

Proceeding Paper

# Online Adaptation of a Compensatory Neuro-Fuzzy Controller Parameters Using the Extended Kalman Filter: Application on an Inverted Pendulum <sup>†</sup>

Hocine Khati <sup>\*</sup>, Hand Talem, Mohand Achour Touat, Rabah Mellah and Said Guermah

Design and Drive of Production Systems Laboratory, Department of Automation, Faculty of Electrical and Computing Engineering, University Mouloud Mammeri of Tizi-Ouzou, Tizi-Ouzou 15000, Algeria; talemhand2015@gmail.com (H.T.); mohand\_touat@yahoo.fr (M.A.T.); mellah.rab@gmail.com (R.M.); saidguermah@yahoo.fr (S.G.)

<sup>\*</sup> Correspondence: hoc.khati@gmail.com or hocine\_khati@yahoo.fr; Tel.: +213 556-83-24-73

<sup>†</sup> Presented at the 1st International Conference on Computational Engineering and Intelligent Systems, Online, 10–12 December 2021.

**Abstract:** This paper presents the implementation of a Compensatory Adaptive Neuro-Fuzzy Inference System (CANFIS) controller to control an inverted pendulum. This controller is developed in order to readjust the parameters relating to the membership functions and the fuzzy rules used as well as to optimize the dynamics of the latter, using a learning algorithm based on the extended Kalman filter. The CANFIS controller is developed on the Simulink environment of the MATLAB software and is implemented on a Raspberry Pi 3 board, with a view to analyzing its real behavior, and testing its speed as well as its robustness through the use of the “Processor-In-the-Loop” (PIL) technique. The results obtained through PIL tests showed the effectiveness of the neuro-fuzzy controller equipped with a compensator.

**Keywords:** CANFIS; extended Kalman filter; Raspberry Pi 3



**Citation:** Khati, H.; Talem, H.; Touat, M.A.; Mellah, R.; Guermah, S. Online Adaptation of a Compensatory Neuro-Fuzzy Controller Parameters Using the Extended Kalman Filter: Application on an Inverted Pendulum. *Eng. Proc.* **2022**, *14*, 11. <https://doi.org/10.3390/engproc2022014011>

Academic Editors: Abdelmadjid Recioui, Hamid Bentarzi and Fatma Zohra Dekhandji

Published: 29 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

The control of non-linear systems is always a rather complex task in the field of automation, because several control techniques require a perfect knowledge of the behavior of the system in order to be able to achieve good performance. The description of a system equipped with its actuators and sensors and the various physical phenomena that appear during operation show the high complexity due to numerous non-linearities as well as the difficulty in perfectly modeling the dynamics of the mechanical sensors-actuators-system. Controlling the system for performing tasks requiring good performance becomes difficult, as knowledge of the system is incomplete and imperfect.

Adaptive control using artificial intelligence techniques has been the subject of several research works in the control of nonlinear systems [1–4]. The combination of fuzzy logic with neural networks has been recognized as a promising solution to compensate for the non-structural uncertainties of nonlinear systems. Therefore, combining fuzzy logic with neural networks in the form of a single network structure allows building more efficient neuro-fuzzy controllers, and this by taking advantage of the inference capacity of fuzzy reasoning and computational parallelism of neural networks. Among the most efficient adaptive neuro-fuzzy systems, we cite the CANFIS (Compensatory Adaptive Neuro-Fuzzy Inference System) structure, which consists of using a fuzzy compensator which compensates for the bad choice of membership functions, and which uses techniques which allow us to adjust the dynamics of fuzzy rules, so as to adapt to the environment [5]. However, the use of such an adaptive structure requires a high computation time and its implementation in practice then requires powerful control boards in order to minimize the computational time of the algorithm in order to converge on the desired performance.

