)\varphi_c^2(k) > 0.$$

To ensure the last inequality, the learning rate is bounded as

$$\eta < \frac{2\varphi_c^T(k)\tilde{\beta}_c(k)h_e(k+1)g(k)}{h_e^2(k+1)g^2(k)\varphi_c^2(k)}. \tag{48}$$

From (45), it is inferred that

$$\tilde{\beta}_c^T(k)\varphi_c(k) = \frac{e(k+1) - \varepsilon(k)}{g(k)},$$

then, the boundary of the learning rate becomes

$$\eta < \frac{2\{e(k+1) - \varepsilon(k)\}h_e(k+1)}{h_e^2(k+1)g^2(k)\varphi_c^2(k)}.$$

Considering that  $2e(k+1) \geq h_e(k+1)$  and  $\varepsilon(k)$  is small enough to be negligible, the learning rate boundary is rewritten as

$$\eta \leq \frac{h_e(k+1)^2}{h_e^2(k+1)g^2(k)\varphi_c^2(k)} < \frac{2\{e(k+1) - \varepsilon(k)\}h_e(k+1)}{h_e^2(k+1)g^2(k)\varphi_c^2(k)},$$

and the final boundary of the learning rate becomes

$$0 < \eta \leq \frac{1}{g_{max}^2\varphi_{cmax}^2} \leq \frac{1}{g^2(k)\varphi_c^2(k)}, \tag{49}$$

where  $g_{max} \geq |\hat{g}(k)|$  and  $\varphi_{cmax}^2 \geq \varphi_c^2(k) \forall k \geq 0$ .

In a similar sense, we can say that

$$\eta \leq \frac{\text{sign}\{g(k)\}g_{\min}}{g(k)} \frac{1}{g_{\max}^2\varphi_{c\max}^2} \leq \frac{1}{g_{\max}^2\varphi_{c\max}^2},$$

hence, if we set

$$\eta = \frac{\text{sign}\{g(k)\}g_{\min}}{g(k)} \frac{1}{g_{\max}^2\varphi_{c\max}^2},$$

and substitute it into the upgrade law (40), we obtain

$$\beta_c(k+1) = \beta_c(k) + \frac{\text{sign}\{g(k)\}g_{\min}}{g(k)} \frac{1}{g_{\max}^2\varphi_{c\max}^2} h_e(k+1)\hat{g}(k)\varphi_c(k)$$

and according to (39), it is rearranged as

$$\beta_c(k+1) = \beta_c(k) + \frac{1}{g_{\max}^2\varphi_{c\max}^2} h_e(k+1)\text{sign}\{g(k)\}g_{\min}\varphi_c(k) \tag{50}$$

If we define

$$\eta_c = \frac{1}{g_{\max}^2\varphi_{c\max}^2},$$

and substitute it in (50), we obtain the upgrade law in (41).

From the previous boundary, it is derived that the Lyapunov differentiation (46) can be rewritten as

$$\begin{aligned} \Delta L(k+1) &= \left(g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k)\right)^2 - e^2(k) \\ &\quad - 2\eta|\varphi_c^T(k)\tilde{\beta}_c(k)h_e(k+1)g(k)| + \eta^2h_e^2(k+1)g^2(k)\varphi_c^2(k). \end{aligned} \tag{51}$$

With the knowledge of the boundaries of function  $h_e(k+1)$ , it is also deduced that  $|h_e(k+1)| \leq 1.2$ . Replacing the boundaries, the last equation is rearranged as

$$\begin{aligned} \Delta L(k+1) &\leq \left(g(k)\tilde{\beta}_c^T(k)\varphi_c(k) - \varepsilon(k)\right)^2 - e^2(k) \\ &\quad - 2.4\eta|\varphi_c^T(k)\tilde{\beta}_c(k)g(k)| + 1.44\eta^2g^2(k)\varphi_c^2(k). \end{aligned}$$

