International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 2, April 2025, pp. 1499~1510
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i2.pp1499-1510

Developing an algorithm for the adaptive neural network for direct online speed control of the three-phase induction motor

Ahmad J. Al-Mahasneh[1], Samer Z. Salah[2], Jasim A. Ghaeb[1], Mohammed Baniyounis[1]
[1]Department of Mechatronics Engineering, Faculty of Engineering, Philadelphia University, Amman, Jordan
[2]Department of Mechatronics Engineering, School of Engineering, The University of Jordan, Amman, Jordan

Article Info

Article history:

Received Jun 30, 2024
Revised Oct 24, 2024
Accepted Dec 14, 2024

Keywords:

Adaptive neural control
General regression neural network
Induction motor
Proportional integral derivative control
Speed control

ABSTRACT

In this paper, an online adaptive general regression neural network (OAGRNN) is presented as a direct online speed controller for a three-phase induction motor. To keep the induction motor running at its rated speed in real-time and under a variety of load conditions, the speed error and its derivative are continuously measured and fed back to the OAGRNN controller. The OAGRNN controller provides the inverter with the control signal it needs to produce the proper frequency and voltage for the induction motor instantly. Notably, the OAGRNN controller demonstrated remarkable performance without the need for a learning mode; it was able to track the desired motor speed, starting its operation from scratch. A setup utilizing a three-phase induction motor has been developed to show the high capacity of OAGRNN for tracking the desired speed of the motor while subjected to the varied load torque. The performance of OAGRNN is examined in two phases: the MATLAB simulation and the experimental setup. Furthermore, when the OAGRNN performance is compared with that of the proportional integral (PI) controller, it demonstrates its outstanding ability and superiority for online adjustments related to the three-phase induction motor's speed control.

Corresponding Author:

Mohammed Baniyounis
Mechatronics Engineering Department, Faculty of Engineering, Philadelphia University
Amman, Jordan
Email: mbaniyounis@philadelphia.edu.jo

1. INTRODUCTION

The variable frequency drive (VFD), which allows for the adjustment of motor speed through changing stator frequency, is one technique for varying the speed of three-phase induction motors. To improve speed control, the authors in [1] used closed-loop fuzzy logic control. Examining and comparing the dynamic speed with the open loop, according to the simulation results, using fuzzy logic control increased transient reaction performance by 13%. Due to its adaptability and affordability, a programmable logic controller (PLC) was used in the induction motor speed control [2]. The solid-state PLC replaces mechanical relays in industrial applications because it performs discrete or sequential logic in an industrial environment. In [3], it was suggested that an induction motor drive based on an artificial neural network (ANN) be used. Comparing the performance of the ANN to the traditional proportional integral (PI) controller, the ANN showed more improvements. Hussein [4] using space vector pulse width modulation (SVPWM), a supply voltage with variable amplitude and frequency is used to adjust the three-phase induction motor's speed. The feedback closed-loop control uses the integral controller, and an acceptable motor responds to the varied torque with a speed that is achieved. Farhi *et al.* [5] proposed two algorithms to obtain high-performance control of induction motor drives. The first approach is the super twisting algorithm (STA) to reduce the chattering effect and improve control accuracy. In contrast, the second one is the barrier super twisting algorithm (BSTA) to eliminate the chattering phenomenon by providing continuous output control signals.

Numerous engineering applications have made use of neural networks, including the engineering control perspective in real-time operations [6]–[8]. By developing computer programs or systems that perceive, acquire knowledge, and behave like people, artificial intelligence (AI) aims to replicate human intelligence in technology to obtain quick and efficient responses [9]–[11]. For instance, in [12], multiple neural networks (NNs) are trained online to approximate the dynamic model of the induction motor, and then a nonlinear control is designed using these approximations. The NNs are used as dynamic estimating and do not directly control the motor speed. This situation would require more computational complexity to control the problem. Also, in [13], NNs have been used for online estimation of the stator and rotor resistances of the induction motor for speed sensor-less indirect vector-controlled drives. While it is a new approach to control, it is model-based, and the NN is not used as a controller but instead as an estimator that will be used for the controller design based on the state-space of the system. In study [14], in the first stage, model-predictive control (MPC) is used to collect the training data for the NN and then once these data are used to train the NN, the NN is used as a controller to track the voltage of the induction motor.