Due to the computational complexity of these algorithms, several processes are generally necessary so that the controllers are suitable for implementations, because on the one hand, many development boards do not support the double-precision floating-point data type on the MATLAB-Simulink software, and on the other hand, designs based on this floating-point data type consume a significant number of hardware resources compared to designs based on the fixed-point data type [6]. However, in this design methodology, the adaptation of the controller is limited by the fixed point encoding. Several research works have been carried out in this area [7–10]. In this article, we will use MATLAB-Simulink software to implement the CANFIS controller on a Raspberry Pi 3 board to control an inverted pendulum. These control boards are known by their powerful processors, namely the ARM Cortex processors, and support both fixed and floating point data types. The MATLAB software therefore allows us to considerably reduce the design time of the algorithm, and to implement it on the Raspberry board. The use of the Processor-In-the-Loop (PIL) technique makes it possible to test the behavior of the CANFIS controller on the hardware, while the rest of the control loop is on the Simulink.

This document is organized as follows: in the second part, we present the structure of the fuzzy neuro network with a fuzzy compensator as well as the mathematical formalism, then, we study the learning algorithm. In the third part, we present the control scheme based on the Raspberry Pi 3 board, suitable for dynamic control of the inverted pendulum using the CANFIS controller. In the fourth part, we present the results of the PIL tests obtained by applying this synthesized strategy to the command in tracking of a reference trajectory of the inverted pendulum. Finally, we end our work with a general conclusion and some perspectives.

## 2. Description of the Adaptive Neuro-Fuzzy Network with a Compensator

CANFIS is a fuzzy inference system based on a multi-layered adaptive network, where the learning algorithm not only adjusts the membership functions, but also optimizes the dynamics of fuzzy reasoning by adjusting the degree of compensation. In this section, we develop an ANFIS controller equipped with a fuzzy compensator, where the adaptation of its parameters is ensured by a learning algorithm based on the extended Kalman filter. To simplify understanding and without loss of generality, we consider a system with two inputs  $x_1$ ,  $x_2$  and one output  $u$  (Figure 1), modeled by a fuzzy system of the Takagi–Sugeno type, composed of the following four fuzzy rules:

$$\text{Rule 1 : if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } u_1 = f_1(x_1, x_2) = a_1x_1 + b_1x_2 + c_1 \quad (1)$$

$$\text{Rule 2 : if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_2 \text{ then } u_2 = f_2(x_1, x_2) = a_2x_1 + b_2x_2 + c_2 \quad (2)$$

$$\text{Rule 3 : if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_1 \text{ then } u_3 = f_3(x_1, x_2) = a_3x_1 + b_3x_2 + c_3 \quad (3)$$

$$\text{Rule 4 : if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } u_4 = f_4(x_1, x_2) = a_4x_1 + b_4x_2 + c_4 \quad (4)$$

The degree of activation of a rule is defined by  $w_i$ .

We consider the pessimistic and optimistic operations, given respectively as follows:

$$z_i = w_i \quad (5)$$

$$m_i = (w_i)^{\frac{1}{2}}. \quad (6)$$

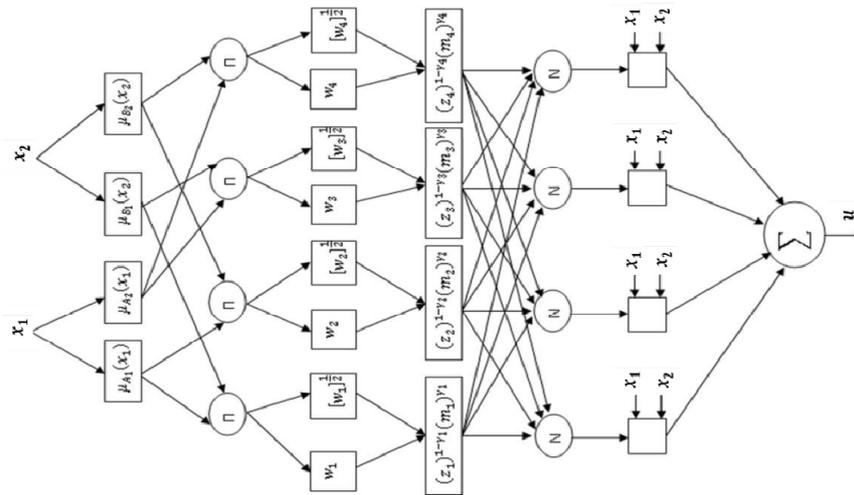
The fuzzy compensator expressed by this relation:

$$C_i(z_i, m_i, \gamma_i) = (z_i)^{1-\gamma_i} (m_i)^{\gamma_i}, \quad (7)$$

where  $\gamma_i \in [0, 1]$  is the compensatory degree. Therefore, the net value of the fuzzy neural-network equipped with a fuzzy compensator is given as follows:

$$u = \frac{\sum_{i=1}^4 (a_i e + b_i \Delta e + c_i) (w_i)^{\alpha_i}}{\sum_{i=1}^4 (w_i)^{\alpha_i}} \tag{8}$$

$$\alpha_i = 1 - \frac{\gamma_i}{2} \tag{9}$$



**Figure 1.** Structure of the Adaptive Neuro-Fuzzy Inference System Controller with compensator.

Consider that  $x_1$  and  $x_2$  are the position error  $e$  and its derivative  $\Delta e$ . We associate two fuzzy sets for each of the inputs, namely  $N$  (Negative) and  $P$  (Positive).  $\mu_N$  et  $\mu_P$  represent the appropriate membership degrees of the variables  $x_i$  with respect to fuzzy subsets  $A_i$  and  $B_i$ , defined by the following member functions:

$$\mu_N(x_1) = \exp\left(-\left(\frac{x_1 - \rho_1}{\sigma_1}\right)^2\right) \tag{10}$$

$$\mu_P(x_1) = \exp\left(-\left(\frac{x_1 - \rho_2}{\sigma_2}\right)^2\right) \tag{11}$$

$$\mu_N(x_2) = \exp\left(-\left(\frac{x_2 - \beta_1}{\delta_1}\right)^2\right) \tag{12}$$

$$\mu_P(x_2) = \exp\left(-\left(\frac{x_2 - \beta_2}{\delta_2}\right)^2\right), \tag{13}$$

with:  $(\rho_i, \sigma_i)$  and  $(\beta_i, \delta_i)$  being the (mean, variance) of the fuzzy set of the membership functions respectively of  $x_1$  and  $x_2$ .

*Learning Algorithm*

The controller is characterized by a vector of parameters  $\Phi$ . Our objective is to find the values of the vector  $\hat{\Phi}$  by minimizing the following criterion [11]:

$$J(k) = \frac{1}{2} (y_d(k) - y(k))^2 = \frac{1}{2} (e(k))^2, \tag{14}$$

where:  $y(k)$  is the real output of the system and  $y_d(k)$  is its desired output. The extended Kalman filter approach consists of linearizing the output regulator  $u_d$  at each step  $k$  around the estimated vector  $\Phi_d(k)$ . This is equivalent to writing:

$$u_d(k) = u(k) + \Psi(k)(\Phi_d(k) - \Phi(k)). \tag{15}$$

With:

$$\Psi(k) = \frac{\partial u(k)}{\partial \Phi(k)} = \left[ \frac{\partial u(k)}{\partial \Phi_1(k)}, \dots, \frac{\partial u(k)}{\partial \Phi_n(k)} \right], \tag{16}$$

where  $n$  is the dimension of the vector  $\Psi^T$ . Consequently, the parameters are adjusted according to the following relation [12]:

$$\Phi(k + 1) = \Phi(k) + K'(k)\Psi^T(k)e_u \tag{17}$$

$$K'(k) = \frac{P(k)}{\Psi(k)P(k)\Psi^T(k) + B(k)} \tag{18}$$

$$P(k + 1) = P(k) - K(k)\Psi(k)P(k) \tag{19}$$

$$K(k) = K'(k)\Psi^T(k), \tag{20}$$

where  $\Psi(k)$  is the Jacobian matrix (the observation matrix of the system),  $P(k)$  is the estimation matrix of the error covariance and  $B(k)$  is the measurement covariance matrix. In order to approximate the variation  $e_u$ , we linearize the inverse model of the system around  $y(k)$  according to the following relation:

$$u = \frac{\partial u(k)}{\partial y(k)}(y - y(k)) + u(k), \tag{21}$$

where  $y(k)$  is the system output and  $u(k)$  its input. For this linearized model, the value  $u_d(k)$  can be expressed as follows:

$$u_d(k) = \frac{\partial u(k)}{\partial y(k)}(y_d(k) - y(k)) + u(k). \tag{22}$$

Therefore, we have:

$$u_d(k) - u(k) = \frac{\partial u(k)}{\partial y(k)}(y_d(k) - y(k)). \tag{23}$$

Then:

