Int. J. Nonlinear Anal. Appl. 16 (2025) 6, 59–70
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2024.32466.4832

Online neuro-inverse dynamics controller for nonlinear induction furnace system: Fault hiding approach

Narges Torabi[a], Reza Ghasemi[b,*]

[a] Isfahan University of Technology (IUT), Isfahan, Iran
[b] University of Qom, Qom, Iran

(Communicated by Haydar Akca)

Abstract

In this paper, an online neural inverse controller is used to deal with actuator faults. In such a way that the inverse of the nonlinear induction furnace system (IFS) is used as a fault-tolerant controller (FTC) so that it can cover the fault of the actuator. The design is such that an online neural network is used to model the NIFC, the three-layer neural network is converted into a four-layer RBF neural network, and the last layer is the nonlinear IFS, and this layer is It is unchangeable and the controller and the system are connected and finally form a four-layer neural network. So, an intelligent inverse model of the IFS is used as FTC to cover the actuator fault of the nonlinear IFC. This controller design is done in two ways: in the first part, five inputs are used for training the neural network, one of which is the neural network training error, but in the second part, in addition to the five inputs of the first part, the derivative of the error is used. And the error integral has also been used in neural network training and the advantage of the second plan is to reduce overshoot. Finally, a fault actuator is applied to the nonlinear IFS in the 10th to the 30th second, despite the presence of the intelligent FTC, this defect is covered in less than one second, and the system continues to function normally despite the operator's defect in this interval of time.

Keywords: Inverse Neural Control, Fault-Tolerant Control, Induction Furnace, RBF Neural Network
2020 MSC: 68T07, 78M32

1 Introduction

In today's industry, the main requirements for any system, in addition to good performance and efficiency, are availability, reliability, and safety for that system. Therefore, the monitoring tasks in the control unit of any system have a high priority in achieving these goals [3, 18]. These components include the detection method, fault isolation, and the use of fault tolerant control(FTC) to maintain good performance in the fault state, and in fact, fault has become a more interesting topic for modern safety and technological systems in the last decade [38]. In fact, FTC is of high importance for every aspect of systems, this issue specifically prevents defects from turning into failure [7, 4]. Where the increasing demand for more reliable performance necessitates the development of sophisticated techniques to provide timely and accurate fault diagnosis and fault tolerance [3, 8]. Normally, a complete system consists of three parts: the actuator, the main structure, and the sensor, so a fault monitoring system must be specifically designed

*Corresponding author
Email addresses: n.torabi@ec.iut.ac.ir (Narges Torabi), r.ghasemi@qom.ac.ir (Reza Ghasemi)

Received: November 2023 *Accepted:* March 2024

to monitor and fix the fault of each sub-system [34] so that any fault Before it leads to disastrous consequences. In terms of fault diagnosis, our interest is mainly focused on checking actuators and sensors. Because these subsystems are more prone to faults, where their healthy function usually guarantees a smoother and more reliable operation for the system [34, 10]. Existing FTC techniques are divided into two categories, active approach and passive approach [13], where the passive fault-tolerant controller (PFTC) parameters are chosen very conservatively [6, 5] so that the desired system may be in The nominal state loses its function. On the other hand, the active fault tolerant controller (AFTC) is resistant to the fault and is reconfigured according to the fault information provided by the fault detection modules [25, 4]. As mentioned in [12], fault detection and isolation schemes have been applied to the nonlinear system of gas turbines, based on two observers to detect and isolate the actuator fault. Likewise, a nonlinear control approach is used to predict the future behavior of satellite systems to prevent faults/failures that may affect their thrusters [26].

Fault detection methods can be classified into two groups based on hardware redundancy and analytical redundancy, especially the one that is more popular in practice because it does not require additional hardware [10]. Data-driven, model-based methods are examples of analytical redundancy methods. Data mining methods are usually performed using techniques such as fuzzy logic [18], artificial neural networks [13] and genetic algorithms, etc. Neural networks (NN) can be used to automate the operation of high-speed trains. Unexpected disturbances, faults, and nonlinearities are considered by applying a neuro-adaptive fault-tolerant control scheme. As a result, velocity and position tracking are improved despite uncertainties [14]. In addition, a model-based fault accommodation system based on recurrent wavelet Elman neural networks has been developed to estimate the uncertainties and disturbances in the control of a robotic arm during locomotion. Wavelet functions are used as activation functions for hidden neurons [17]. Inverse neural network models are the next evolution. As controllers, they offer appropriate parameters to achieve a particular target input at their output. A special type of neural network is called a Radial Basis Function (RBF). An RBF neural network is used to approximate the nonlinear function and compensate for the uncertain part of the system. In addition, the Lyapunov stability theory is used to adapt the weight of the RBF neural network. The approach controls a six-degree-of-freedom robotic arm [16] and a permanent magnet motor drive [2]. The authors in [32] developed an improved model of predictive power control (MPPC) for fault-tolerant converter operation connected to the grid. In an inverse controller's induction furnace system, the inverse model is derived using an RBF neural network offline and sliding mode. Due to the Chattering effects of this type of controller [28], the intelligent online inverse for nonlinear induction furnace systems is presented.