For negative or positive constants  $a$  and  $b$ , the inequality  $(a+b)^2 \leq 2a^2 + 2b^2$  is always met. With that property in mind, the last equation can be rewritten as

$$\begin{aligned} \Delta L(k+1) &\leq 2g^2(k)\tilde{\beta}_c^2(k)\varphi_c^2(k) + 2\varepsilon^2(k) - e^2(k) \\ &\quad - 2.4\eta|\varphi_c^T(k)\tilde{\beta}_c(k)g(k)| + 1.44\eta^2g^2(k)\varphi_c^2(k) < 0. \end{aligned} \tag{52}$$

To ensure stability as in the previous equation, we must analyze the different term boundaries. The boundary of the tracking error  $e(k+1)$  is established from the equation

$$-e^2(k) + 1.44\eta^2g^2(k)\varphi_c^2(k) + 2\varepsilon^2(k) < 0,$$

and is defined as

$$e^2(k) > \Omega_e, \tag{53}$$

with  $\Omega_e \triangleq 1.44\eta^2g^2(k)\varphi_c^2(k) + 2\varepsilon^2(k)$ .

On the other hand, the boundary of the weight vector error  $\tilde{\beta}(k)$  is defined from the equation

$$2[g^2(k)\tilde{\beta}_c^2(k)\varphi_c^2(k) - 1.2\eta|\varphi_c^T(k)\tilde{\beta}_c(k)g(k)| + 1.44\eta^2g^2(k)\varphi_c^2(k) + 2\varepsilon^2(k)] < 0,$$

where adding and subtracting  $1.2\eta$  to and from the equation means some terms can be rearranged on a square binomial:

$$2[\sqrt{0.6\eta} - |\varphi_c^T(k)\tilde{\beta}_c(k)g(k)|]^2 + 1.44\eta^2g^2(k)\varphi_c^2(k) + 2\varepsilon^2(k) - 1.2\eta^2 < 0.$$

Isolating the term  $\tilde{\beta}_c(k)$ , it is bounded as

$$|\tilde{\beta}_c(k)|_1 > \Omega_{\beta_c}, \tag{54}$$

where  $\Omega_{\beta_c} \triangleq \frac{\sqrt{1.2\eta} + \sqrt{-1.44\eta^2g^2(k)\varphi_c^2(k) - 2\varepsilon^2(k) + 1.2\eta^2}}{\sqrt{2}|\tilde{\beta}_c(k)|_1|g(k)|}$ .

This concludes the stability proof with the boundedness of the closed-loop system’s tracking error and internal signals. The next section shows experimental results to validate the proposed controller performance. □

### 5. Validation Results

For experimentation, we proposed a Cartesian robot with the motor speed controlled by a driver (the frequency and direction as the input) and the output as the sensed force. Both the input and output are considered for the z-axis of the robot. The robot was designed at Cinvestav Saltillo. The robot uses servo-motors with a terminal voltage of 60VDC, a continuous torque of 0.353 Nm, and an incremental encoder (AMT102), controlled with a generic driver. The generic driver is connected to a computer by NI DAQ Multifunction SCB-68, which also controls the pulse generator Agilent 33220A and the power supply B&K Precision 1666. The force sensor TW Transducer 9105-TW-MINI58 is also connected to a PC with an ATI Industrial Automation 9620-05-DAQ. The control algorithm is run on MATLAB 2013, and the computer has a processor GenuineIntel 7, a RAM of 2 GB, and an integrated hard disk of 150 MB. Figure 5 shows a picture of the experimental setup, and Figure 6 shows a diagram of how the system works.

