MODELING AND CONTROLLING HEAT TRANSFER IN CHAMBERS: A COMPARATIVE STUDY OF CLASSICAL AND INTELLIGENT APPROACHES

Radiša Jovanović, Mitra Vesović, Natalija Perišić

Faculty of Mechanical Engineering, University of Belgrade, Belgrade, Serbia

[*] Mitra Vesović; E-mail: mvesovic@mas.bg.ac.rs

This paper introduces nonlinear approaches which include neural networks and ANFIS to identify and control heat transfer within a chamber. Initially, traditional linear models are obtained using transfer functions with delays through Matlab identification tools. However, this traditional linear model failed to faithfully represent the system when the input was changed. This outcome was expected since linear models are reliable only within specific operational ranges. To create a novel model that is applicable across the entire state space, two alternative identification methods, utilizing neural networks and an adaptive neuro-fuzzy inference system were introduced. After testing them with input data not used during the training, the models were compared and all of them showed satisfiying results.. In the continuation of the research, control techniques based on these techniques were presented. After assigning an arbitrary temperature as a reference signal, inverse models were made and four controllers in direct inverse control sheme were compared: three feedforward neural networks with different numbers of neurons in the hidden layer and the adaptive neuro-fuzzy inference controller. The results and possible improvements are discussed in the conclusion.

Key words*: heat flow system identification, direct inverse control, artificial intelligence, neural networks, ANFIS*

1. Introduction

In a multitude of diverse industries, including manufacturing, production, and services, precise temperature control is of great importance. It is a crucial factor that deeply influences the quality of products, as well as operational security [1]. The operational efficiency of heating, ventilation, and air conditioning (HVAC) systems relies significantly on the principles of heat transfer [2]. Precise models of heat transfer play a key role in the understanding of energy conversion processes, and they serve as valuable tools in enhancing the effectiveness of potential control mechanisms. This means that optimal temperature control, crucial in many industry fields, depends on identifying heat transfer processes and using optimal control strategies.

This study deals with the identification of heat flow processes and temperature control, using the Quanser Heat-flow Experiment (HFE) [3]. There are several research papers about modeling and control of the same system. For example, in [4] fractional order integral and derivative (FO-ID) controller is tuned, while it is considered that the system can be represented by the first order plus time delay transfer function. The proposed controller is compared with fraction order PI and classical PI and PID controllers tuned by Zigler Nichol's method. It was shown that fraction order controllers

outperform classical controllers and that for the proposed FO-ID, the resulting closed loop system is robust to gain variations. Research [5] deals with the design of a memristor-based two-degree-of-freedom controller for temperature profile tracking control. As a result of the simulation, it was shown that two-degree-of-freedom control structures reach the reference value faster than one-degree-of-freedom control structures, while memristor-based control structures eliminate the error earlier than control structures with standard controllers. Identification and control of a heat flow system designed based on the Takagi-Sugeno fuzzy model optimized using the Grey Wolf Optimization Algorithm are done in [6]. Three different experiments were done and the Optimized Takagi-Sugeno fuzzy controller proved to be suitable for the task. In [7], authors developed an adaptive neuro-fuzzy inference system (ANFIS) whose architecture is found by using the genetic evolutionary optimization algorithm, for the identification of heat flow process.

In this research identification and control of heat transfer are done by using nonlinear approaches that involve artificial intelligence. Intelligent control is realized in direct inverse control (DIC) scheme. In literature, feedforward neural networks (FFNNs) are commonly used in DIC schemes, such as for the control of the Unmanned Aerial Vehicle [8] or for active vibration suppression control [9]. In this paper, FFNNs are also used for this purpose, but we used ANFIS model as well. The comparison was made between a few FFNNs with variations in the number of neurons in the hidden layer and the ANFIS model. All experiments are performed on the real system in laboratory conditions.

2. System description

The Quanser HFE in Fig. 1, is a sophisticated system designed for the study and analysis of heat flow and thermal dynamics. The device is an advanced rheostat, which is designed to optimize heat conduction and enable accurate temperature monitoring at different locations. Three temperature sensors are evenly distributed along the aluminum plate. These sensors are fast-setting platinum transducers, ensuring precise and rapid temperature data collection. The setup includes a blower that plays a crucial role in controlling the airflow within the system. A tachometer is used to measure the blower's fan speed. The system has a coil-based heating device that provides a controlled heat source. To control the delivered power, analog signals (Vh voltage for heating and Vb voltage for the blower) are employed. These signals enable reliable adjustments in power supply, facilitating various heat flow experiments. Quanser's software is utilized to collect and analyze data from the temperature sensors. This software provides a user-friendly interface for real-time monitoring and in-depth analysis of the experiments. HFE offers a controlled and precise environment for conducting experiments and investigations related to heat flow.

