The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
18–21 September, 2019, Metz, France

NN-SANARX Model Based Control of a Water Tank System Using Embedded Microcontroller Arduino

Vjatseslav Skiparev[1] , Juri Belikov[2] , Eduard Petlenkov[1]
[1] Department of Computer Systems, Tallinn University of Technology,
Akadeemia tee 15a, 12618 Tallinn, Estonia {vjatseslav.skiparev, eduard.petlenkov}@ttu.ee
[2] Department of Software Science, Tallinn University of Technology,
Akadeemia tee 15a, 12618 Tallinn, Estonia juri.belikov@taltech.ee

***Abstract* — The paper is devoted to the application of neural network based SANARX method, implemented on embedded device, for the real-time control of liquid level inside a single water tank. The system is implemented as laboratory prototype with two bottles. The embedded neurocontroller is implemented in Arduino Mega 2560 Developing Board, which has been chosen due to sufficient computational power. Presented microcontroller program is an adaptation of mathematical model designed and verified in MATLAB/Simulink environment.**

Keywords* — *Computational intelligence; embedded control; industrial applications.

I. Introduction

Level regulators are used in industry to maintain a constant fluid pressure, or a constant fluid supply to a process, or in a waste storage [1]. The common examples of possible industrial applications include chemical industry and food processing as well as different irrigation systems like dams, water reservoirs, etc. Through the years various techniques have been used to solve the problem of liquid level control.

Most of these solutions by embedded systems were focused on modeling, analysis, and implementation of simple algorithms, including classical PID and its various modifications [2]. However traditional approach based on PID-type control scheme is not fully applicable when it has to deal with nonlinear features of a tank, especially with multi-loop control tasks [3]. Furthermore, systems become more complex and requirements rise (energy efficiency, reliability, etc.). Therefore, methods based on computational intelligence have started to gain popularity and are applied either solely or in combination with some classical techniques. There are numerous papers addressing application of computational intelligence based techniques to control the liquid-level system, see [4]–[6] for Neural Networks based methods, [7], [8] for Neuro-Fuzzy hybrid ANFIS, and the references therein.

Recently, some authors have addressed a problem of implementing control algorithms in embedded devices as a step toward practical applications. For example, Arduino boards have several examples control of real processes. The classical PID successful practiced for position control of DC motor using the combination of Arduino Mega and Matlab [9]. Fuzzy-logic based PID was implemented on embedded computer-friendly ARM and combined with Arduino Uno for control of level inside a single water tank system [10]. Another solution consider Arduino board as a monolith regulator has been successfully tested for speed tracking of induction motors using modified Ziegler-Nichols PID and Fuzzy-logic controller [11]. The work [12] shows the inverse control model used to optimize performance of PV system.

In this paper we join the above trend by exploring abilities of Simplified Additive Nonlinear AutoRegressive eXogenous (SANARX) model implemented in Arduino board to control a water level in a tank system. This paper can be seen as a natural continuation of our previous research in [5] focusing on a hardware implementation.

The paper is organized as follows. Section II is devoted to description of the physical system. In Section III the control algorithm is briefly sketched, complemented by details on the hardware implementation. Sections IV and V present the simulation and experimental results of the proposed technique. Conclusions and a brief discussion are presented in the last section.

II. Process Identification and Physical System Description

Since the real water tank system has only two-state outflow valves, dynamics of described system can be approximates by the first-order nonlinear differential equation with constant flow resistance as

$$\frac{\mathrm{d}H}{\mathrm{d}t} = K_{in}u(t) - K_{out}\sqrt{y(t)}, \qquad (1)$$

where $u(t)$ represents inflow water, $y(t)$ is the outflow water, K_{in} is the coefficient of input fluid flow into the tank, and K_{out} is the output value.

The assembled prototype in Fig. 1 consist of filling and sinking bottles. The measuring of liquid level is performed using a low-cost ultrasonic distance sensor HC-SR04. The 6 V dosing pump performs role of actuator that transports fluid from reservoir to the upper bottle. Pump regulates water level according to the two-positional relay. To sink extra liquid into reservoir, two-positional 5V electromagnetic solenoid valve can be used. The USB cable is used to supply the power to equipment, except for the pump, which has a separate power source. In addition, we use a cable for data exchange between board and Simulink model.