It is therefore important to design a controller that meets the intended purpose. A nonlinear inverse control strategy can provide an intelligent fault-tolerant control method. An inverse model of a similar system is used as a nonlinear controller [9, 35]. In addition to their ability to deal with nonlinear systems, artificial neural networks can also learn complex nonlinear relationships, making them useful tools [1]. An intelligent system determines the exact orientation angle for solar sun trackers [23]. In the hot rolling of steel quality control, inverse mapping is used between the output and input of the NN. The inverse fault-tolerant controller consists of a reconfigurable block and a detector unit. Detector units calculate the differences between system outputs and model outputs. When the error exceeds the standard limit, the reconfiguration section is enabled. From the controller's point of view, the reconfiguration section ensures that the system is fault-free. Therefore, it makes the system resistant to this flaw and hides it. Therefore, using a neural network for inverse system modeling is appropriate for this control strategy.

This strategy is often needed for nonlinear induction furnaces with actuators, sensors, or other parts of the structure that heated parts may damage (melting, milling, etc.). An induction furnace consists of a coil, melting shell, and power supply. Current flows through the coil, which is the primary winding, causing current to flow into the coreless melting shell, which is the secondary coil. As a result, the raw material melts [28]. Induction furnaces operate by passing a current through a coil, which creates a magnetic field in the coil core. As a result, heat is generated. Thus, the induction furnace comprises two parts, electric and thermal. The electrical part of the furnace includes the burner, the inverter, and the capacitive bank, while the thermal part includes the coil and the workpiece [28]. Induction furnace temperature control is an important issue in the industry. Therefore, PID controllers are used to control the temperature of induction furnaces. RBF neural networks are used to determine the parameters of this PID controller [37]. The temperature of the induction furnace is controlled with fuzzy logic [36]. The PID controller has been applied to control the temperature of the induction furnace system [27]. The PI controller has been developed in three modes: 1) linear PI for the linear model, 2) linear PI for the saturated linear model, and 3) anti-wind-up PI control for the saturated linear model [22]. In 2016, fuzzy control was used [21] and in the same year, PID control based on BP neural networks was implemented [11]. In the study by Torabi et al. [28], the actuator defect is examined in a nonlinear induction furnace system that uses a slip-mode controller for fault-tolerant control. The operator's fault is considered an indefinite one in this study. The inverse model of the system is also used in the control discussion, which is trained offline using an RBF neural network. However, the amount of output error is significant due to the modeling error and the presence of chatting in the sliding mode controller. In this paper, RBF neural network training has been applied online to reduce modeling errors. The inverse system model has been used as a fault-tolerant controller to minimize output errors and provide the fault-tolerant control of a faulty system. With this approach, the faulty system tracks its ideal value, and consequently, the tracking error tends to be near zero quickly.

Authors in [29] deal with detection of similarity in Carbon Fiber-Reinforced Polymer Plate Defects based on fuzzy systems. [20] develops similarity in Profiled Temporal Association Pattern Mining based on both fuzzy systems and prevalence estimation. Fuzzy similarity measurement and their applications are develop in [30]. [33] deals with Fuzzy-Wavelet Neuro Adaptive controller to apply on Multivariable Servo Actuator in Real-Time. Authors in [15] derive BP Neural Network for Permanent Magnet Synchronous Linear Motor. A neural-based model reference controller is developed for a class of discrete time nonlinear systems in [24]. [19] presents neuro-predictive regulator to stabilize nonlinear model of gimbal process with interaction.

This paper presents an FTC based on fault hiding and inverse neural networks. NNs can model several nonlinear systems, including their inverse, so using RBFs in this control scheme is promising. Here, the control design employs the inverse model in series with the system, and whatever is given as input to the inverse model would be received as output from the system. Both additive and multiplicative modes are investigated for the actuator fault. Next, the nonlinear induction furnace is described. It is then subjected to an actuator fault. In the third part, inverse system modeling is followed by direct system modeling. This leads to inverse system control. There are two ways to achieve inverse system control. Simulations and results related to the article are presented in the fourth part. The results are then presented.

The structure of the article is as follows: the first part is the introduction of the article, and in the second part, the dynamic equations of the nonlinear IFS are given, and the effect of the fault of the actuator in the form of multiplication and addition is shown in the mentioned equations. In the third part, the modeling of the system was done using the RBF neural network, and in the second part of the same part, the inverse modeling of the IFS was done using the neural network. This inverse modeling is used as a controller. It is done in two ways and both methods are designed online. In the fourth section, the results of the simulations of these two control methods are given, and in the final section, five conclusions are presented

2 NONLINEAR IFS WITH ACTUATOR FAULT

Because the power supply of the induction furnace includes a rectifier, inverter and capacitive bank, there is a possibility of a fault in the actuator of this nonlinear model of the induction furnace. In this section, a healthy system is modeled by the RBF neural network, a fault occurs in 10th second, and it is observed that this fault has affected the system performance. Figure (1) shows the structure of the non-linear system of the induction furnace where the failure of the actuator occurs in the power part.