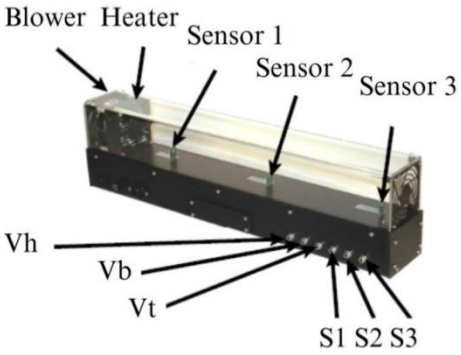

Figure 1 The HFE set-up [4]

3. HFE Modelling

When it comes to establishing a nonlinear input-output relationship or dealing with a large number of input variables but few samples with heterogeneous data, the more recent approaches tend to offer notable advantages over the more established (traditional) ones. Similar to conventional models, modern approaches assess the achieved performance through established statistical methodologies. Artificial intelligence is precisely one of these contemporary techniques. Fuzzy logic algorithms, artificial neural networks, metaheuristic algorithms, and expert systems are examples of algorithms that incorporate aspects of human thought processes and problem-solving techniques. This chapter will describe three mathematical models of the system: linear model derived through identification, model based on neural identification using feedforward network and fuzzy-neural model: ANFIS.

3.1. Traditional model

Explicit mathematical equations are the basis of traditional models such as transfer functions. Classical models frequently excel in interpretability, simplicity of use, explicit assumptions, stability, and robustness, even when more recent models might offer flexibility.

An open-loop experiment was carried out to determine the heat flow system's mathematical representation. All of the temperature sensors were used to measure the temperature within the chamber. The experiment started, and five seconds in, a 5V step signal was applied. Throughout the whole testing, the blower input voltage stayed at 3V. The experiment automatically finished after two minutes. Unsurprisingly, sensor 1 displayed a faster temperature rise than sensors 2 and 3 due to its closer proximity to the heater and blower, Fig. 2. As a result, the chamber's temperature increased at different rates. Three models were created, each of which matched the sensor's temperature data.

In the literature, it is usual that HFE models are represented as first-order transfer functions with delay [1], [3-7],[10]. Furthermore, the step responses of these models are displayed in Fig. 2. The findings obtained from the second sensor (s2) yield the lowest mean square error (MSE). Using the Matlab identification tool, the linear models are obtained in the form of the transfer functions W_i, i = 1, 2, 3, using Laplace transform, that converts a time domain into domain of complex variable s, Eq. (1).

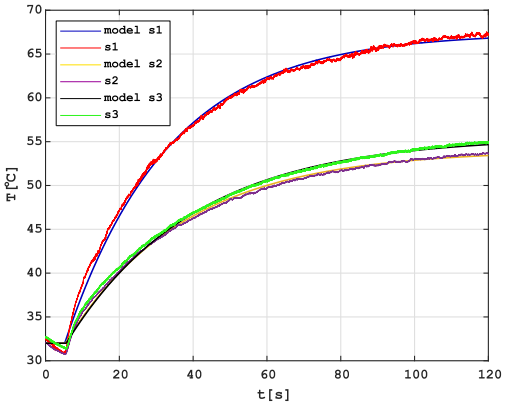

Figure 2 Traditional model: Transfer functions with delay [7]

$$W_1(s) = \frac{0.2523}{s+0.03563} e^{-0.198 s},\ \text{MSE} = 0.2096;$$
$$W_2(s) = \frac{0.137}{s+0.03107} e^{-0.396 s},\ \text{MSE} = 0.0949; \quad (1)$$
$$W_3(s) = \frac{0.1458}{s+0.03242} e^{-0.594 s},\ \text{MSE} = 0.1349$$

3.2. Intelligent models

Advanced modeling, which can include machine learning and artificial intelligence, offers distinct advantages over traditional linear modeling approaches. Unlike linear methods, nonlinear models can capture complex and non-linear relationships in data, making them highly adaptable to a wide range of real-world scenarios. They excel in various tasks, especially modeling [11], although their usage has been limited due to the black box nature and unclear processes.