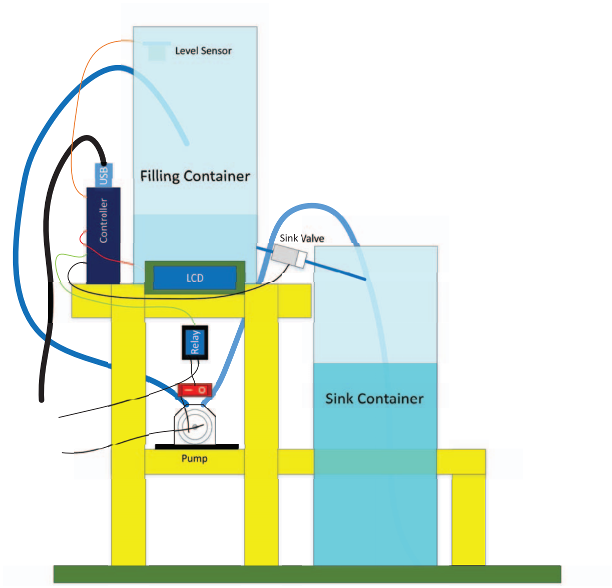


Figure 1. Water tank prototype.

The experimental data was collected by filling and sinking water in the range of 40 to 160 mm as shown in Fig. 2. The coefficients in (1) can be identified as $K_{in} = 0.243$ and $K_{out} = 0.0165$.

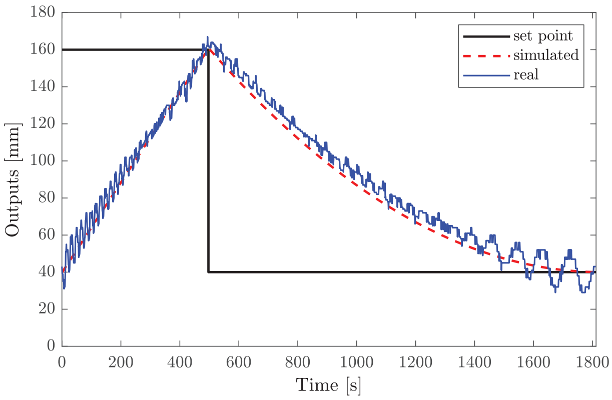


Figure 2. Identification results.

III. Control System and Software Implementation of Algorithm

To design a controller based on the idea of dynamic feedback linearization technique, in this paper we propose to utilize the so-called neural networks based Simplified Additive NARX model [5]. The considered process has relatively simple dynamics, and therefore, neural network consist of two input/output pairs for each hidden sub-layer, see Fig. 3. The first sub-layer has a pure linear activation function, whereas the second sub-layer has tansig function. The output neuron again contains pure linear function. The final model is given by

$$\begin{aligned} y(t) = 105.78\big((v(t) - C_2(\text{tansig}(W_{21}y(t-1)) \\ + \text{tansig}(W_{22}u(t-1))) \\ - y(t)C_1W_{11})(C_1W_{12})^{\mathsf{T}}\big), \end{aligned} \tag{2}$$

where C_n and W_n are the output and input matrices of synaptic weights. Observe that the equation (2) was adjusted for Arduino microcontroller amplifying the output signal by 105.78 times. The value was chosen in accordance with desirable error band ± 5 mm from the set value.

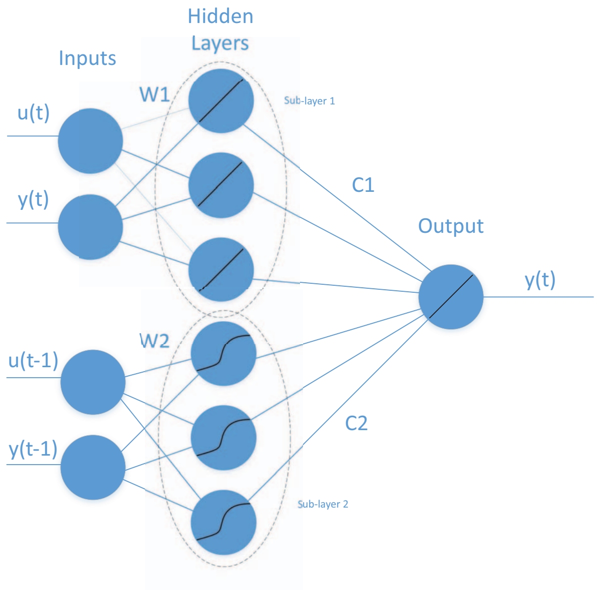


Figure 3. Structure of implemented NN-SANARX model.