$$e_u(k) = \frac{\partial u(k)}{\partial y(k)}e(k). \tag{24}$$

We also have:  $\partial e(k) = \partial y_d(k) - \partial y(k)$ . We assume that  $y_d(k)$  is constant in the neighborhood of  $y(k)$ . Then:

$$\partial e(k) = -\partial y(k). \tag{25}$$

Therefore:

$$e_u(k) = \frac{-\partial(k)}{\partial e(k)}e(k). \tag{26}$$

The relation (17) can be written as follows:

$$\Phi(k + 1) = \Phi(k)K'(k)\Psi^T(k)\frac{\partial u(k)}{\partial e(k)}e(k). \tag{27}$$

In order to eliminate the constraint  $\gamma_i \in [ 0 \ 1 ]$  defined in Equation (7), we define:

$$\gamma_i = \frac{p_i^2}{p_i^2 + q_i^2}. \tag{28}$$

Consequently, the parameters vector to be readjusted is given by:

$$\Phi_i^T(k) = [a_1, b_1, c_1, p_1, q_1, \dots, a_4, b_4, c_4, p_4, q_4] \tag{29}$$

$$\Psi(k) = \left[ \frac{\partial u(k)}{\partial a_1}, \frac{\partial u(k)}{\partial b_1}, \frac{\partial u(k)}{\partial c_1}, \frac{\partial u(k)}{\partial p_1}, \frac{\partial u(k)}{\partial q_1}, \dots, \frac{\partial u(k)}{\partial a_4}, \frac{\partial u(k)}{\partial b_4}, \frac{\partial u(k)}{\partial c_4}, \frac{\partial u(k)}{\partial p_4}, \frac{\partial u(k)}{\partial q_4} \right] \tag{30}$$

With:

$$\frac{\partial u}{\partial a_i} = \frac{e[w_i]^{\alpha_i}}{\sum_{i=1}^4 [w_i]^{\alpha_i}} \tag{31}$$

$$\frac{\partial u}{\partial b_i} = \frac{\Delta e[w_i]^{\alpha_i}}{\sum_{i=1}^4 [w_i]^{\alpha_i}} \tag{32}$$

$$\frac{\partial u}{\partial c_i} = \frac{[w_i]^{\alpha_i}}{\sum_{i=1}^4 [w_i]^{\alpha_i}} \tag{33}$$

$$\frac{\partial u}{\partial \gamma_i} = -\frac{1}{2} \left[ \sum_{i=1}^4 a_i e + b_i \Delta e + c_i \right] \frac{z_i \ln(w_i)}{\sum_{i=1}^4 z_i} \tag{34}$$

$$\frac{\partial u}{\partial p_i} = - \left[ \frac{2p_i(q_i)^2}{(p_i)^2 + (q_i)^2} \right] \frac{\partial u}{\partial \gamma_i} \tag{35}$$

$$\frac{\partial u}{\partial q_i} = \left[ \frac{2q_i(p_i)^2}{(p_i)^2 + (q_i)^2} \right] \frac{\partial u}{\partial \gamma_i}. \tag{36}$$

We have:

$$u = \frac{\sum_{i=1}^4 (a_i e + b_i \Delta e + c_i) (w_i)^{\alpha_i}}{\sum_{i=1}^4 (w_i)^{\alpha_i}} = \frac{h}{g}. \tag{37}$$

Then:

$$\frac{\partial u}{\partial e} = \frac{\frac{\partial h}{\partial e} \cdot g - \frac{\partial g}{\partial e} \cdot h}{g^2}. \tag{38}$$

With:

$$\frac{\partial h}{\partial e} = \sum_{i=1}^4 ([a - 2(e - \rho_j)(a_i e + b_i \Delta e + c_i)] w_i^{\alpha_i}) \tag{39}$$

$$\frac{\partial g}{\partial e} = -2 \sum_{i=1}^4 ((e - \rho_j) w_i^{\alpha_i}), \tag{40}$$

where  $j$  is the integer part of  $(i - 1)/2$ .