2.1 Nonlinear IFS Modeling

A current is generated through the winding core by passing a current through a coil and developing a magnetic field. Induction furnace state-space equations are expressed as follows [28]:

$$\begin{cases} \dot{x_1} = -\left(\frac{J_1}{mc}\right) x_1 + \left(\frac{k_R}{mc}\right)) x_2{}^2 - \left(\frac{1}{mc}\right) d \\ \dot{x_2} = -\left(\frac{R_{eq}}{L_{eq}}\right) x^2 + \left(\frac{k_{pwm}\ k_G}{L_{eq}}\right) u \end{cases} \tag{2.1}$$

$x_1 = \theta$ is the temperature of the induction furnace, $x_2 = I_{coil}$ is the magnitude of the current through the induction coil, $u = v$ is the amplitude of the input voltage to the induction furnace, and d is the perturbation value. L, R, KG and KP are respectively equivalent to inductance, resistance, set point gain and PWM gain, and finally m is the mass value of the work piece. The furnace's power supply consists of a rectifier (diode bridge), an inverter (dc to ac), and a capacitor bank. The presence of these elements in the power supply greatly increases the likelihood of actuator faults occurring in the nonlinear system of the induction furnace. A fault-tolerant controller for the nonlinear induction furnace system was therefore needed.

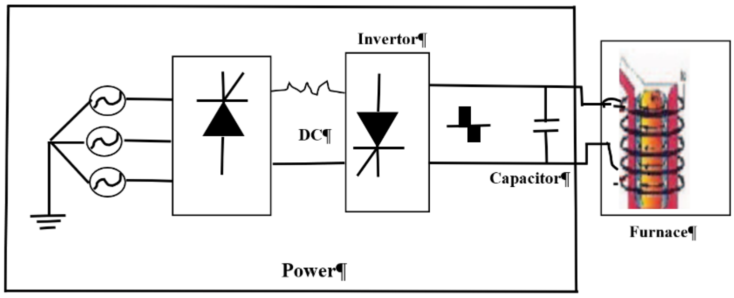

Figure 1: nonlinear induction furnace system (IFS)(24)

2.2 IFS with Actuator Fault Modeling

An actuator fault can occur because the power supply in an induction furnace is also known as the actuator. This actuator fault is represented as the equation (2.2):

$$\begin{cases} \dot{x_1} = \left(-\frac{J_1}{mc}\right) x_1 + \left(\frac{k_R}{mc}\right)) {x_2}^2 - \left(\frac{1}{mc}\right) d \\ \dot{x_2} = \left(-\frac{R_{eq}}{L_{eq}}\right) x_2 + (1-\alpha_1) \left(\frac{k_{pwm}\ k_G}{L_{eq}}\right) u \\ y = x_1 \end{cases} \tag{2.2}$$

$\alpha_f \in [0, 1)$ is the effect of the actuator fault. This equation can be rewritten as follows:

$$\begin{cases} \dot{x_1} = \left(-\frac{J_1}{mc}\right) x_1 + \left(\frac{k_R}{mc}\right)) {x_2}^2 - \left(\frac{1}{mc}\right) d \\ \dot{x_2} = \left(-\frac{R_{eq}}{L_{eq}}\right) x_2 + \left(\frac{k_{pwm}\ k_G}{L_{eq}}\right) u - \alpha_f \left(\frac{k_{pwm}\ k_G}{L_{eq}}\right) uy = x_1 \end{cases} \tag{2.3}$$

It is evident that the performance of the faulty system has changed, and the output temperature is lower than the output temperature of the health system when the fault was present. The dynamics of mentioned system can be considered as

$$\begin{cases} \dot{x_1} = (x, u)\, d \\ y = x_1 \end{cases} \tag{2.4}$$

A neural network inverse controller is being investigated to address this issue to improve the system's performance.

3 Neuro-modelling of the nonlinear IFS

In neural networks, the modeling can be done directly or in reverse, which is the case here since inverse control is used. The system and the inverse model are connected in this type of design. According to Figure (2), a signal is given as input to the inverse model of the system, and it is also inverse as output. The inverse model behaves in the same way as the system itself.

Fault-tolerant controllers must be able to maintain closed-loop system stability despite a fault. When a fault occurs, the difference between the reference input and the system output should be minimized. Inverse controllers meet this requirement to a large extent. The direct model of the system is examined first, followed by the inverse model and design of the inverse controller.

3.1 Direct Neuro-Modeling

In a direct model, inputs are used to train a neural network so that outputs can be generated. In the induction furnace system, past outputs and past and present inputs are used to determine the output because it is a nonlinear and dynamic system. In this case, the neural network is analogous to an induction furnace. Training the neural network and updating its parameters is done by comparing the neural network's output to the system's output. Equations (2.4) describe the model structure of the nonlinear induction furnace.

$$\hat{y}(k) = \hat{f}(u(k), u(k-1), u(k-2), y(k-1), y(k-2)) \tag{3.1}$$

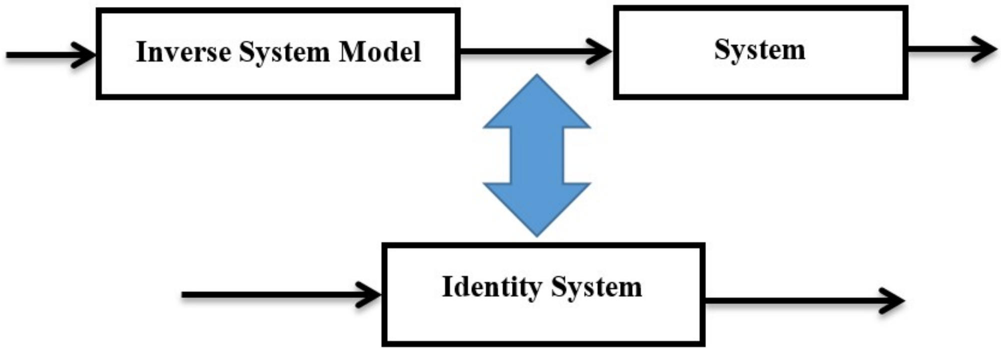

Figure 2: Inverse control scheme

$\hat{f}$ represents the input map to the output of the RBF neural network, $u(k)$ is current input and $u(k-1), u(k-2)$ are the first and second delayed inputs, $y(k-1), y(k-2)$ are the first and second delayed outputs respectively and represents the output value predicted by the neural network as follows. Therefore:

$$\hat{y_k} = \sum_j w_j \varphi (\| X_k - C_k \|) \tag{3.2}$$

where c_j stands for the middle layer Gaussian centers calculated by the K-min algorithm, W_j are the weights of the neural network, $\varphi(.)$ is the radial function of RBF neural network. $\hat{y_k}$ shows the neural network outputs and

$$x_k = [u(k), u(k-1), u(k-2), y(k-1), y(k-2)]^T.$$

The figure 2 illustrates this type of system training and modelling. Consider the following output error as below

$$e = y - \hat{y}. \tag{3.3}$$

And consequently, we consider the cost function in equation (3.2)

$$j = \frac{1}{2}e^2 = \frac{1}{2}\sum_{k=1}^{n}(y_k - \hat{y_k})^2. \tag{3.4}$$

Weights are updated using Error Back Propagation (BP) as below:

$$w(k) = w(k-1) + \eta \frac{\partial J}{\partial W}. \tag{3.5}$$

The derivative of J with respect to w is written as (3.6):

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W}. \tag{3.6}$$

Consequently, we have

$$\frac{\partial J}{\partial W} = (y_k - \hat{y_k})(-1)\varphi(\cdot). \tag{3.7}$$

The final updating law is depicted as (3.8).

$$w(k) = w(k-1) + \eta(y_k - \hat{y_k})(-1)\varphi(\cdot). \tag{3.8}$$

w is the weights of the middle layer of the RBF neural network, J is the least-squares error, which is the same as the modeling error and η shows the learning rate.

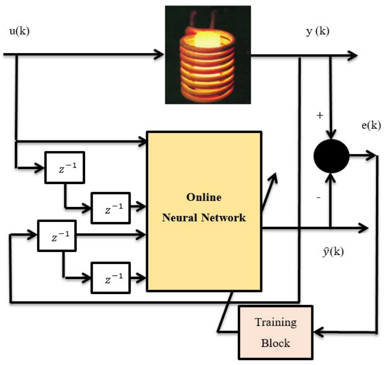

Figure 3: Direct neural network training

3.2 Inverse Neuro-Modeling

In inverse modeling, the neural network system is arranged in series. They are designed with two modeling and control methods in this article.

3.2.1 First Approach: The Neuro-Model Procedure

A neural network is used to model the inverse dynamics during training, known as inverse modeling. The training is dynamic. Five inputs are used to train the neural network and reverse model the induction furnace system online: the temperature, the first delayed time, and the second temperature. The first and second delays of the induction furnace system voltage are also inputs to this neural network. Voltage and temperature are the inputs and outputs of the induction furnace system, respectively. RBF neural network parameters are trained and updated using reverse modeling error signals. This type of training is illustrated in Figure 4. In the induction furnace input to the RBF neural network, temperature and the first and second temperature delays correspond to the first and second voltage delays.The inverse dynamics are given by equation (3.9):

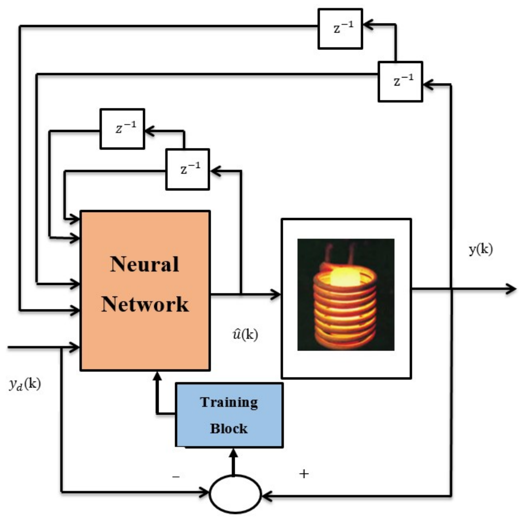

Figure 4: Online Inverse controller (first method)

$$u(\hat{k}) = \hat{f^{-1}}(u(k-1), u(k-2), y(k), y(k-1), y(k-2), y_d(k)). \tag{3.9}$$

There are $\hat{f}^{-1}$ reverse mapping systems in this method. The difference between the output of the inverse neural network and the input of the induction furnace system is used as a training signal. Therefore, the system can be considered a layer within the neural network [37]. Based on the inverse dynamic structure mentioned in figure 4, the structure of the system can be represented as:

$$y(k) = f(\hat{u}(k), \hat{u}(k-1), \hat{u}(k-2), y(k-1), y(k-2)). \tag{3.10}$$

In this paper, the RBF neural network is used for modeling. The system is the fourth layer in this type of neural network, and it cannot be modified. The output error must now be reduced to a minimum. Therefore

$$e = (y - y_d) \tag{3.11}$$

and the cost function is candidate as below.