Fig. 4 depicts block scheme of a control algorithm implemented in microcontroller. The diagram illustrates major feature of NN-SANARX, where the current level of water $y(t)$ has influence identified weights for hidden neurons with linear function, meanwhile the previous sensor value $y(t-1)$ has array of weights for nonlinear neurons. The value of control signal $u(t)$ depends on difference between reference value $v(t)$ and $y(t-1)$. Controlled devices have two states—if $u(t)$ is positive then the filling process is performed, otherwise is going an opposite action. The previous control signal $u(t-1)$

reserved to correct output signal of NN-SANARX controller in control loop.

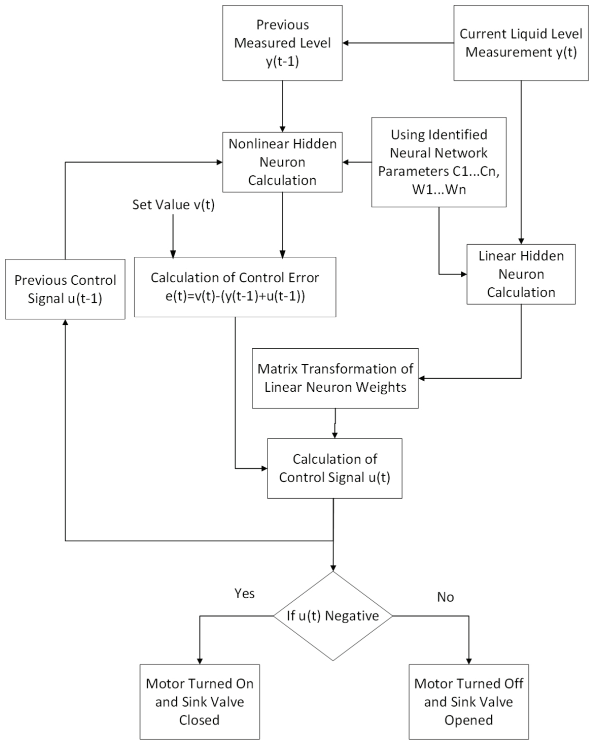


Figure 4. Block-diagram of NN-SANARX algorithm in Arduino.

Pseudocode in algorithm 1 is written in C/C++ language. It demonstrates a programming adaptation of SANARX Neural Network from Fig. 3 with identified synaptic weights W_n and C_n. The feature of code is direct separation of inverted matrix T_i and hidden sublayers to linear hL_i and nonlinear hNL_i as arrays. In last stage nonlinear output u_{NL}, linear output u_L and matrix transformation T_{Sum} sums to output u_t according to equation (2). In addition, code reflects discretization of control output u_t from Fig. 4 with deadband zone -20.0—20.0 that transforms to direct control of pump and valve.

A. Combined and Independent Control System

Combined control system in Fig. 5 presents fusion of Master Controller—model created in MATLAB/Simulink and Slave Controller—Arduino program which receives control signal directly from Master. In this system Master/Slave architecture organized in the following way: microcontroller receives sensor signal, transforms to integer value, and sends it to the control model through the Serial Port. Then, the model calculates control output value and sends discrete signal to the board, that activates actuators.

The independent control system is organized in a very similar manner to the one depicted in Fig. 5, except for

Algorithm 1 Implementation of NN-SANARX in Arduino.

Input: v_t, y_t, y_{t-1}, C_1, C_2, W_{11}, W_{12}, W_{21}, W_{22}
Output: u_t, u_{t-1}

while t **do**
  $y_{t-1} \leftarrow y_t$;
  for $i \leftarrow 3$ `to` 0 **do**
    $T_i = C_{1,i}W_{12,i}$;
  end for
  for $i \leftarrow 0$ `to` 3 **do**
    $hL_i = y_t W_{11,i}$;
    $hNL_i = \text{tansig}(y_{t-1}W_{21,i}) + \text{tansig}(u_{t-1}W_{22,i})$;
  end for
  $o_{NL} \leftarrow 0$;
  $o_L \leftarrow 0$;
  $T_{sum} \leftarrow 0$;
  for $i \leftarrow 0$ `to` 3 **do**
    $o_{NL} \leftarrow o_{NL} + C_{2,i}hNL_i$;
    $o_L \leftarrow o_L + C_{1,i}hL_i$;
    $T_{sum} \leftarrow T_{sum} + T_i$;
  end for
  $u_t = 105.78(((v_t - o_{NL}) - o_L)T_{sum})$;
  $u_{t-1} = u_t$;
  if $u_t > 20.0$ **then** Pump ON, Valve OFF
  end if
  if $u_t < -20.0$ **then** Pump OFF, Valve ON
  end if
end while