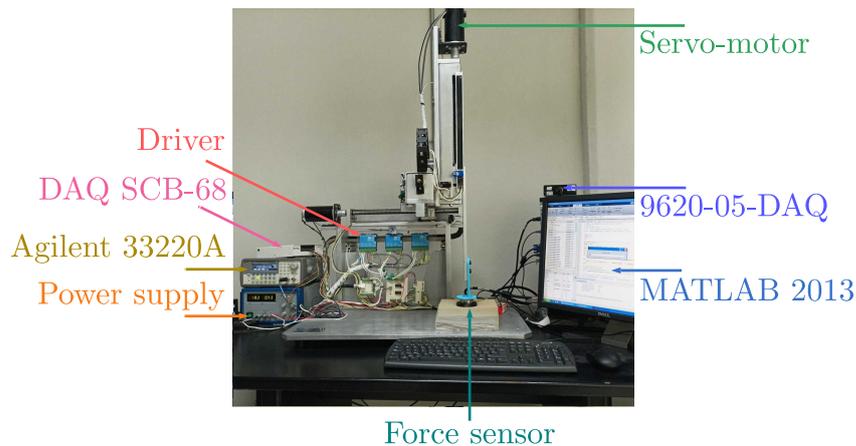


Figure 5. Cartesian robot setup picture.

For performance comparison, the controller proposed by M. L. Corradini in the article *A Robust Sliding-Mode Based Data-Driven Model-Free Adaptive Controller* [37] was replicated.

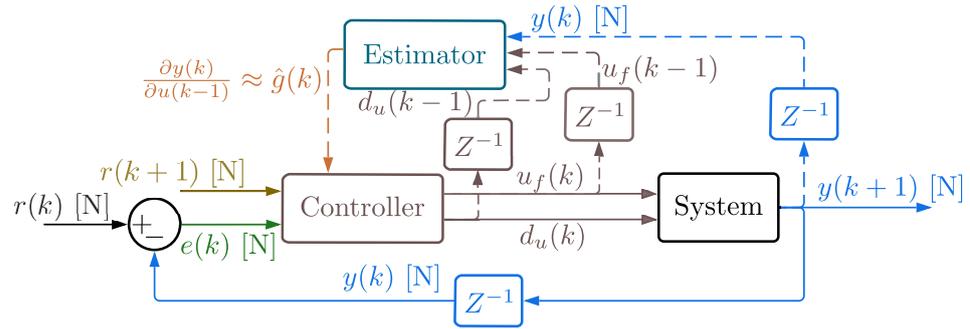


Figure 6. Cartesian robot setup diagram.

The error metrics presented are the Sum Square Error [SSE] defined as

$$SSE = \sum_{k=1}^n e(k)^2, \tag{55}$$

and the Mean Absolute Percentage Error [MAPE]

$$MAPE = \frac{1}{n} \sum_{k=1}^n \frac{|r(k) - y(k)|}{|r(k)|}. \tag{56}$$

The control algorithms are written for MATLAB 2013. Given the type of system, it is clear that the control laws need to be separated between the direction of the motor and the pulse speed; the control law information is divided as the motor direction