$$J_{mse} = \frac{1}{2}e^2 = \frac{1}{2}\sum_{k=1}^{n}(y_k - y_{d,k})^2 \tag{3.12}$$

y and yd represent the desired output and an induction furnace system output, respectively. The updating law of weighting parameters

$$w(k) = w(k-1) - \mu e \hat{f}(\hat{u}) g \tag{3.13}$$

where μ shows learning rate and consequently, $\frac{\partial J_{mse}}{\partial W}$ can be rewritten as

$$\frac{\partial J - mse}{\partial W} = \frac{\partial J - mse}{\partial e}\frac{\partial e}{\partial y}\frac{\partial y}{\hat{u}}\frac{\partial \hat{u}}{\partial W}. \tag{3.14}$$

The $\hat{u}$ represents the neural network output, which is the value of the control input. This means that the induction furnace system is considered as the last layer of the neural network so that a term containing can be created in formula (3.8);

$$\frac{\partial e}{\partial \hat{u}} = \frac{\partial e}{\partial y}\frac{\partial y}{\partial \hat{u}}. \tag{3.15}$$

Equation (3.9) is rewritten using Equation (3.8):

$$\frac{\partial e}{\partial \hat{u}} = (y - y_d)\frac{\partial y}{\partial \hat{u}}. \tag{3.16}$$

Therefore, the derivatives of each term in Equation (3.12) can be written as follows:

$$\frac{\partial J - mse}{\partial W} = e \times 1 \times \hat{f}(\hat{u}) \times g. \tag{3.17}$$

Finally, the updating law for the weights is written as:

$$w(k) = w(k-1) - \mu(e \times 1 \times \hat{f}(\hat{u}) \times g) \tag{3.18}$$

g displays the Gaussian functions at the middle layer of the RBF neural network and w to the weights. As a result, the inverse model of the system can be trained correctly, and the inverse model is used as a controller to hide the activator fault in the system. The results of this controller are presented next. The inverse model of the system is trained using its sign instead of $\hat{f}$, and it is used as a controller to hide the activator fault. This controller will be discussed in the next section.

3.2.2 Second Approach: The Neuro-Model Procedure

It is performed in the same way as the first method, with the difference being the input for neural network training. In this case, in addition to the inputs associated with the method before the error e, the error derivative and error integral are also used for training. This method reduces the overshoot and enhances the controller's performance.

$$\hat{u}(k) = \hat{f}^{-1}(u(k-1), u(k-2), y(k), y(k-1), y(k-1), e(k), de(k), se(k)) \tag{3.19}$$

where $se(k)$ and $de(k)$ stand for the error integral, and derivative, consequently. that the $e(k)$, $de(k)$ and $se(k)$ signals in the inverse neural network system are only used as learning signals and are not part of the inverse signals and play the role of limiting training for the inverse neural network. The neural network uses a PID methodology to update weights. Figure 5 illustrates this procedure. The design softens the existing overshoot, whose results are described in the next section. It is easily seen from the figure that in the first method for training the RBF neural network, we had six inputs, of which one is the output error, but in the second method, in addition to the same six inputs as in the first method, there were two other inputs added. One is the error derivative and the other is the error integral. This method reduces the modeling error. Due to the derivative and integral effect of the error. In the second method, the overshoot has been decreased compared to the first one.

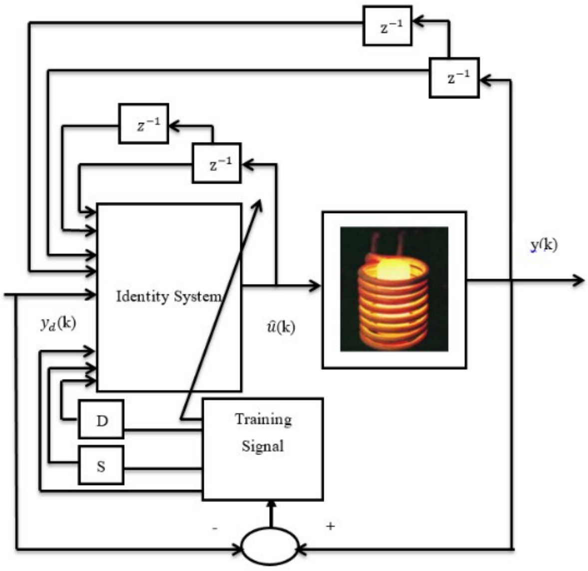

Figure 5: Online Inverse controller (second method)

Table 1: The values of the nonlinear IFS parameters

Parameters	Value
U	0-15 volt
K_G	0.256
K_PWM	28.02
R_eq	0.0481
L_eq	$2.721*10^-4$

4 Results and Discussion

The non-linear system of the induction furnace consists of three parts: the actuator, the sensor, and the main structure. There will be a possibility of failure in all three sub-systems, but according to the block diagram of the induction furnace system in Figure (1) and the components of the actuator, there is a possibility of occurrence The fault in the actuator is more than the other two subsystems. Therefore, in this article, we have studied the fault of the actuator. In order not to need additional hardware for re-tracking and the system does not stop due to being faulty during actuator, we use the inverse of the system online and intelligently, which is modeled using the RBF neural network, and In this way, the controller is the inverse model of the system and will fix the problem in a short time

4.1 Neuro-Modeling of IFS

Modeling of this type is done online. This was discussed in the previous section. Model output and the actual output of the induction furnace are negligible to zero. Therefore, the model is accurate.