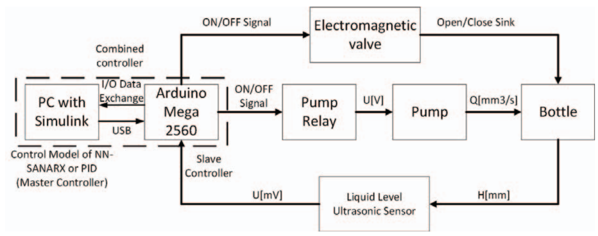


Figure 5. Structure of combined control system.

microcontroller performs the direct control in the prototype. In this architecture MATLAB/Simulink is used only for taking parameters from simulated model: synaptic weights for neurocontroller and K_p, T_i, T_d parameters for the PID controller. The latter is implemented for comparison purposes.

IV. Simulation Results

We start with simulation experiments that allow to validate control performance of Classical PID and Neural Networked SANARX for two cases of constant and single step set values. In order to make simulation more close to reality, the input signal is rounded to integer values. This scenario demonstrates an approximation of real sensor signal without ultra-sound oscillations and extra noise. Figures 6 and 7 present the simulation results for the proposed control schemes at constant and step-down set value, before and after the quantization. In most performance indices the NN-based control has better result than the classical PID controller, see Table I.

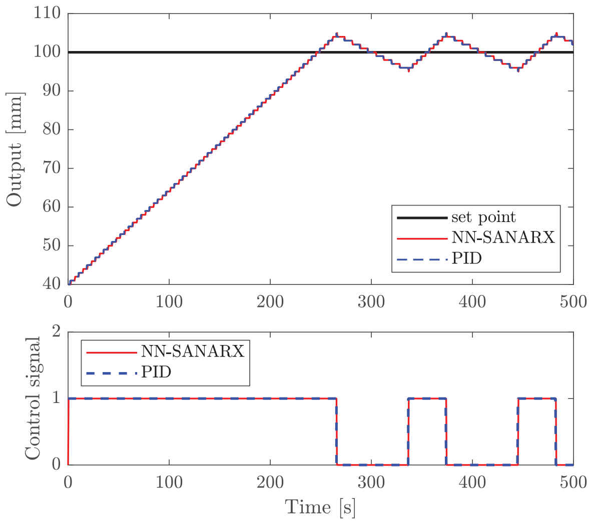


Figure 6. Simulation of control in constant set value with and without quantized signal.

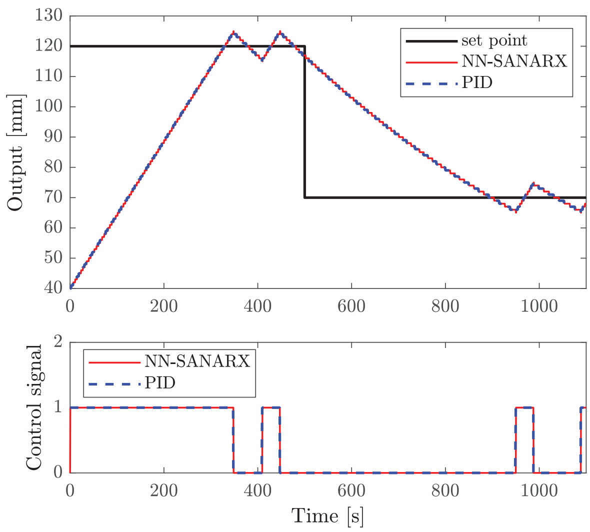
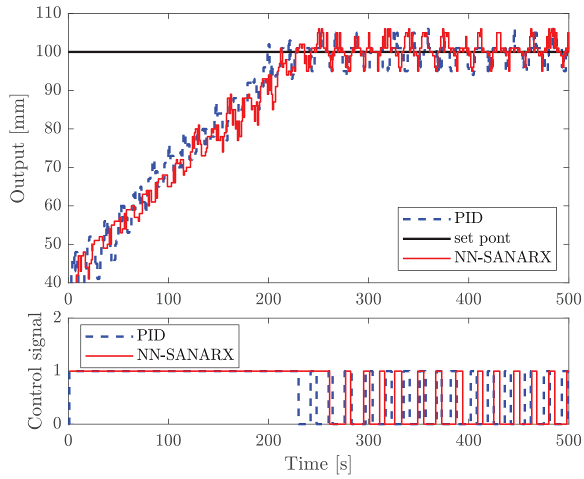