3.2.1 FFNN models

The principal advantage of utilizing an artificial neural network (ANN) model is its built-in ability to learn from itself and to efficiently represent and approximate complex nonlinear relationships between input variables and output results [12-13]. The FFNN, a popular ANN design, was employed in this study to forecast the chamber's temperature based on the input voltage. An FFNN comprises interconnected neurons in input, output, and hidden layers with flexible weighted connections. Nonlinear activation functions in the hidden layer enable universal approximation capabilities. Training typically employs the back-propagation algorithm, incorporating techniques such as Levenberg-Marquardt, Bayesian regularization, robust, and gradient descent for weight and bias modification. A trained neural network can be used to forecast the results of inputs that were not used during training. A multilayer FFNN with a single hidden layer and a backpropagation learning algorithm is employed in this work. The network output in the scalar case for the single hidden layer network is as follows, Eq. (2):

$$\hat{y} = f_2\left\{\sum_{i=1}^{h} w_i f_1\left[\sum_{j=1}^{n} w_{ij} x_j + b_{i0}\right] + b_0\right\}, \quad (2)$$

where the network output $\hat{y}$ is the predicted value of the variable y, w_i and w_{ij} are the weights and b_{i0} and b_0 are biases. The scalar h denotes the number of neurons in hidden layer and n denotes the number of inputs, while f_1 and f_2 are activation functions in hidden and output layer, respectively.

3.2.2 ANFIS model

ANFIS is a remarkable machine learning model that combines the best aspects of neural networks and fuzzy logic. It was developed by Jang, Sun, and Mizutani in the early 1990s. [14]. To increase prediction accuracy, the ANFIS neural network component modifies the fuzzy sets and operator settings. For this, the gradient descent technique known as the backpropagation algorithm is typically used to reduce the discrepancy between the expected and actual outputs. The structure of ANFIS is made up of five layers, and, usually, papers show a structure with two input x_k, k = 1,2 and one output. Fig. 3 represents architecture, based on the first-order Takagi–Sugeno model, with the two

membership functions j = 1, 2 and four rules m = 1, 2, 3, 4. With A_{1k}, $A_{2k, \ldots,} A_{jk}$ denoting linguistic labels, q_{mk} and c_m denoting the consequent parameters, a typical set of rules can then be written:

If x_1 **is** A_{11} **and** x_2 **is** A_{12} **then** $f_1 = q_{11}x_1+q_{12}x_2+c_1$

If x_1 **is** A_{11} **and** x_2 **is** A_{22} **then** $f_2 = q_{21}x_1+q_{22}x_2+c_2$,

If x_1 **is** A_{21} **and** x_2 **is** A_{12} **then** $f_3 = q_{31}x_1+q_{32}x_2+c_3$

If x_1 **is** A_{21} **and** x_2 **is** A_{22} **then** $f_4 = q_{41}x_1+q_{42}x_2+c_4$,

- Input Layer computes and outputs the necessary membership function's membership degree. Every node in this layer can change its shape while being trained. Each input node sends the input value to the next layer by defining an input variable, $\mu_{A_{jk}}$ corresponds to one membership function of *k-th* input. Gaussian or bell-shaped membership functions are commonly used in the literature, despite various other types being explored. The Gaussian membership function, which has two parameters (in the case of ANFIS those are called premise parameters: standard deviation β and mean α).

$$\mu_{Ajk}(x_k) = e^{-\frac{(x_k-\alpha_{jk})^2}{2\beta_{jk}^2}}. \tag{3}$$

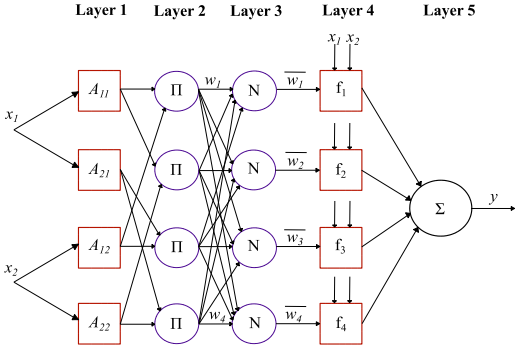

Figure 3 ANFIS architecture with two inputs

- The nodes in the second layer don't change like they did in the first layer – they stay fixed. The firing strength of the associated rule w_m, is shown by each node's output. A stronger firing strength implies that the regulation has more of an impact on the outcome.

$$\begin{aligned} w_1 &= \mu_{A_{11}}(x_1)*\mu_{A_{12}}(x_2);\ w_2 = \mu_{A_{11}}(x_1)*\mu_{A_{22}}(x_2), \\ w_3 &= \mu_{A_{21}}(x_1)*\mu_{A_{12}}(x_2);\ w_4 = \mu_{A_{21}}(x_1)*\mu_{A_{22}}(x_2), \end{aligned} \tag{4}$$

where * denotes T-norm.