As shown in Figure 6, the system and modeling output have the same value. Based on the modeling error as shown in figure 7, the truth of this statement is shown.

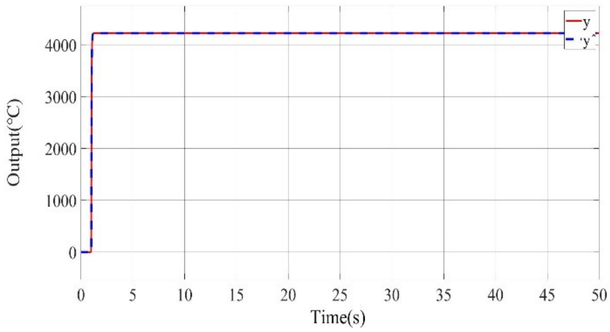

Figure 6: Modeling of induction furnace system

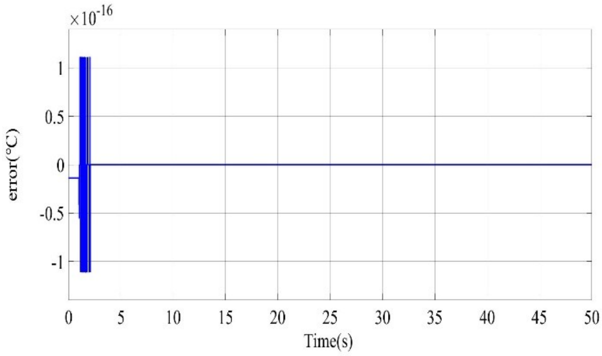

Figure 7: Modeling error

4.2 Comparison of Healthy And Faulty Systems

The activator fault is used here. The fault is applied from 10 to 30 seconds. Figure (8) shows that the output temperature of the system has dropped during this period, while the difference between the faulty and healthy system outputs has grown significantly. The system will malfunction if an activator fault occurs, so a tolerable fault controller is required to fix it.

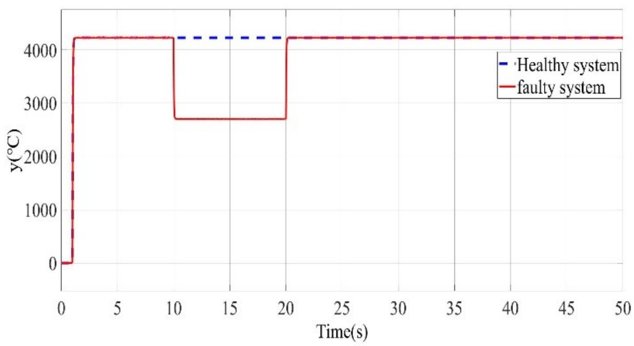

Figure 8: Comparison between healthy system and defective system

So, as can be seen in Figure 8, the actuator fault caused the system to deviate from its normal operating mode. Consequently, the temperature of the faulty furnace system is drastically reduced in association with the healthy one. This leads to the design proper Neuro inverse dynamical controller in the following section.

4.3 Design of Neuro-Inverse Dynamical FTC

In this design, the inverse model of the induction furnace system is used for the controller. The inverse model is obtained online using an RBF neural network, and the system is placed in series with the neural network. As the last layer of a neural network, the system is fixed and unchangeable. In this layer, induction furnaces are used to train the neural network by updating the weights with the error propagation algorithm and the Gaussian centers with the K-mean algorithm. As discussed in the previous section, there are two methods of teaching. Given the overshot, the first method is also suggested. In the second method, the derivative and integral errors in the input vector are used to reduce the overshoot value, as can be seen in Figures (9) and (10), respectively. Figure (10) shows that the overshot has been reduced to Figure (9).

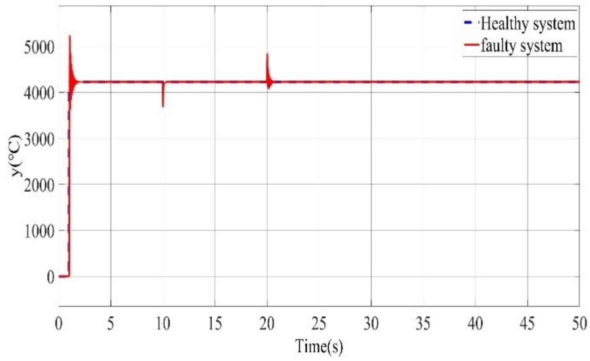

Figure 9: Hiding fault with online inverse controller (first method)

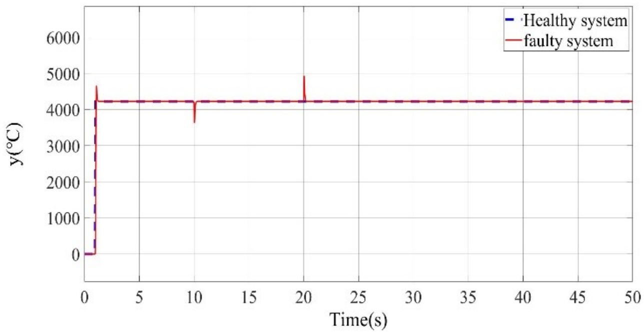

Figure 10: Hiding fault with online inverse controller (second method)

The results of two hiding fault controllers are presented in this section:

1) As shown in Figure 9, the first method uses six inputs for the neuro-system. In this case, the actuator fault is well fixed, but the temperature in the faulty system has a large overshoot that is unacceptable in this system.