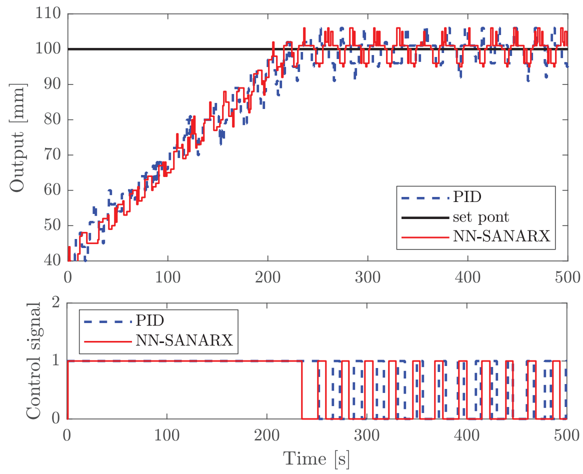
Figure 7. Simulation of control in step-down set value with and without quantized signal.

V. Experimental Results

Here, we present the real experimental results for NN-SANARX and PID control algorithms. Mentioned algorithms have two implementations: (i) software in Matlab/Simulink and (ii) hardware created with programming language C/C++. Both approaches have equal control parameters. Figures 8 and 9 demonstrate comparison of algorithms performance on the real prototype in steady-state conditions.



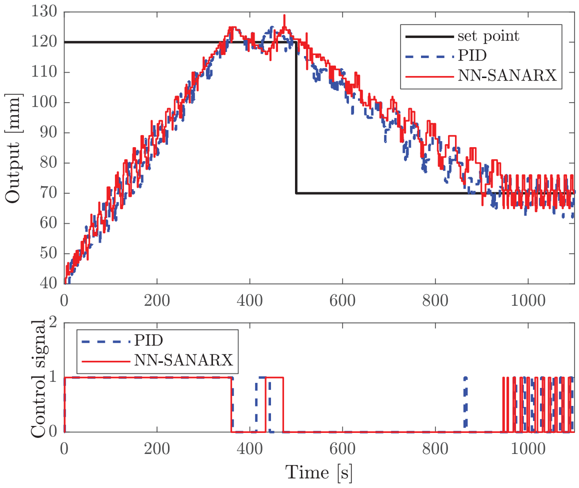
(a) Arduino



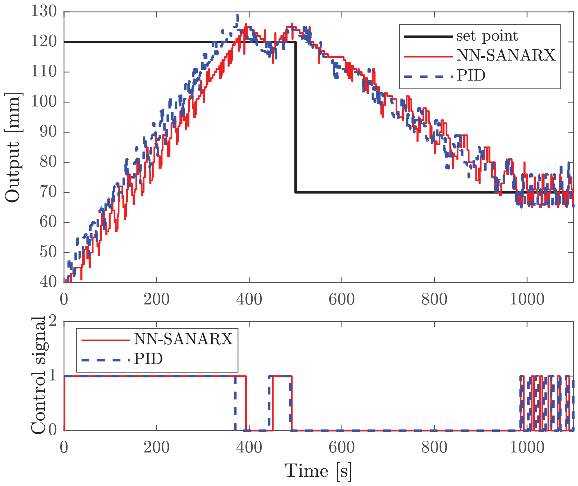
(b) Simulink

Figure 8. Experimental results using Arduino and Simulink for the constant set value.

All measurements performed by Ultra-Sonic sensor HC-SR04. Note that this model during measurements generates significant oscillations and has tendency arising of chaotic signals even at the same water level. As a result, measurements in repeated experiments have observable differences. Therefore, the overall control performance cannot fully display the real picture. On the other hand, except for the problematic sensor, both algorithms have similar reaction in the steady-state. Note that in most cases Neural Network control algorithm has smaller MSE, but in some experiments larger SSE. All displayed experimental results contain local *peaks* of noise, which were filtered out during performance calculations. Observe that Arduino has less MSE and SSE in experiment with step set point, but in experiment with linear set point larger than Simulink SSE. Mostly this difference happens since the sensor is less accurate for 100 mm water level than



(a) Arduino



(b) Simulink

Figure 9. Experimental results using Arduino and Simulink for the step-down set value.

for higher levels. Another factor is control performed by Simulink, which creates additional delay for data receiving. Table I summarizes the comparison results for NN-SANARX and PID control algorithms.