Although the proposed approach has shown good results in the dynamic and steady-state responses, it is still dependent on the MPC for the data collection. Also, no online adaptation of the NN weights is accommodated. In many other applications, NN is used as an estimator for speed [15]. The proportional integral derivative (PID) controller is extensively utilized in the industrial sector to regulate linear systems. However, when faced with nonlinear behavior, its effectiveness can be limited. To enhance the capability of PID controllers, particularly in the case of doubly fed induction motors, researchers have integrated artificial intelligence through optimization algorithms. Mahfoud *et al.* [16] proposed the utilization of an ant colony optimization (ACO) algorithm to fine-tune the gains of the PID controller and effectively manage torque and speed in the doubly fed induction motor (DFIM). The implementation of the intelligent ACO-direct torque control (DTC) control using MATLAB-Simulink demonstrated satisfactory performance in terms of speed, stability, precision, and torque ripples, surpassing the capabilities of conventional DTCs. Mahfoud *et al.* [17] implemented a genetic algorithm (GA) to fine-tune and optimize the parameters of the PID controller.

This approach aimed to address multiple objectives, including mitigating speed overshoot, decreasing response time, reducing the rate of total harmonic distortion (THD) in the stator and rotor currents, and minimizing both the rejection time of speed and the amplitude of torque and flux ripples in the DFIM. Brushless direct current (BLDC) motors are extensively employed in mechanical applications due to their efficiency, suitable torque, and compact size [18]. However, achieving optimal performance and tuning parameters for maximum force output is challenging with a basic custom PID controller. Mahmud *et al.* [19] implemented an adaptive PID controller that utilizes an additional feedback signal to address non-linearity, parameter variations, and load fluctuations in the BLDC motor drive system. The results demonstrate that adaptive PID controllers are suitable for dynamic movements and effectively minimize parameter changes. Premkumar *et al.* [20] suggested a fuzzy-anti windup-PID (FAW-PID) controller for speed control, which reduces the saturation effect on the speed response of the motor. The induction motor's control system parameters, including the PI controller, the FAW-PID controller, and the suggested controller, are measured and compared. The FAW-PID controller beats the other controllers in terms of performance. Boukhalfa *et al.* [21] compared three hybrid techniques for DTC of the dual star induction motor (DSIM) drive. Proportional integral derivative-particle swarm optimization (PID-PSO), fuzzy-PSO, and GA-PSO are used to improve the speed-regulated loop behavior of the DSIM. As a result, fuzzy-PSO is the best solution. The primary function of fuzzy-PSO is to reduce high torque ripples, improve rising time, and avoid disturbances that impact drive performance.

Load torque changes of the induction motors cause noticeable deviations in the desired speed if there is no effective controller. In situations where load varies, an adaptive controller could be a good option for maintaining the speed of induction motors. Based on that and relying on analyzing the previous literature, an online adaptive general regression neural network (OAGRNN) [22] has not been utilized in direct online speed control of a three-phase induction motor, and thus, it is proposed in this work to control the speed of three phase induction motor directly and in the online learning mode where no previous training data or model of the induction motor is required for the controller design. Instead, it learns from scratch in real-time to provide the inverter with the control signal it needs to produce the proper frequency and voltage for the induction motor instantly. Different load torque variations are applied on the induction motor to test the OARNN controller's ability in online adaptations.

The OARNN controller's performance is examined in both MATLAB simulation and experimental setup. The rest of the paper is organized as follows: a literature review is given in section 2. In section 3, the OAGRNN is proposed to track the desired speed of the induction motor while subjected to the varied load torque. Additionally, the PID controller is explained. Simulation and experimental results, as well as a comparison between the proposed OAGRNN controller and PI controller, are given in section 4. Finally, the contribution and conclusion of this work are given in section 5.