2) According to Figure 10, the reverse dynamic is taught using eight inputs and is used as a fault-tolerant controller. The two inputs are also the same derivative and error integral for the first method. By using this method, both overshoot and settling time are reduced.

5 Conclusion

Fault tolerance control can control and restore a faulty system to its optimal state. An active tolerable control method based on inverse strategy is presented in this paper. To implement the inverse strategy, RBF neural networks are used. An activator fault is applied to the system, and the fault-tolerant system returns to its ideal state and optimal performance in less than one second. To reach this conclusion, two different approaches were used, and the second method, using a PID controller in the neural network, was able to reduce overshoot. Within 10 to 30 seconds, an actuator fault is applied to the induction furnace system, which, without the fault-tolerant controller, causes the system's output to go out of the desired state. However, using the fault-tolerant controller, which is the reverse model of the system, this actuator fault leads to a fixed system malfunction. This inverse model of the induction furnace system is taught in two ways. The first method uses five inputs for inverse modeling, and the second method uses five inputs in addition to five inputs in the first method. This method reduces overshot compared to the first case with output error, derivative error, and integer error. An extension of this method to recurrent neural network to improve results and practical implementations of our approach can be considered in future studies.

References

[1] A. Alexandridis, M. Stogiannos, N. Papaioannou, E. Zois, and H. Sarimveis, *An inverse neural controller based on the applicability domain of RBF network models*, Sensors **18** (2018), no. 1, 315.

[2] Q. Chen, G. Liu, D. Xu, L. Xu, G. Xu, and N. Aamir, *Decoupling control of a five-phase fault-tolerant permanent magnet motor by radial basis function neural network inverse*, AIP Adv. **8** (2018), no. 5.

[3] J. Chen and R.J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Vol. 3, Springer Science & Business Media, 2012.

[4] S. Cho, Zh. Gao, and T. Moan, *Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines*, Renew. Energy **120** (2018), 306–321.

[5] B. Cui, Bing, Ch. Zhao, T. Ma, and Ch. Feng, *Distributed adaptive neural network control for a class of heterogeneous nonlinear multi-agent systems subject to actuation failures*, Int. J. Syst. Sci. **48** (2017), no. 3, 559–570.

[6] Z. Gallehdari, N. Meskin, and Kh. Khorasani, *A distributed control reconfiguration and accommodation for consensus achievement of multiagent systems subject to actuator faults*, IEEE Trans. Control Syst. Technol. **24** (2016), no. 6, 2031–2047.

[7] Zh. Gao, B. Han, G. Jiang, J. Lin, and D. Xu, *Active fault tolerant control design approach for the flexible spacecraft with sensor faults*, J. Franklin Inst. **354** (2017), no. 18, 8038–8056.

[8] H. Gao, Y Song, and Ch. Wen, *Backstepping design of adaptive neural fault-tolerant control for MIMO nonlinear systems*, IEEE Trans. Neural Networks Learn. Syst. **28** (2016), no. 11, 2605–2613.

[9] M.A. Hussain, J. Mohd Ali, and M.J. H. Khan, *Neural network inverse model control strategy: discrete-time stability analysis for relative order two systems*, Abstr. Appl. Anal. **2014** (2014).

[10] L. Jin, B. Liao, M. Liu, L. Xiao, D. Guo, and X. Yan, *Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network*, Front. Neurorobotics **11** (2017), 50.

[11] K. Jun-Jie, C.U.I. Yan-Jun, W. Zhi-Qiang, F. Qiao-Yun, and D.O.N.G. Jia, *Medium frequency induction furnace temperature control based on BP neural network PID*, IEEE Int. Conf. Inf. Automation (ICIA), IEEE, 2016, pp. 1110–1115.

[12] H. Kazemi and A. Yazdizadeh, *Fault detection and isolation of gas turbine engine using inversion-based and optimal state observers*, Eur. J. Control **56** (2020), 206–217.

[13] M. Kordestani, K. Salahshoor, A. Akbar Safavi, and M. Saif, *An adaptive passive fault tolerant control system for a steam turbine using a PCA based inverse neural network control strategy*, World Automation Congress (WAC), IEEE, 2018, pp. 1–6.

[14] D.-Y. Li, P. Li, W.-Ch. Cai, X.-P. Ma, B. Liu, and H.-H. Dong, *Neural adaptive fault tolerant control for high speed trains considering actuation notches and antiskid constraints*, IEEE Trans. Intell. Transport. Syst. **20** (2018), no. 5, 1706–1718.

[15] Y. Li, Y. Yu, X. Hu, and X. Liu, *Vector control of permanent magnet synchronous linear motor based on improved bp neural network*, Mechatron. Syst. Control **49** (2021), no. 2, 109–114.

[16] W. Liao, W. Liang, and A. Wang, *Inverse dynamics control of a parallel robot based on RBF neural network*, Int. Conf. Adv. Mechatron. Syst. (ICAMechS), IEEE, 2017, pp. 372–375.