$$d_u(k) = \begin{cases} 1 & u(k) \geq 0, \\ 0 & u(k) < 0. \end{cases}$$

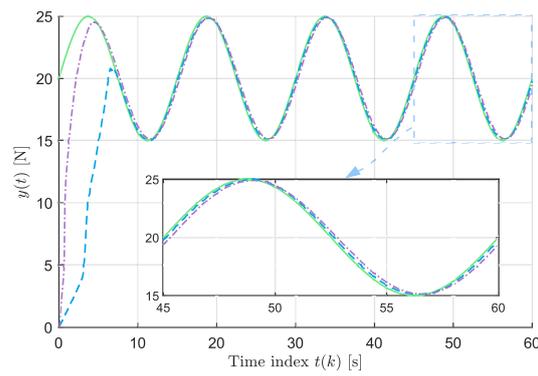
and the frequency sent to the diver

$$u_f(k) = |u(k)|.$$

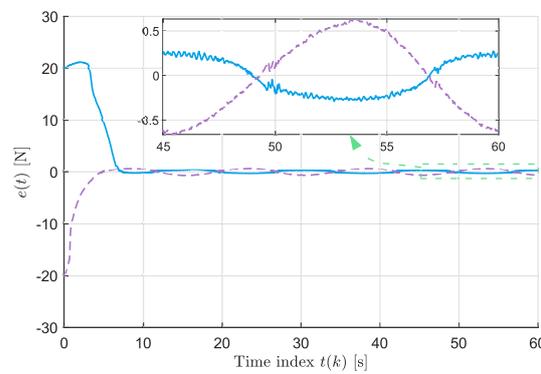
This means that if the control law equals  $u(k) = 5$ , the motor direction  $d_u(k) = 1$  will be moving to the right and the driver will be sent a pulse frequency of  $u_f(k) = 5$  kHz. On the other hand, if the control law equals  $u(k) = -5$ , the motor direction  $d_u(k) = 0$  will be moving to the left and the driver will be sent a pulse frequency of  $u_f(k) = 5$  kHz.

Figures 7–11 show the controller’s performance, and Table 1 shows the metrics results for both controllers. As is seen in both the figures and the table, the proposed controller has a more significant tracking error at the beginning than the comparison controller but has a better performance in the final cycle. The proposed controller has better performance for the initial and final cycle of the estimation. It can be seen that both controllers have high-frequency disturbance on both the estimator and the system performance, and the proposed controller has a slower settling.

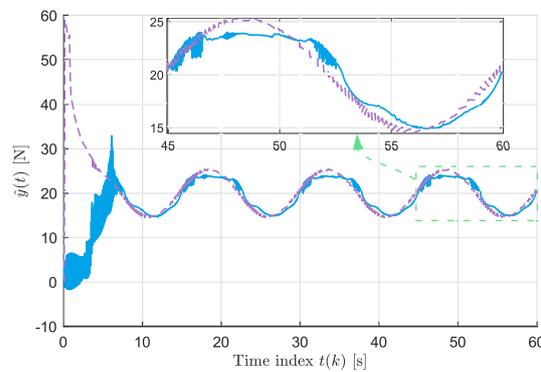
Figure 12 shows the PPD estimation  $\hat{g}(k)$  as proposed in comparison with  $\hat{\phi}_1(k)$  of the comparison controller. The PPD is usually approximated as  $\frac{\partial y(k+1)}{\partial u(k)} \approx \frac{\Delta y(k+1)}{\Delta u(k)}$  with  $\Delta y(k+1) = y(k+1) - y(k)$  and  $\Delta u(k) = u(k) - u(k-1)$ . Figure 12 shows the approximation according to the system performance and controllers. It can be seen that the high frequency in both controllers causes the estimation to be scattered. Figure 13 shows and approximation of the PPD of the system with each controller, where it can be seen the comparison controller produces more sign changes. In contrast, if we think about how the control direction and the PPD should behave with a *smooth* input to the system, as shown in Figures 14 and 15, the direction of the function  $\hat{g}(k)$  has an expected direction during the simulation, unlike the comparison controller.



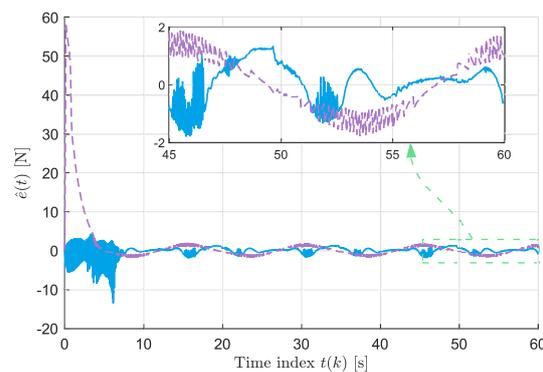
**Figure 7.** Cartesian robot system performance: [—] desired trajectory, [- -] proposed controller, and [- · -] comparison controller.