### 3. Implementation of the CANFIS Controller Equipped with a Compensator on the Raspberry Pi 3 Board: Application on the Inverted Pendulum

In this section, we use the “Processor-In-the-Loop” technique in MATLAB-Simulink, which allows us to test the behavior of the CANFIS controller on the Raspberry Pi 3 board [13]. In this mode of execution, the CANFIS controller runs in the hardware while the

rest of the closed loop runs on the Simulink environment. In order to test the performances of the controller, we considered the stabilization problem of an inverted pendulum on a cart. The dynamic equations of the nonlinear system are given by [14]:

$$\dot{x}_1 = x_2 \tag{41}$$

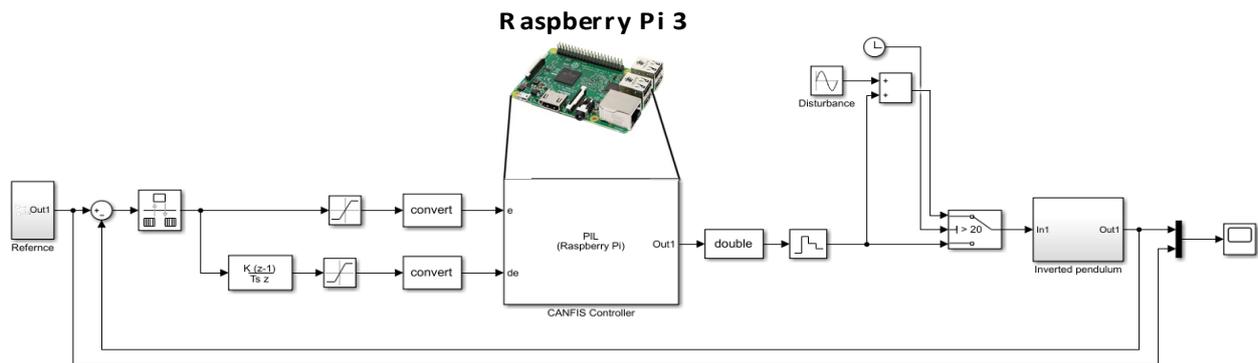
$$\dot{x}_2 = \frac{g \sin x_1 - \cos x_1 \left( \frac{ml}{m_c+m} x_2^2 \sin x_1 - \frac{1}{m_c+m} \tau(t) \right)}{\frac{4}{3}l - \frac{ml}{m_c+m} \cos^2 x_1}, \tag{42}$$

where  $x_1$  is the position of the vertical rod (in radian),  $x_2$  is the angular velocity and  $\tau(t)$  the torque applied to the rod. Its parameters are given in the Table 1.

**Table 1.** Inverted pendulum parameters.

$g$	Gravity constant	9.8 m/s <sup>2</sup>
$m_c$	Carriage mass	1 kg
$m$	Pendulum mass	0.1 kg
$l$	Rod length	0.5 m

The implementation of the CANFIS controller was done with a frequency of 10 KHz. The execution diagram of the regulation chain (CANIFIS controller, System and disturbance) in PIL mode under the MATLAB Simulink environment is presented in Figure 2.



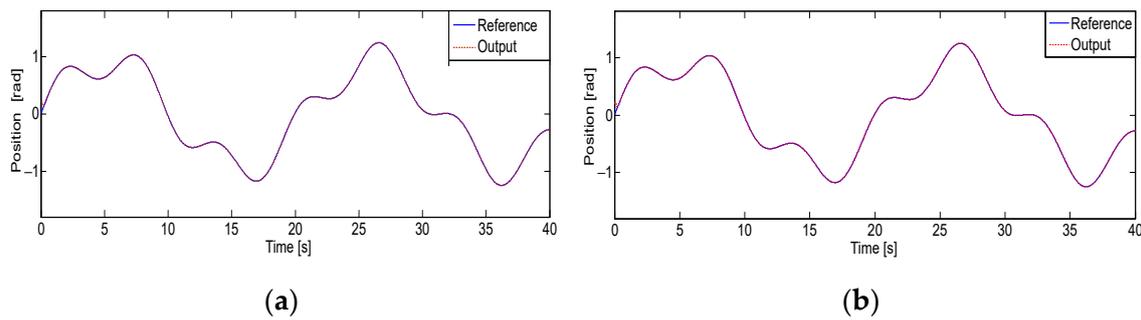
**Figure 2.** Execution diagram in PIL mode under Simulink.