2. THREE-PHASE INDUCTION MOTOR

When a three-phase supply voltage is applied to the stator of an induction motor, a three-phase set of stator currents is flowing. These currents produce a magnetic field rotating at a synchronous speed, which depends on the system frequency applied to the stator in hertz and the number of poles in the machine [23], [24]. A part of this flux is mutual with the rotor, and therefore, the energy is induced and transferred across the air gap to the rotor. Electro-motive-force (e.m.f.) is produced in the rotor as long as there is a slip between the rotor and the stator. The equivalent circuit of the rotor circuit is shown in Figure 1(a), and the rotor current is given by (1):

$$I_2 = \frac{E_2}{\sqrt{\left(\frac{r_2}{s}\right)^2 + \mathrm{X}^2{}_2}} \tag{1}$$

Where I_2 is rotor current, E_2 is rotor e.m.f, s is rotor slip, r_2 and X_2 are the resistance and leakage reactance of the rotor. After separating the rotor resistance term (r_2/s) into two series combinations, the equivalent circuit of the rotor circuit is shown in Figure 1(b), and (1) becomes:

$$I_2 = \frac{E_2}{\sqrt{\left(r_2 + \frac{r_2(1\text{-s})}{s}\right)^2 + \mathrm{X}^2{}_2}} \tag{2}$$

The rotor losses are obtained by multiplying the first part of the term $(r_2 + \frac{r_2(1\text{-s})}{s})$ by I_2, and the mechanical power is obtained by multiplying the second part by I_2. The losses in the rotor circuit can also be given by (3):

$$I^2{}_2 r_2 = \mathrm{sE}_2 I_2 \cos\phi = \mathrm{sP}_2 \tag{3}$$

and the second component, which is mechanical power, is represented by (4):

$$P_m = (1\text{-s})P_2 \tag{4}$$

where P_2 is rotor input power, P_m is mechanical power, and ϕ is rotor phase angle.

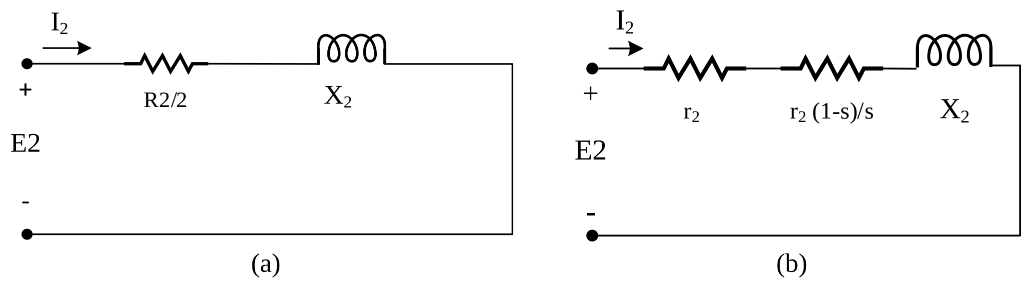


Figure 1. Equivalent circuit of the rotor (a) before and (b) after splitting the resistance

The rotor torque or induced torque (T_{ind}) is given by (5):

$$T_{\mathrm{ind}} = \frac{P_m}{\omega_r} \tag{5}$$

or;

$$T_{\mathrm{ind}} = \frac{(1\text{-s})P_2}{(1\text{-s})\omega_s} = \frac{P_2}{\omega_s} \tag{6}$$

where ω_s is stator synchronous speed (r/s) and ω_r is rotor speed (r/s). Equation (6) concludes that the rotor torque depends on rotor input power and the stator synchronous speed. One of the most common techniques for controlling the speed of three-phase induction motors is the use of a voltage source inverter. It is used with the motor to control voltage and frequency. The pulse-width-modulation (PWM) method is used inside the inverter to provide the required supply frequency and voltage. The supply voltage to frequency ratio must remain consistent across the speed range to maintain a steady electromagnetic torque. The adjusted frequency and voltage are fed to the motor for speed control.