**Figure 8.** Cartesian robot tracking error: [—] proposed controller and [- -] comparison controller.



**Figure 9.** Cartesian robot system estimation: [—] proposed controller and [- -] comparison controller.



**Figure 10.** Cartesian robot estimation error: [—] proposed controller and [- -] comparison controller.

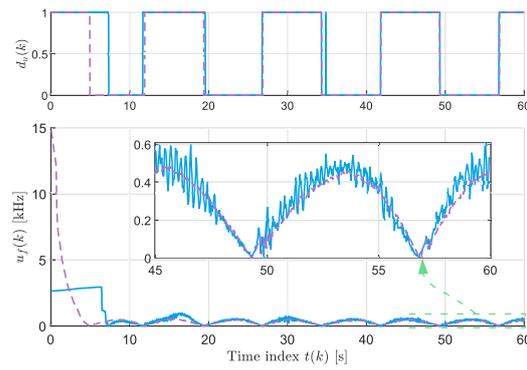


Figure 11. Cartesian robot control law: [—] proposed controller and [- -] comparison controller.

Table 1. Performance metrics—Cartesian robot.

		Proposed Controller		Comparison Controller	
		$T_i$	$T_f$	$T_i$	$T_f$
MAPE	$e(k)$	28.27%	0.98%	9.89%	2.01%
	$\hat{e}(k)$	72.21%	2.50%	163.13%	4.18%
SSE	$e(k)$	$5.53 \times 10^4$	19.71	$1.26 \times 10^4$	107.25
	$\hat{e}(k)$	$4.12 \times 10^3$	221.11	$1.04 \times 10^5$	489.34

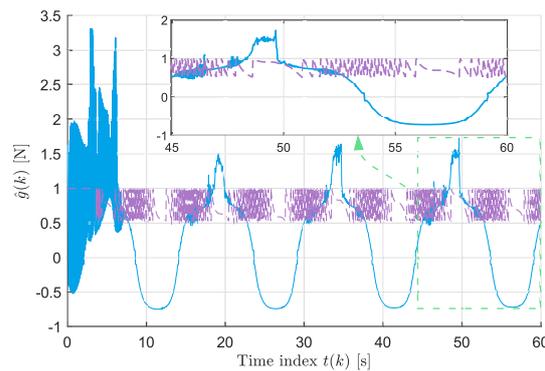


Figure 12. Cartesian robot PPD estimation: [—] proposed controller  $\hat{g}(k)$  and [- -] comparison controller  $\hat{\phi}_1(k)$ .

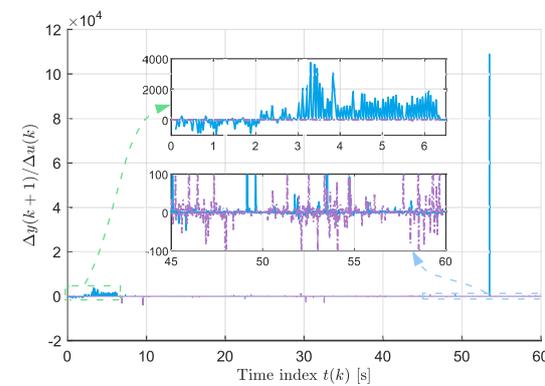
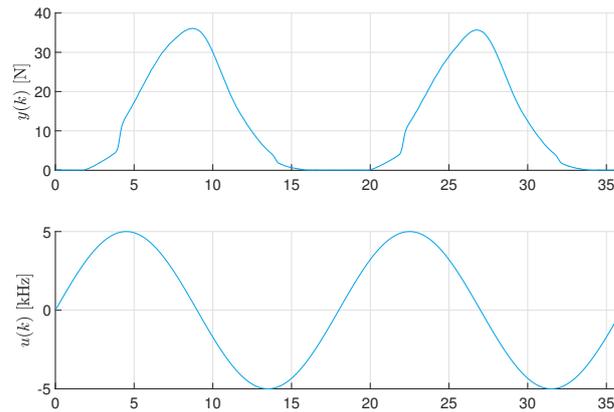
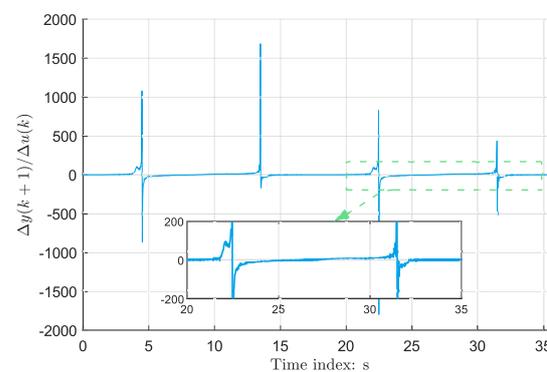


Figure 13. Cartesian robot PPD approximation  $\frac{\Delta y(k+1)}{\Delta u(k)}$ : [—] proposed controller and [- -] comparison controller.