VI. Discussion and Conclusions

In this paper we demonstrate the possibility of implementing a classical dynamic output feedback linearization algorithm combined with neural network based SANARX model on the embedded devices such as Arduino Mega Development Board. The proposed algorithm is used to control water level inside real prototype of a single tank system. We present comparison of efficiency of mentored control architectures by graphs and by calculated MSE and SSE. Arduino as independent system has better performance by absence of problems tied with communication, includes time delay and additional influence of noise in communication cable. According to this facts and

Table I. Statistical indices in steady-state.

Approach	Line (MSE)	Line (SSE)	Step (MSE)	Step (SSE)
sim. PID	8.24	0.09	7.93	0.6
sim. NN-SANARX	7.9956	0.09	7.33	0.57
sim. quantized PID	7.28	0.38	6.93	0.64
sim. quantized NN-SANARX	7.15	0.37	6.94	0.56
independent PID	8.19	0.29	11.71	0.82
independent NN-SANARX	7.65	0.72	11.18	0.79
combined PID	8.42	0.17	12.66	0.89
combined NN-SANARX	7.91	0.33	11.39	1.11

illustrated results, we may conclude that the proposed control scheme can be used in real time applications. The solution can be applied for other similar existing control system, and can be used as a start for potential work related to various Neural Networks based control techniques on embedded devices, including industrial PLC. It is important to mention that despite the high level of noise generated by sensor, both algorithms appeared to be robust.

Recall that the actual control speed in both simulations and hardware implementation is almost the same. Therefore, Arduino Mega 2560 provides enough computational power to regulate process using artificial neural networks. In the initial version of the algorithm we considered the idea of a direct import of retrained synaptic weights from MATLAB to Arduino. However, we have faced two problems: (i) it is difficult to export multiple float values by serial port and (ii) it is necessary to keep the connection between PC and board at each prototype launch. Another idea would be to implement the retraining algorithm directly on the embedded device. These will make subject for the future research.

References

[1] W. C. Dunn, *Fundamentals of Industrial Instrumentation and Process Control*. McGraw-Hill Professional, 2005.

[2] K. Sundaravadivu, V. Jeyakumar, and K. Saravanan, "Design of fractional order PI controller for liquid level control of spherical tank modeled as fractional order system," in *IEEE International Conference on Control System, Computing and Engineering*, 2011, pp. 522–525.

[3] I. S. Isa, B. C. C. Meng, Z. Saad, and N. A. Fauzi, "Comparative study of PID controlled modes on automatic water level measurement system," in *International Colloquium on Signal Processing and its Applications*, 2011.

[4] M. U. Adam, M. H. Ali, and S. L. Dhar, "Determination of the performance of neural PID, fuzzy PID and conventional PID controllers on systems with several overshoots," *International Journal of Advance Research in Science and Engineering*, vol. 4, no. 1, pp. 771–778, 2015.

[5] J. Belikov and E. Petlenkov, "NN-SANARX model based control of a multi tank liquid-level system," *International Journal of Computational Intelligence Systems*, vol. 8, no. 2, pp. 265–277, 2015.

[6] D. Wu, F. Karray, and I. Song, “Water level control by fuzzy logic and neural networks,” in *Conference on Control Applications*, 2005.
[7] X.-K. Wang, X.-H. Yang, G. Liu, and H. Qian, “Adaptive neuro-fuzzy inference system PID controller for SG water level of nuclear power plant,” in *International Conference on Machine Learning and Cybernetics*, 2009.
[8] K. M. Passino, *Perspectives in Control: New Concepts and Applications*, NJ, Ed. IEEE Press, 2001.
[9] O. Ulkir, I. Ertugrul, and N. Akkus, “Position control of a DC motor using Matlab,” *International Journal of Research in Engineering and Advanced Technology*, vol. 4, no. 5, pp. 37–40, 2016.
[10] S. Krivić, M. Hujdur, A. Mrzić, and S. Konjicija, “Design and implementation of fuzzy controller on embedded computer for water level control,” in *International Convention MIPRO*, 2012.
[11] T.-J. Ho and C.-H. Chang, “Robust speed tracking of induction motors: An arduino-implemented intelligent control approach,” *Applied Sciences*, vol. 8, no. 159, 2018.
[12] C. R. Algarín, D. S. Hernández, and D. R. Leal, “A low-cost Maximum Power Point Tracking system based on neural network inverse model controller,” *Electronics*, vol. 7, no. 1, 2018.