#### 4. Results of the PIL Tests

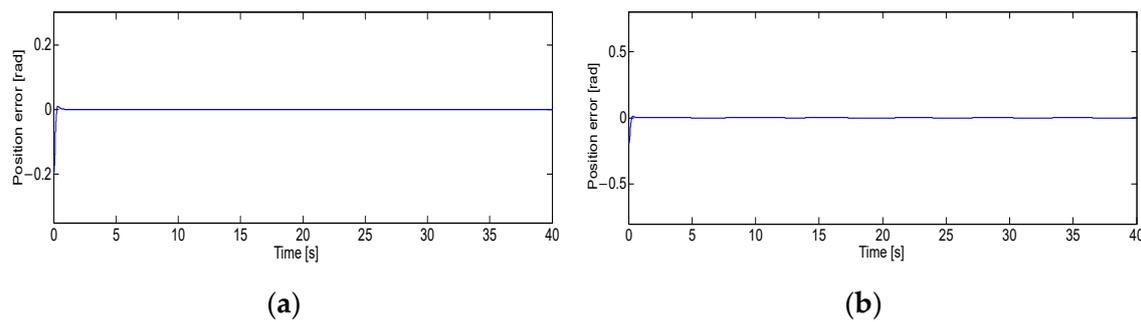
In this section, we present the simulation results of the PIL mode. In order to test the robustness of the two proposed controllers, we added a disturbance on the input of the system at the instant  $t = 20$  s, given as follows:

$$D(t) = 150 \sin(t). \tag{43}$$

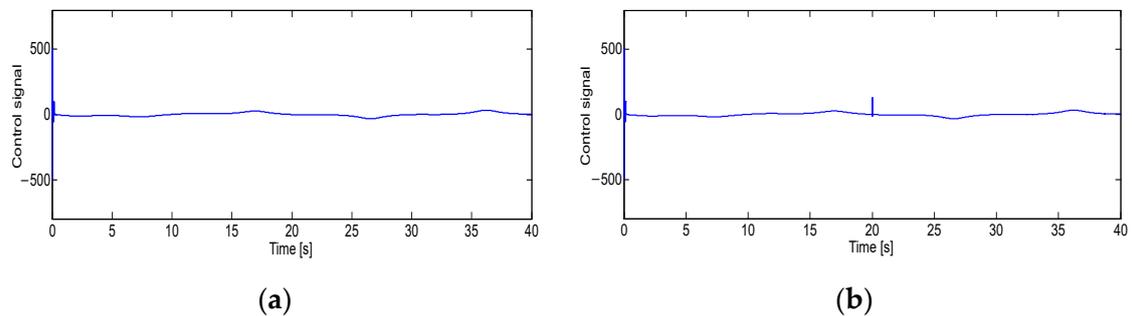
Figure 3a,b, Figure 4a,b and Figure 5a,b represent the results of the position tracking, the position tracking errors as well as the control signals delivered by the CANFIS controller. From these figures, we can see that the proposed controller has good performance (tracking, response time and robustness) with a low precision error, in both test cases (with and without disturbances). This is due to the use of a compensator which compensates for errors in the choice of membership functions, even readjusting the dynamics of fuzzy rules, using a learning algorithm based on the extended Kalman filter. The implementation of the proposed controller allowed us to obtain good performances thanks to the sampling frequency of the Raspberry Pi 3 board.



**Figure 3.** Position tracking: (a) Without disturbances; (b) With disturbances.



**Figure 4.** Position error: (a) Without disturbances; (b) With disturbances.



**Figure 5.** Control signal: (a) Without disturbances; (b) With disturbances.

## 5. Conclusions

In this article, we have proposed the implementation on a Raspberry Pi 3 board of a neuro-fuzzy controller CANFIS, in order to control an inverted pendulum. The CANFIS controller presents good results because it uses a learning algorithm based on the extended Kalman filter, which converges very quickly, and which readjusts the membership functions and optimizes at the same time the dynamics of fuzzy reasoning by adjusting the degree of compensation. The development and implementation of the controller was done through MATLAB's Simulink environment, which simplified the study and implementation of the code on the Raspberry Pi 3 board.