**Figure 14.** System performance  $y(k)$  and input  $u(k)$ .



**Figure 15.** Smooth input PPD approximation  $\frac{\Delta y(k+1)}{\Delta u(k)}$ .

## 6. Conclusions

Our proposed controller and estimator address the challenges posed by unknown non-affine discrete-time systems with varying control directions. They incorporate novel cost functions, which play a crucial role in adapting to changes in the control direction and ensuring system stability. Through rigorous analysis based on Lyapunov theory, we proved the convergence of these methods, which instills confidence in their effectiveness.

Experimental validation conducted on a force-feedback control system, characterized by its time-varying control direction, demonstrates the practical utility of the proposed estimator and controller. The results illustrate the smooth and adaptive nature of the system's response to changes in the control direction, highlighting the efficacy of the proposed methods in real-world scenarios.

While the proposed controller may initially exhibit a slower response compared to state-of-the-art alternatives, its adaptive nature ultimately enables a remarkable performance. By continuously estimating and adjusting the control direction, the system achieves an impressive tracking accuracy and robustness over time. Other systems can implement the proposed estimator and controller by acquiring data for estimator training and spending some time training with the actual system. Additionally, it is worth noting that due to the MiFREN base of these methods, human knowledge can be transferred into the system through the initialization of weight vectors, which will be enhanced with the learning algorithm.

**Author Contributions:** Both authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the Mexican Research Organization CONAHCyT scholarship #811678 and was developed at Cinvestav Saltillo.

**Data Availability Statement:** The experimental code and generated data are available upon request. Please contact the corresponding author.

**Acknowledgments:** This study was developed at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Unidad Saltillo.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

DDC	Data-driven control
FREN	Fuzzy Rules Emulated Network
IFT	Iterative feedback tuning
ILC	Iterative learning control
MAPE	Mean Absolute Percentage Error
MiFREN	Multi-input Fuzzy Rules Emulated Network
PPD	Pseudo partial derivative
SSE	Sum Square Error
UUB	Uniformly ultimately bounded
VRFT	Virtual reference feedback tuning