- By dividing a rule's firing strength by the total of all firing strengths, the third layer calculates the normalized firing strength of each rule, making sure that the values are between 0 and 1.

$$\bar{w}_m = \frac{w_m}{\sum_m w_m}, m = 1,2,3,4. \tag{5}$$

- The nodes in the fourth layer combine the firing strengths of the rules to get a crisp output value. It is also often called the defuzzification layer. Various defuzzification techniques exist, including the weighted average method and the center of gravity method. In order to get the weighted

consequent values, this layer multiplies each rule's consequent parameter by the normalized firing strength.

$$\bar{w}_m f_m = \bar{w}_m \sum_{k=1}^{2} q_{mk} + c_m. \tag{6}$$

The weighted consequent values from each rule are added up in the last layer, which produces the ANFIS system's overall output, where *l* is the total number of rules, l= 4 for Fig.3.

$$y = \sum_m^l \bar{w}_m f_m \frac{1}{\sum_m^l w_m} \sum_m^l w_m \left(\sum_{k=1}^{2} q_{mk} + c_m \right), \tag{7}$$

3.3. Traditional model testing

The transfer function chosen to characterize the system is generated from sensor s2 using Eq. (1) in light of the excessive noise in sensor 3's output signal and the greater error displayed by sensor 1. The sensor situated at the center of the chamber exhibits the lowest mean square error (MSE), as depicted in Fig. 2. Further, the comparison between the model and the actual object, but with an altered input signal in Fig. 4 is studied. In this instance, the experiment started with a 4V step signal delivered five seconds after the beginning (identification, i.e. object model is obtained for an input voltage of 5 V). The findings obtained a much higher MSE of 1.1888, and in a steady state, there is a mismatch of about 1.5°C between the model and the actual output signal, and this difference tends to increase as we travel away from the initial identification point. Fig. 4 has objective to highlight the system's nonlinearity. As a result, it is found that this particular linear model is insufficient for accurately representing the system when the input changes.

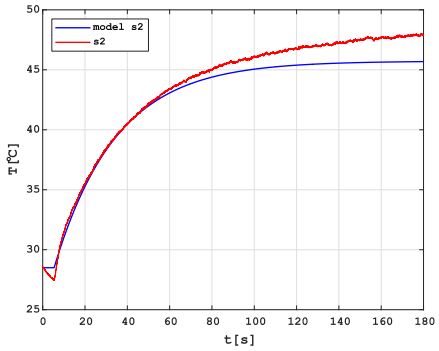

Figure 4 Traditional model (linear with time delay) testing [7]

4. Nonlinear identification

Just like in the case of the linear model, when dealing with nonlinear models, the data acquisition process during testing was exclusively centered around the sensor s2. The selection of the MSE criterion remained consistent as the performance metric of choice. All models (FFNNs and ANFIS) were created using different input voltages: 1V, 1.5V, 2V, 2.5V, 3V, 3.5V, 4V, 4.5V, and 5V (while the blower input voltage stays constant), and it was tested for an input it was not trained on: 3.2V. Although these models cannot be directly compared with linear one, because the linear model was designed for a single input voltage, the object's non-linear character motivates the employment of non-analytical yet alternative methodologies.

4.1. Neural networks

In this research, data was divided randomly, with 70% taken for training, 15% for validation and 15% for testing. Because it offers quick convergence, Levenberg-Marquardt (LM) algorithm was used for training and MSE is chosen as performance criteria to make the later comparison with other techniques. Learning rate is fixed at 0.001. Using historical input-output data, training was conducted offline. The previously mentioned parameters are maintained for all architectures. Using input u as input voltage and output y as temperature from the current and the previous instants (k and k-1), object identification is accomplished and FFNN output can be defined as follows, Eq. (8):

$$y(k+1) = N\left(u(k-1), u(k), y(k-1), y(k)\right) \tag{8}$$

4.1.1 Different architectures and results

Although the training was conducted offline, testing of all networks was carried out online using Simulink to compare the behavior of the object and its model in real-time for three minutes. Three different FFNN architectures with five, eight and ten neurons in the hidden layer are used in order to find the best possible model, Fig. 5-7. The experiments were also made with other network architectures, i.e. with a different number of neurons in the hidden layer as well as a larger number of hidden layers, but this did not bring improvements.