3. ONLINE ADAPTIVE GENERAL REGRESSION NEURAL NETWORK

The objective of the proposed controller is to fix the speed of the rotor at a desired value ω_{rd}(r/s) even when load variations exist. In this work, online adaptive OAGRNN [22] is proposed to perform the speed control task of the three-phase induction motor. The induction motor rotor speed tracking error is defined as (7):

$$e(t) = \omega_{\mathrm{rd}} - \omega_r \tag{7}$$

where ω_{rd} is the desired speed and ω_r is the actual speed of the motor. The structure of the suggested OAGRNN is depicted in Figure 2, consisting of three layers: the input, hidden, and output layers. The output of OAGRNN can, with n-hidden layer size and m-inputs, be expressed as (8):

$$u(t) = W^T h(z(t)) \tag{8}$$

where $u(t)$ is the control signal, $W \in R^{n\times 1}$ is a vector of the output weights of OAGRNN and $h(z) \in R^{n\times 1}$ is a vector of the outputs of the hidden layer of OAGRNN. $h(z)$ is defined as (9):

$$h(z) = exp(\frac{-||W_i - z(t)||^2}{2\sigma^2}) \tag{9}$$

where $W_i \in R^{m\times n}$ is a matrix of the input weights of OAGRNN and $z(t) = [e(t), e'(t)]$ and $\sigma \in (0, \infty)$ is a hyperparameter of OAGRNN called the spread parameter, where $e(t)$ and $e'(t)$ are the motor error and its derivative.

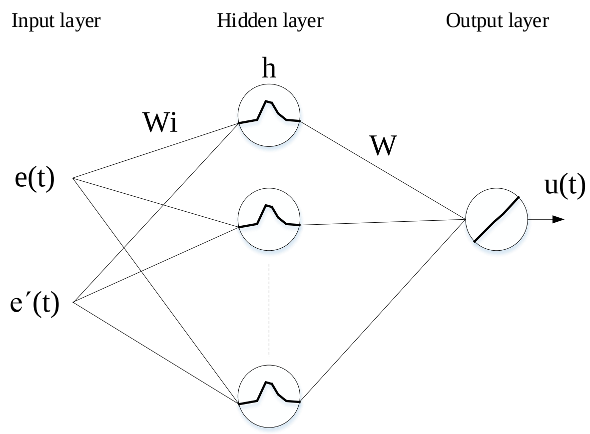


Figure 2. OAGRNN structure

3.1. Online adaptive speed control for the thee-phase induction motor

In this work, the OAGRNN controller approximates the control law $u(t)$ to match the desired speed even with varying load torques. Thus, the OAGRNN adapts its weights online to keep the speed fixed. Therefore, a weights adaptation rule is required. The adaptation rule in this work is designed as (10):

$$W(t+1) = W(t) - \gamma h(z(t))(e(t) + e^{'}(t)) \tag{10}$$

where γ is an online positive adaptation gain for OAGRNNs weights. The closed-loop block diagram is depicted in Figure 3. OAGRNN algorithm is depicted in Algorithm 1.

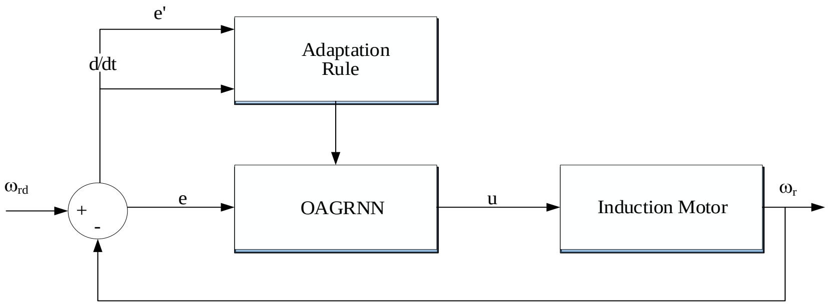


Figure 3. Block diagram of the closed-loop system

Algorithm 1. OAGRNN controller algorithm

```
1. INITIALIZE ωrd(t), γ, Wi= [-1:0.1:1; -1:0.1:1], =[ZEROS(SIZE(Wi,1))]
2. FOR t=0 to t_final, OBSERVE ωr(t) and CALCULAT z(t) = [e(t), e'(t)]
      2.a CALCULATE h(z) USING: h(z) = exp(-||Wi - z(t)||^2 / 2σ^2)
      2.b UPDATE W USING: W(t+1) = W(t) - γh(z(t))(e(t) + e'(t))
3. ALCULATE u(t) USING: u(t) = W^T h(z(t))
```