## References

1. Ma, Y.S.; Che, W.W.; Deng, C.; Wu, Z.G. Distributed model-free adaptive control for learning nonlinear MASs under DoS attacks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 1146–1155. [[CrossRef](#)] [[PubMed](#)]
2. Liu, X.; Guo, H.; Cheng, X.; Du, J.; Ma, J. A Robust Design of the Model-Free-Adaptive-Control-Based Energy Management for Plug-in Hybrid Electric Vehicle. *Energies* **2022**, *15*, 7467. [[CrossRef](#)]
3. Chapagain, K.; Gurung, S.; Kulthnavit, P.; Kittipiyakul, S. Short-term electricity demand forecasting using deep neural networks: An analysis for Thai data. *Appl. Syst. Innov.* **2023**, *6*, 100. [[CrossRef](#)]
4. Roman, R.C.; Precup, R.E.; Petriu, E.M.; Dragan, F. Combination of data-driven active disturbance rejection and Takagi-Sugeno fuzzy control with experimental validation on tower crane systems. *Energies* **2019**, *12*, 1548. [[CrossRef](#)]
5. Barth, J.M.; Condomines, J.P.; Bronz, M.; Moschetta, J.M.; Join, C.; Fliess, M. Model-free control algorithms for micro air vehicles with transitioning flight capabilities. *Int. J. Micro Air Veh.* **2020**, *12*, 1756829320914264. [[CrossRef](#)]
6. Hou, Z.; Jin, S. *Model Free Adaptive Control: Theory and Applications*; CRC Press: Boca Raton, FL, USA, 2013.
7. Zhou, L.; Li, Z.; Yang, H.; Fu, Y.; Yan, Y. Data-driven model-free adaptive sliding mode control based on FFDL for electric multiple units. *Appl. Sci.* **2022**, *12*, 10983. [[CrossRef](#)]
8. Ahsan, M.; Salah, M.M.; Saeed, A. Adaptive Fast-Terminal Neuro-Sliding Mode Control for Robot Manipulators with Unknown Dynamics and Disturbances. *Electronics* **2023**, *12*, 3856. [[CrossRef](#)]
9. Heertjes, M.F.; Van der Velden, B.; Oomen, T. Constrained iterative feedback tuning for robust control of a wafer stage system. *IEEE Trans. Control. Syst. Technol.* **2015**, *24*, 56–66. [[CrossRef](#)]
10. Roman, R.C.; Precup, R.E.; Hedrea, E.L.; Preitl, S.; Zamfirache, I.A.; Bojan-Dragos, C.A.; Petriu, E.M. Iterative feedback tuning algorithm for tower crane systems. *Procedia Comput. Sci.* **2022**, *199*, 157–165. [[CrossRef](#)]
11. Duan, L.; Hou, Z.; Yu, X.; Jin, S.; Lu, K. Data-driven model-free adaptive attitude control approach for launch vehicle with virtual reference feedback parameters tuning method. *IEEE Access* **2019**, *7*, 54106–54116. [[CrossRef](#)]
12. Roman, R.C.; Precup, R.E.; Bojan-Dragos, C.A.; Szedlak-Stinean, A.I. Combined model-free adaptive control with fuzzy component by virtual reference feedback tuning for tower crane systems. *Procedia Comput. Sci.* **2019**, *162*, 267–274. [[CrossRef](#)]
13. Zhuang, Z.; Tao, H.; Chen, Y.; Stojanovic, V.; Paszke, W. An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 3461–3473. [[CrossRef](#)]
14. Maqsood, K.; Luo, J.; Yang, C.; Ren, Q.; Li, Y. Iterative learning-based path control for robot-assisted upper-limb rehabilitation. *Neural Comput. Appl.* **2021**, 1–13. [[CrossRef](#)]
15. Nussbaum, R.D. Some remarks on a conjecture in parameter adaptive control. *Syst. Control. Lett.* **1983**, *3*, 243–246. [[CrossRef](#)]
16. Arefi, M.M.; Zarei, J.; Karimi, H.R. Adaptive output feedback neural network control of uncertain non-affine systems with unknown control direction. *J. Frankl. Inst.* **2014**, *351*, 4302–4316. [[CrossRef](#)]
17. Mawlani, P.; Arbabtafti, M. Observer-based self-organizing adaptive fuzzy neural network control for non-linear, non-affine systems with unknown sign of control gain and dead zone: A case study of pneumatic actuators. *Trans. Inst. Meas. Control* **2022**, *44*, 2214–2234. [[CrossRef](#)]
18. Xia, J.; Zhang, J.; Feng, J.; Wang, Z.; Zhuang, G. Command filter-based adaptive fuzzy control for nonlinear systems with unknown control directions. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1945–1953. [[CrossRef](#)]