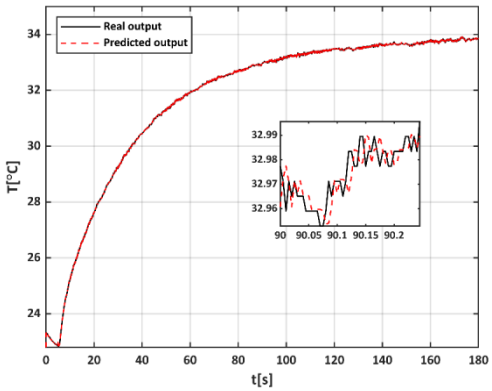

Figure 5 FFNN with 5 neurons

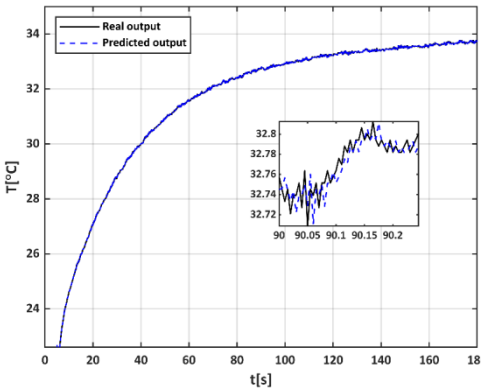

Figure 6 FFNN with 8 neurons

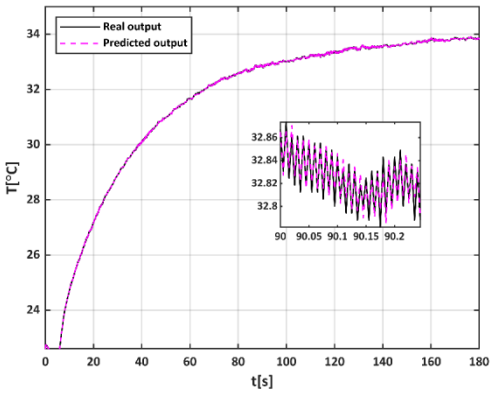

Figure 7 FFNN with 8 neurons

The best of all architectures were those three, and the best among them proved to be the network with eight number of neurons, with a MSE of 0.0116. On the Fig. 5 FFNN with five neurons is presented, and the best one is shown in Fig.6 where real output is denoted with black and predicted output with the blue dashed line. The MSE for this network is 0.0230. The model responses match well with the actual object response in all three Figures as shown in the enlarged view of the graph from 90 to 90.2 seconds. In Fig. 7, FFNN with 10 neurons is shown, with MSE of 0.0151.

4.2. ANFIS

As in the case of neural networks, for training the ANFIS model, the data is divided in the ratio 70% -15%-15% for training, validation and testing, respectively. A hybrid training algorithm was applied, which includes two different parameter adjustment methods. As to Jang's [14] findings, the hybrid algorithm modifies the premise parameters during the backward pass, which are then carried over to the forward pass of the ANFIS. The least squares approach will determine the consequent parameters when the input enters Layer 4 in a forward pass. The MSE criterion was consistently chosen as the preferred performance metric. The training was done offline, and testing online, in the same way and with the same prerecorded input data as with neural networks, so the output has the same dependency from the previous inputs as in Eq. (8).

4.2.1 Modeling with ANFIS: Results and Analysis

The number of rules in a traditional fuzzy inference system is chosen by the engineer or researcher who has experience with the system that has to be modeled. Finding the bare minimum of membership functions required to reach a given performance level in advance is not an easy task. In this attempt, the intended input-output data was examined, and trial and error was used to determine the number of membership functions that should be assigned to each input variable. There are 24 premise parameters in the model, which includes 3 Gaussian membership functions per input, with 2 parameters each and a total of 4 inputs. The first step is to apply grid partitioning (GP) to the input-output data pairs in order to derive the basic fuzzy model. The input and output data are used to build the GP technique, which guarantees that the membership functions are uniformly spaced and have the same shapes. Even with a moderate number of inputs, however, the enumeration of all conceivable combinations of membership functions for each input might result in an exponential increase in the number of rules. The present scenario's grid partitioning produces 81 rules (3^4) for the fuzzy inference system with 4 inputs and 3 Gaussian membership functions per input since ANFIS-GP models suffer with the curse of dimensionality (when number of inputs or number of membership functions increases, then the number of fuzzy rules also increases exponentially).