3.2. Stability analysis of OAGRNN

To analyze the stability of the OAGRNN controller, the Lyapunov direct method is utilized as follows: Firstly, let us define a Lyapunov function $V(t)$ as (11) [25]:

$$V(t) = \frac{1}{2}e^2(t) + \frac{1}{2}\widetilde{W}(t)^T\widetilde{W}(t) \tag{11}$$

Where $e(t)$ is the tracking error, and $\widetilde{W}(t)$ is the OAGRNN weighting error, and it is defined as (12):

$$\widetilde{W}(t) = \hat{W} - W(t) \tag{12}$$

Where $\hat{W}$ is an ideal constant weight vector that makes OAGRNN approximate the control law arbitrary small error ϵ according to the universal approximation theorem of single hidden layer neural networks, and $W(t)$ is the current weight vector of OARGRNN.

Taking the first-time derivative of $V(t)$ as (13):

$$V'(t) = e(t)e'(t) + \frac{1}{2}\widetilde{W}^T(t)\widetilde{W}(t) \tag{13}$$

Utilizing Cauchy-Schwarz inequality, (13) can be rewritten as (14):

$$V'(t) < 2e^2(t) + 2e'^2(t) + |\widetilde{W}|'^2(t) + |\widetilde{W}|^2(t) \tag{14}$$

Substituting (10) into (14) yields:

$$V'(t) < 2e^2(t) + 2e'^2(t) + \gamma^2(h(z(t))(e(t) + e'(t)))^T h(z(t))(e(t) + e'(t)) + |W|^2(t) \tag{15}$$

Because $\text{h}(z(t)) \leq n$ since the Gaussian activation function is utilized in the hidden layer of OAGRNN, n is the number of hidden neurons in OAGRNN, (15) can be simplified as (16):

$$V'(t) < 2e^2(t) + 2e'^2(t) + \gamma^2 n^2(e^2(t) + e'^2(t) + 2e(t)e'(t) + |\widetilde{W}|^2(t)) \tag{16}$$

To ensure Lyapunov stability, $V'(t) < 0$ and thus, (16) becomes:

$$2e^2(t) + 2e'^2(t) + \gamma^2 n^2(e^2(t) + e'^2(t) + 2e(t)e'(t) + |\widetilde{W}|^2(t)) < 0 \tag{17}$$

To achieve uniform ultimate boundness, the learning rate γ should be selected as (18):

$$\gamma > -\sqrt{\frac{2e^2(t)+2e'^2(t)}{e^2(t)+e'^2(t)+2e(t)e'(t)+|\widetilde{W}|^2(t)}} \tag{18}$$

According to the universal approximation theorem, $|\widetilde{W}|^2(t) \leq \varepsilon^2$, where is a small arbitrary positive error threshold and thus, (18) becomes (19).

$$\gamma > -\sqrt{\frac{2e^2(t)+2e'^2(t)}{e^2(t)+e'^2(t)+2e(t)e'(t)+\varepsilon^2}} \tag{19}$$

3.3. Proportional integral derivative controller

In this work, in both the simulation and experimental settings, a PI controller is also used to adjust the induction motor speed in order to compare the performance of the OAGRNN controller. In the industry, the PID controller is utilized to control a wide range of process variables. One way to describe it would be as an accurate controller. In the ideal type [26], the proportional gain constant (K_c) is uniformly applied to all components: proportional (P), integral (I), and derivative (D) terms, and the corresponding control function can be expressed as (20):

$$u(t) = K_c\left(e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt}\right) \tag{20}$$

Where K_c is proportional gain, T_i is integral time, T_d is derivative time. In the parallel PID type, every action parameter (K_c, T_i, and T_d) in the PID is independent of the others. It looks promising, suggesting that alterations to the controller would only have an impact on a specific aspect of its functionality. But sometimes, it is better if the gain parameter affects the three control actions in the same way. In this type, the control function can be represented as (21).

$$u(t) = K_p e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt} \tag{21}$$

4. RESULTS AND DISCUSSION

Two tests were carried out to evaluate the effectiveness of the suggested OAGRNN controller. While the second test was conducted in the laboratory using the practical model, the first test was conducted using the simulation model created in MATLAB. The simulation and experimental results are presented in the two subsections below.