19. Cao, Y.; Zhao, N.; Xu, N.; Zhao, X.; Alsaadi, F.E. Minimal-approximation-based adaptive event-triggered control of switched nonlinear systems with unknown control direction. *Electronics* **2022**, *11*, 3386. [[CrossRef](#)]
20. Kamalamiri, A.; Shahrokhi, M.; Mohit, M. Adaptive finite-time neural control of non-strict feedback systems subject to output constraint, unknown control direction, and input nonlinearities. *Inf. Sci.* **2020**, *520*, 271–291. [[CrossRef](#)]
21. Yu, J.; Shi, P.; Lin, C.; Yu, H. Adaptive neural command filtering control for nonlinear MIMO systems with saturation input and unknown control direction. *IEEE Trans. Cybern.* **2019**, *50*, 2536–2545. [[CrossRef](#)]
22. Habibi, H.; Nohooji, H.R.; Howard, I. Adaptive PID control of wind turbines for power regulation with unknown control direction and actuator faults. *IEEE Access* **2018**, *6*, 37464–37479. [[CrossRef](#)]
23. Ren, H.P.; Wang, X.; Fan, J.T.; Kaynak, O. Adaptive backstepping control of a pneumatic system with unknown model parameters and control direction. *IEEE Access* **2019**, *7*, 64471–64482. [[CrossRef](#)]
24. Wang, S.; Fu, M.; Wang, Y. Robust adaptive steering control for unmanned surface vehicle with unknown control direction and input saturation. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1212–1224. [[CrossRef](#)]
25. Askari, M.R.; Shahrokhi, M.; Talkhonchek, M.K. Observer-based adaptive fuzzy controller for nonlinear systems with unknown control directions and input saturation. *Fuzzy Sets Syst.* **2017**, *314*, 24–45. [[CrossRef](#)]
26. Boulkroune, A.; M'saad, M. On the design of observer-based fuzzy adaptive controller for nonlinear systems with unknown control gain sign. *Fuzzy Sets Syst.* **2012**, *201*, 71–85. [[CrossRef](#)]
27. Bai, W.; Liu, P.X.; Wang, H. Neural-network-based adaptive fixed-time control for nonlinear multiagent non-affine systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 570–583. [[CrossRef](#)]
28. Wang, S.; Liu, Y.; Yu, H.; Chen, Q. Approximation-Free Control for Nonaffine Nonlinear Systems with Prescribed Performance. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 2277–2282.
29. Zhao, Y.; Niu, B.; Zong, G.; Zhao, X.; Alharbi, K.H. Neural network-based adaptive optimal containment control for non-affine nonlinear multi-agent systems within an identifier-actor-critic framework. *J. Frankl. Inst.* **2023**, *360*, 8118–8143. [[CrossRef](#)]
30. Binazadeh, T.; Rahgoshay, M.A. Robust output tracking of a class of non-affine systems. *Syst. Sci. Control Eng.* **2017**, *5*, 426–433. [[CrossRef](#)]
31. Liu, Y.J.; Wang, W. Adaptive fuzzy control for a class of uncertain nonaffine nonlinear systems. *Inf. Sci.* **2007**, *177*, 3901–3917. [[CrossRef](#)]
32. Hu, Y.; Zhang, C.; Wang, B.; Zhao, J.; Gong, X.; Gao, J.; Chen, H. Noise-Tolerant ZNN-Based Data-Driven Iterative Learning Control for Discrete Nonaffine Nonlinear MIMO Repetitive Systems. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 344–361. [[CrossRef](#)]
33. Wang, M.; Zhang, Y.; Wang, C. Learning from neural control for non-affine systems with full state constraints using command filtering. *Int. J. Control* **2020**, *93*, 2392–2406. [[CrossRef](#)]
34. Zhang, F.; Chen, Y.Y. Fuzzy adaptive output consensus tracking control of multiple nonaffine nonlinear pure-feedback systems. In Proceedings of the 2022 IEEE 17th International Conference on Control & Automation (ICCA), Naples, Italy, 27–30 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 648–653.
35. Treesatayapun, C. The knowledge-based fuzzy rules emulated network and its applications on direct adaptive on nonlinear control systems. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2005**, *13*, 653–672. [[CrossRef](#)]
36. Lewis, F.; Jagannathan, S.; Yesildirak, A. *Neural Network Control of Robot Manipulators and Non-Linear Systems*; CRC Press: Boca Raton, FL, USA, 1998.
37. Corradini, M.L. A robust sliding-mode based data-driven model-free adaptive controller. *IEEE Control Syst. Lett.* **2021**, *6*, 421–427. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.