In the ANFIS structure presented in this paper, there are a total of 81 consequent parameters, because it is chosen to have only one consequent parameter per rule (only an independent consequent parameter is kept). In this case, the number of rules is equal to the number of consequent parameters. The ANFIS's training and structure, as well as the real-time calculation involved in comparing the model's response to the response of the real object, would have become much more complex if this hadn't been done, increasing the number of consequent parameters to 405 (5*81).

Consequently, the total number of parameters while taking into account the premise and consequent parameters is 105. Product T-norm is selected to be employed in the second Layer, and Gaussian membership functions from Eq. (3) are selected to represent inputs. Fig. 8. Represent membership functions A_{jk} for all inputs after the training process, with j=1,2,3 (number of MF) and k=1,2,3,4 (number of inputs).

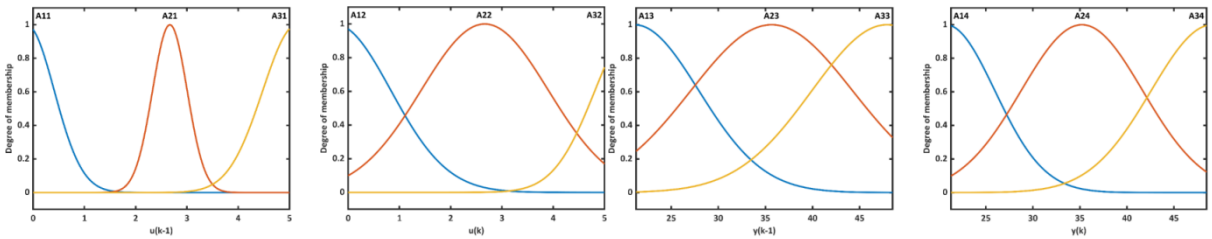

Figure 8 Membership functions *Ajk*, *j*=1,2,3 and *k*=1,2,3,4 – ANFIS model

Although the ANFIS training process was much less time-consuming than the training of all neural networks and in terms of overall MSE this approach showed much better results, Fig.9, with MSE equals to 0.0068. The optimization of ANFIS parameters (by using some other methods than backpropagation and LSE such as metaheuristic) and avoiding grid partition which suffers from *the curse of dimensionality* would probably contribute to even better performance.

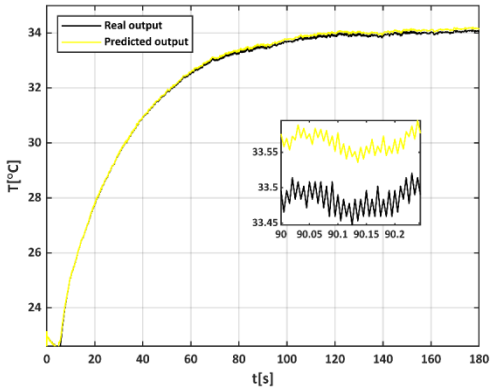

Figure 9 ANFIS model

5. Control systems

Both nonlinear models FFNN and ANFIS have shown better results than traditional linear modelling with delay. Precisely because of this, control will also be discussed through two approaches: the first group of control signals will be based on the FFNN models, and the second on ANFIS. These techniques propose using an inverse model i.e. model where the inputs in the controller are the desired outputs and the outputs from the object. As in the case of identification, all parameters remained unchanged. The data set is divided into training, validation and testing in the ratio 70%-15%-15%, respectively. MSE was chosen as the performance criterion and training was done offline, from pre-recorded data, but network testing was done in real-time, simultaneously comparing the output from the model and the actual output of the object. Neural networks are typically used in direct inverse control in the literature; however, this paper also integrates ANFIS networks.For neural networks, three architectures were chosen with one hidden layer each and five, eight and ten neurons per layer, respectively. The learning rate remained fixed at 0.001, and LM was chosen as the

algorithm. In the case of ANFIS, GP was selected for the initial model, as T norm the product, Gaussian membership functions in the premises, Fig. 11 and a scalar as an independent term in the consequent parameter of the conclusion were used. Controller output can be defined with Eq. (9):

$$u(k) = N\left(y(k+1), y(k), y(k-1), u(k-1)\right). \tag{9}$$

The inverse model is subsequently applied as a controller for the process by inserting the desired output or reference signal yd(k+1) instead of the output y(k+1) [15]. Fig. 10 shows the direct inverse control scheme.