4.1. Simulation results

To demonstrate the superior performance of the OAGRNN controller and compare it with the PI controller, a three-phase induction motor Simulink model was employed, as illustrated in Figure 4. The model was simulated with an Intel Core i7 processor running at a clock speed of 2.7 GHz and 16 GB RAM. Ode3 fixed step solver is used in this simulation and the simulation time is 4 seconds. The adaptation gain of the OAGRNN controller is 0.1. The gain of PI, i.e., K_p and K_i are tuned to be 0.15 and 0.1 respectively. The control frequency is limited between 48.5 and 51.5 Hz. The motor speed is controlled at 1400 rpm.

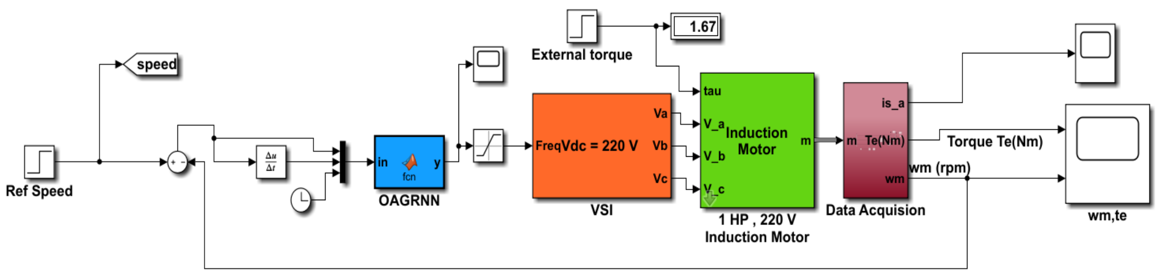
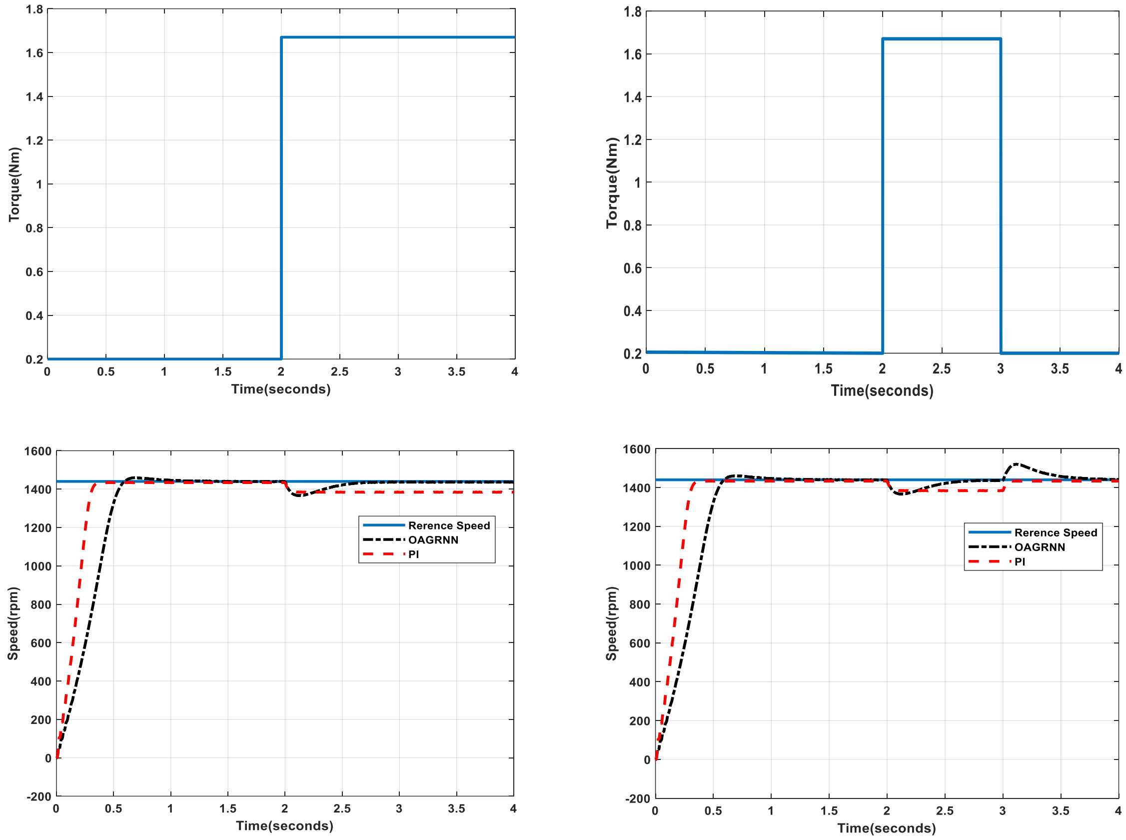