In conclusion, we have tried through this article to show the efficiency of the CANFIS controller on real hardware (Raspberry Pi 3) in the control of non-linear systems. The implementation of such a controller using the PIL technique has demonstrated its speed and robustness to disturbance rejection.

**Author Contributions:** Conceptualization, H.K. and H.T.; methodology, H.K., H.T. and R.M.; software, H.K. and H.T.; validation, H.K., H.T. and R.M.; formal analysis, H.K., H.T., M.A.T. and R.M.; investigation, M.A.T. and R.M.; resources, M.A.T.; data curation, H.K. and H.T.; writing—original draft preparation, H.K., H.T., R.M. and S.G.; writing—review and editing, H.K., H.T. and S.G.;

visualization, H.K. and H.T.; supervision, H.K. and H.T.; project administration, H.K.; funding acquisition, H.K., H.T., M.A.T. and R.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** No Institutional Review Board.

**Informed Consent Statement:** Patient consent was waived.

**Data Availability Statement:** Study did not report any data.

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

## References

1. Gil, P.; Oliveira, T.; Palma, L. Adaptive neuro-fuzzy control for discrete-time nonaffine nonlinear system. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1602–1615. [[CrossRef](#)]
2. Sarhadi, P.; Rezaie, B.; Rahmani, Z. Adaptive predictive control based on adaptive neuro fuzzy inference system for a class of nonlinear industrial processes. *J. Taiwan Inst. Chem. Eng.* **2016**, *61*, 132–137. [[CrossRef](#)]
3. Song, S.; Zhang, B.; Song, X.; Zhang, Z. Adaptive neuro fuzzy backstepping dynamic surface control for uncertain fractional-order nonlinear systems. *Neurocomputing* **2019**, *360*, 172–184. [[CrossRef](#)]
4. Song, S.; Park, J.H.; Zhang, B.; Song, X.; Zhang, Z. Adaptive command filtered neuro-fuzzy control design for fractional-order nonlinear systems with unknown control directions and input quantization. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 7238–7249. [[CrossRef](#)]
5. Mellah, R.; Khati, H.; Talem, H.; Guermah, S. Compensatory of adaptive neural fuzzy inference system. In *Fuzzy Systems*; IntechOpen: London, UK, 2021.
6. Finnerty, A.; Ratigner, H. *Reduce Power and Cost by Converting from Floating Point to Fixed Point*; Xilinx All Programmable, v. 1.0; Xilinx, Inc.: San Jose, CA, USA, 2017; pp. 1–14.
7. Karakuzu, C.; Karakaya, F.; Çavuşlu, M.A. FPGA implementation of neuro-fuzzy system with improved PSO learning. *Neural Netw.* **2016**, *79*, 128–140. [[CrossRef](#)] [[PubMed](#)]
8. Khati, H.; Mellah, R.; Talem, H. Neuro-fuzzy control of a position-position Teleoperation system using FPGA. In Proceedings of the 24th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 26–29 August 2019; pp. 64–69.
9. Khati, H.; Talem, H.; Mellah, R.; Bilek, A. Neuro-fuzzy control of bilateral teleoperation system using FPGA. *Iran. J. Fuzzy Syst.* **2019**, *16*, 17–32.
10. Dorzhigulov, A.; Bissengaliuly, B.; Spencer, B.F.; Kim, J. ANFIS based quadrotor drone altitude control implementation on Raspberry Pi platform. *Civ. Environ. Eng.* **2018**, *95*, 335–345. [[CrossRef](#)]
11. Mellah, R.; Guermah, S.; Toumi, R. Adaptive control of bilateral teleoperation system with compensatory neural-fuzzy controllers. *Int. J. Control Autom. Syst.* **2017**, *15*, 1–11. [[CrossRef](#)]
12. Haykin, S. *Kalman Filtering and Neural Networks*; John Wiley and Sons Inc.: Hoboken, NJ, USA, 2001.
13. Simulink Support Package for Raspberry Pi Hardware. *User's Guide*; MathWorks, MATLAB and SIMULINK, r2021a; The MathWorks Inc.: Natick, MA, USA, 2021.
14. Yoo, B.; Ham, W. Adaptive fuzzy sliding mode control of nonlinear system. *IEEE Trans. Fuzzy Syst.* **1998**, *6*, 315–321.