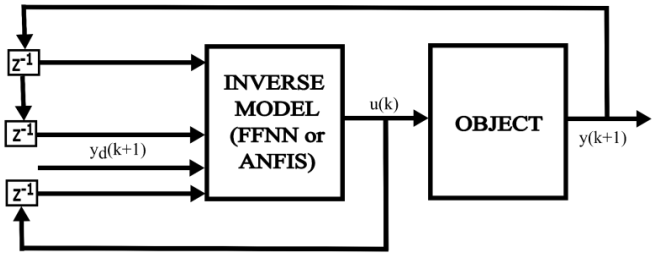

Figure 10 Direct inverse control structure

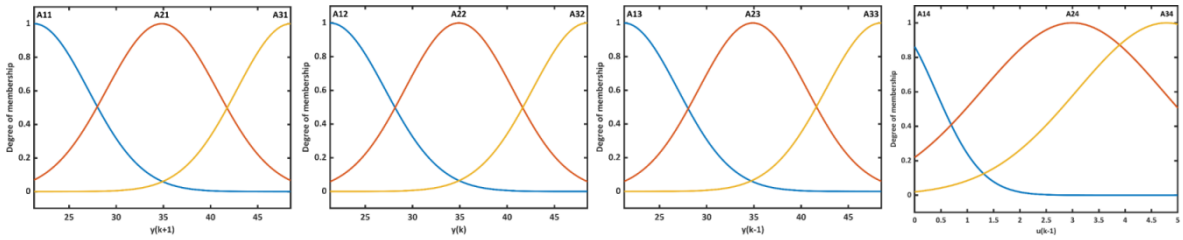

Figure 11 Membership functions *Ajk*, *j*=1,2,3 and *k*=1,2,3,4 – ANFIS controller

In the case of control, increasing the number of neurons in the hidden layer of the network did bring an improvement, which can be seen from Fig. 12.

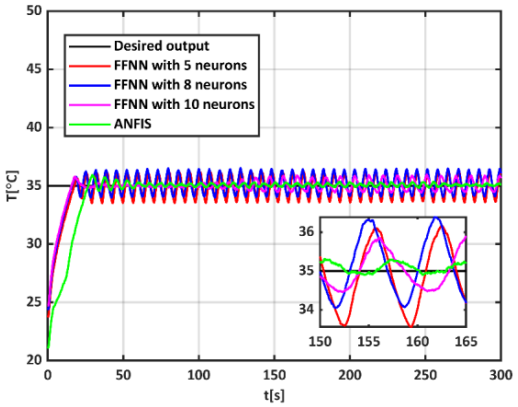

Figure 12 Different controllers

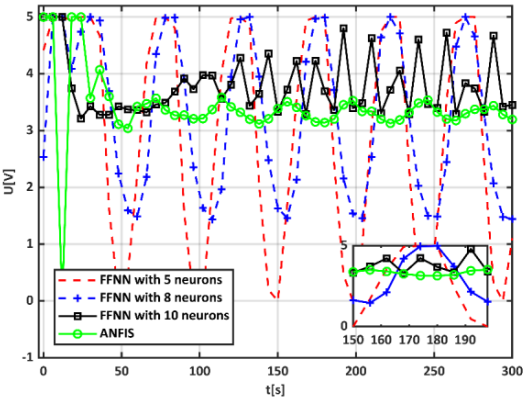

Figure 13 Control signals

A network with 10 neurons makes a smaller overall MSE which equals to 1.4990, compared to networks with 5 and 8 neurons (with MSE equal to 2.4903 and 2.1514). By using ANFIS, the response with the longest rising time is obtained and therefore its total MSE is the highest (5.1548). Although the training itself, as well as the identification process, with ANFIS was shorter than with neural

networks, its operation in the online real experimental environment and time proved to be the slowest at the initial moment. ANFIS, on the other hand, responds significantly better than all FFNN models in terms of smaller oscillations and the error in the steady state, after the transient time. If the system is observed in the steady state, after the transient time ANFIS offers the best results. Control signals are presented on Fig. 13.