Figure 4. Simulink model of the three-phase induction motor controlled by the OAGRNN controller

Three load scenarios are considered in the experiment. In the first scenario, the torque was increased from 0.2 to 1.67 N.m; the response of both PI and OAGRNN controllers is demonstrated in Figure 5(a). OAGRNN initially takes around 0.5 seconds for learning to reach the required speed. In the second scenario, the load was increased from 0.2 to 1.67 N.m and then decreased back to 0.2 N.m, the results as shown in Figure 5(b) demonstrate that OAGRNN was capable of forcing the response to the desired speed when the torque was increased and decreased while the PI controller could not. In the third scenario, the torque was increased from 0.2 to 0.7 N.m and then to 1.67 N.m, the results as shown in Figure 6 demonstrate that the OAGRNN was able to recover the desired for all the torque increments while PI was not.



(a) (b)

Figure 5. Response of OAGRNN compared to PI response when the load torque (a) increased from 0.2 to 1.67 N.m and (b) when the load torque increased from 0.2 to 1.67 N.m and then decreased to 0.2 N.m

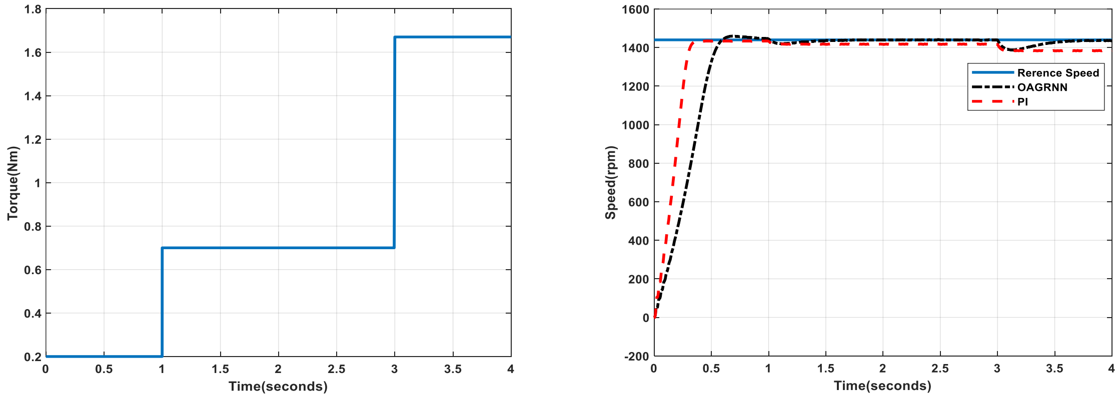


Figure 6. OAGRNN vs PI response for load torque increased from 0.2 to 0.7 N.m and then 1.67 N.m

4.2. Experimental results

To test and validate the proposed algorithm OAGRNN, a three-phase induction motor, together with the power electronics laboratory, is set up. For this purpose, an Arduino Mega 2560 control board is employed, and MATLAB Simulink is used to build the control software. Figure 7 displays the experimental setup. The 3-phase, 4-pole, delta-connection induction motor used in this work has an output power of 300 W and is supplied by 220 V at a rated frequency of 50 Hz. At various load torques, the motor speed is controlled at 2000 rpm.