[17] Ch.-M. Lin and E.-A. Boldbaatar, *Fault accommodation control for a biped robot using a recurrent wavelet Elman neural network*, IEEE Syst. J. **11** (2015), no. 4, 2882–2893.

[18] M. Liu, P. Shi, L. Zhang, and X. Zhao, *Fault-tolerant control for nonlinear Markovian jump systems via proportional and derivative sliding mode observer technique*, IEEE Trans. Circ. Syst. I: Regular Papers **58** (2011), no. 11, 2755–2764.

[19] S. Niazi, A. Toloei, and R. Ghasemi, *Neuro-predictive controller for stabilization of gimbal mechanism with cross-coupling*, Mechatron. Syst. Control. **49** (2021), no. 4.

[20] V. Radhakrishna, Sh.A. Aljawarneh, P. Veereswara Kumar, and V. Janaki, *A novel fuzzy similarity measure and*

prevalence estimation approach for similarity profiled temporal association pattern mining, Future Gener. Comput. Syst. **83** (2018), 582–595.

[21] O. Rafael, O.J. Saul, and R. Morales, *Simulation of a temperature fuzzy control into induction furnace*, 13th Int. Conf. Power Electronics (CIEP), IEEE, 2016, pp. 64–69.

[22] R. Ristiana, A. Syaichu-Rohman, and P.H. Rusmin, *Modeling and control of temperature dynamics in induction furnace system*, 5th IEEE Int. Conf. Syst. Engin. Technol. (ICSET), IEEE, 2015, pp. 6–11.

[23] C. Robles Algarín, D. Sevilla Hernández, and D. Restrepo Leal, *A low-cost maximum power point tracking system based on neural network inverse model controller*, Electronics **7** (2018), no. 1, 4.

[24] J. Sweafford Jr and F. Fahimi, *A neuralnetwork model-based control method for a class of discrete-time nonlinear systems*, Mechatron. Syst. Control **49** (2021), no. 2.

[25] L. Tang, D. Ma, and J. Zhao, *Neural networks-based active fault-tolerant control for a class of switched nonlinear systems with its application to RCL circuit*, IEEE Trans. Syst. Man Cybernet.: Syst. **50** (2018), no. 11, 4270–4282.

[26] M.M. Tavakoli and N. Assadian, *Predictive fault-tolerant control of an all-thruster satellite in 6-DOF motion via neural network model updating*, Adv. Space Res. **61** (2018), no. 6, 1588–1599.

[27] N. Teng and J. Zhang, *Vacuum induction heating furnace temperature control system based on Smith fuzzy-PID*, Int. Conf. Mechatron. Control (ICMC), IEEE, 2014, pp. 2207–2210.

[28] N. Torabi, M.R. Motavalli, A. Mihankhah, and S. Rastani, *Fault tolerant sliding mode intelligent control based on fault hiding for a nonlinear induction furnace system*, Int. J. Dyn. Control **9** (2021), 636–644.

[29] M. Versaci, G. Angiulli, P. Crucitti, D. De Carlo, F. Laganà, D. Pellicanò, and A. Palumbo, *A fuzzy similarity-based approach to classify numerically simulated and experimentally detected carbon fiber-reinforced polymer plate defects*, Sensors **22** (2022), no. 11, 4232.

[30] G. Wei and Y. Wei, *Similarity measures of Pythagorean fuzzy sets based on the cosine function and their applications*, Int. J. Intell. Syst. **33** (2018), no. 3, 634–652.

[31] Sh. Xing, J. Ju, and J. Xing, *Research on hot-rolling steel products quality control based on BP neural network inverse model*, Neural Comput. Appl. **31** (2019), 1577–1584.

[32] G. Yao, Y. Li, Q. Li, Sh. Hu, and N. Jin, *Model predictive power control for a fault-tolerant grid-connected converter using reconstructed currents*, IET Power Electron. **13** (2020), no. 6, 1181–1190.

[33] S. Yaqubi, M. Homaeinezhad, and M.R. Homaeinezhad, *Adaptive fuzzy-wavelet neural networks-based real-time model generation for increasing tracking precision of multivariable servo actuator*, Mechatron. Syst. Control **50** (2022), no. 1.

[34] Y. Zhang and J. Jiang, *Bibliographical review on reconfigurable fault-tolerant control systems*, Ann. Rrev. Control **32** (2008), no. 2, 229–252.

[35] Q. Zheng, Z. Du, D. Fu, Zh. Hu, and H. Zhang, *Direct thrust inverse control of aero-engine based on deep neural network*, Int. J. Turbo Jet-Engines **38** (2021), no. 4, 391–396.

[36] Zh. Zheng, W. Guocheng, and L. Xiaowei, *An intelligent monitoring system of medium-frequency induction furnace based on fuzzy control*, Int. Conf. Intell. Syst. Design Engin. Appl., IEEE, 2010, pp. 261–264.

[37] Zh. Zheng, L. Xiao-Wei, and W. Guo-Cheng, *The study of intelligent control of medium frequency induction furnace based on RBF neural network*, Int. Conf. Intell. Syst. Design Engin. Appl., IEEE, 2010, pp. 740–743.

[38] Q. Zhou, P. Shi, H. Liu, and Sh. Xu, *Neural-network-based decentralized adaptive output-feedback control for large-scale stochastic nonlinear systems*, IEEE Trans. Syst. Man Cybernet. Part B **42** (2012), no. 6, 1608–1619.