6. Conclusion

In this study, the topic of modeling and controlling temperature within a chamber using three temperature sensors has been addressed. After the system was put into operation, a traditional model of transfer function with time delay was developed. The second sensor located in the middle of the chamber was chosen as the reference sensor due to its least noisy response and because the model for this sensor yielded the lowest MSE. As the traditional model performed poorly with changing inputs, four intelligent models were implemented. The first three models utilized the same technique: feedforward neural networks, with varying architectures, while the fourth model was a fuzzy-neuro network, known as ANFIS. Modeling with these advanced techniques produced significantly better results than the standard approach, and these methods were also applied for temperature control. To facilitate control, an inverse model was required, and the structure of direct inverse control was presented. Artificial neural networks, specifically FFNN and ANFIS, were employed as control systems. Increasing the number of neurons in the hidden layer of FFNN improved control performance in terms of MSE, although it introduced more noise into the control signal compared to networks with fewer neurons. As it exhibited the least oscillation amplitude, the FFNN with 10 neurons in the hidden layer was deemed the optimal architecture among artificial neural networks. While ANFIS yielded the best results for systems in a steady state, it performed the poorest in terms of total MSE, primarily due to its slower rise time. In this way, the nonlinear models with two types of networks, FFNN and ANFIS, were created and they demonstrated accuracy across the whole state space. In addition, the control signal was synthesized in both network concepts. FFNN is a well-known and proven technique in DIC, but ANFIS, a type of network in the DIC idea, is much less widely used in the literature; therefore, this research investigates it. The research shows, utilizing the aforementioned approaches in the experiment, that nonlinear modeling and control provide excellent accuracy and can be used to solve these types of problems. Further research could enhance the study by considering other types of artificial neural networks such as radial basis or recurrent networks. ANFIS could be improved by substituting the GP method for finding the initial model with subclustering, which would reduce the number of rules, simplify the system, and likely result in a faster response at the beginning of experiments. Additionally, alternative training algorithms, such as genetic algorithms and particle swarm optimization, or modern metaheuristic algorithms like ant lion optimization (ALO), African vulture optimization algorithm (AVOA), marine predator algorithm (MPA), Sewing Training-Based Optimization (STBO), among others, could be explored to further refine the research and improve control strategies.

Acknowledgment

The results presented here are the result of research supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under Contract 451-03-47/2023-01/ 200105 dated 02/03/2023.

References

[1] Jamil, A. A., *et al.*, Fractional-Order PID Controllers for Temperature Control: A Review, *Energies*, *15* (2022), 10, 3800.

[2] Lu, X., Thermodynamic Model of HVAC Under Building Exterior Wall, *Thermal Science*, *27* (2023), 2A, 983-990.

[3] Jacob A., *et al*., Heat Flow Experiment for MATLAB /Simulink Users, Student Workbook, Quanser Innovate Educate, Accessed 11.8. 2023.

[4] Ahn, H. S., *et al*., Fractional-Order Integral and Derivative Controller for Temperature Profile Tracking, *Sadhana*, *34* (2009), 833-850.

[5] Orman, K., Design of a Memristor-Based 2-DOF PI Controller and Testing of Its Temperature Profile Tracking in a Heat Flow System, *IEEE Access*, *10* (2022), 98384-98390.

[6] Jovanović, R. Ž., Zarić, V. R., Identification and Control of a Heat Flow System Based on the Takagi-Sugeno Fuzzy Model Using the Grey Wolf Optimization Algorithm, *Thermal Science*, *26* (2022), 3, 2275-2286.

[7] Vesović, M., Jovanović R., Heat Flow Process Identification Using ANFIS-GA Model., *Proceedings* Sinteza 2023-International Scientific Conference on Information Technology and Data Related Research, Belgrade, Singidunum University, Serbia, 2023, pp. 44-51.

[8] Muliadi, J., Kusumoputro, B., Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System, *Journal of Advanced Transportation*, 2018 (2018), 1-18.

[9] Ariza-Zambrano, W. C., Serpa, A. L., Direct Inverse Control for Active Vibration Suppression Using Artificial Neural Networks, *Journal of Vibration and Control*, 27 (2020), 1-2, 31-42.

[10] Hamze, S., *et al.*, Estimating heat-transport and time-delays in a heat exchanger. *Proceedings* IEEE Conference on Control Technology and Applications (CCTA) Copenhagen, Denmark, August 21-24, 2018, pp. 1514-1519.

[11] Cifuentes, J., *et al*., Air temperature forecasting using machine learning techniques: a review, *Energies*, *13* (2020), 16, pp. 4215.

[12] Jovanović, R. Ž., *et al*., Ensemble of various neural networks for prediction of heating energy consumption, *Energy and Buildings*, 94 (2015), pp. 189-199.

[13] Jovanović, R. Ž., *et al*., Ensemble of radial basis neural networks with k-means clustering for heating energy consumption prediction, *FME Transactions*, 45 (2017), 1, pp. 51-57.

[14] Jang, J. S., ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23 (1993), 3, pp. 665-85.

[15] Hedjar, R., Online adaptive control of non-linear plants using neural networks with application to temperature control system. *Journal of King Saud University-Computer and Information Sciences*, 19 (2007), pp. 75-94.

Submitted: 08.12.2023
Revised: 31.01.2024
Accepted: 13.02.2024