Figure 8(a) shows the open-loop response of the induction motor speed for a torque change from 0.2 to 1.67 N.m. at 55 sec, demonstrating that the open-loop controller is unable to fix the desired speed when a breaking torque is applied. Figure 8(b) response for the OAGRNN controller for the same load torque shows how the controller can turn the motor back to its desired speed after applying the new torque. In Figure 9, another breaking torque of 2.7 N.m is applied at 55 seconds. The results depicted that the OAGRNN controller could return the motor to its desired speed after the breaking torque was applied, as shown in Figure 9(a). While the open-loop control could not, as shown in Figure 9 (b). In Figure 10, the torque is reduced from 1.67 to 0.2 N.m. Again, the results showed that after applying a light torque, the OAGRNN controller could get the motor speed back to the desired speed, as shown in Figure 10 (a). Whereas the open-loop controller could not, as shown in Figure 10 (b).

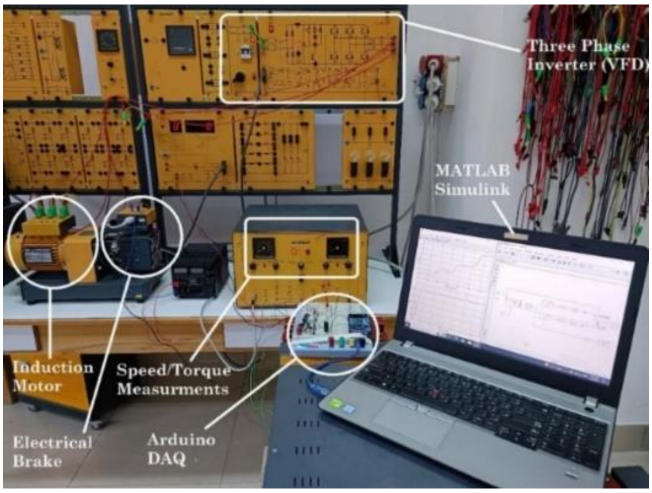
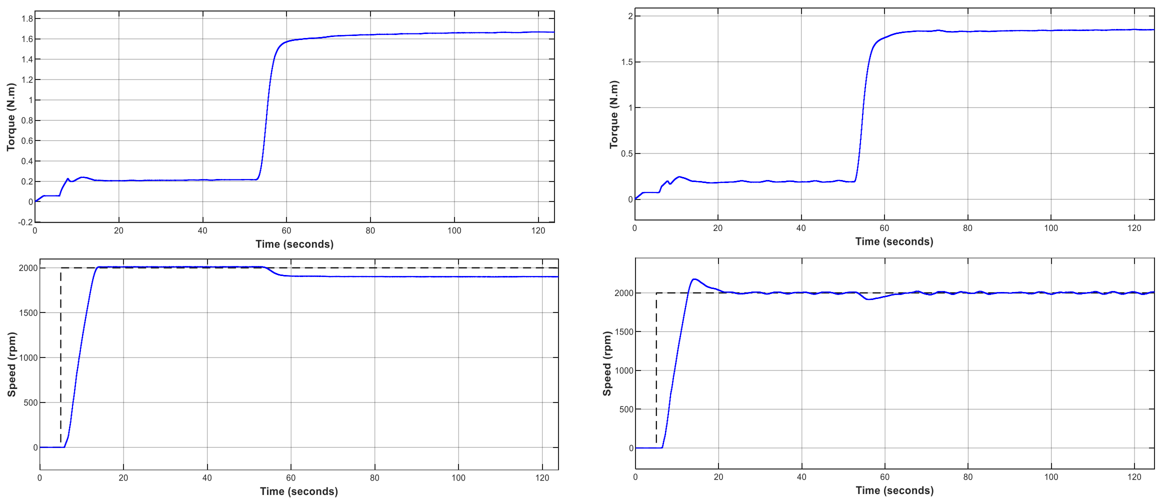
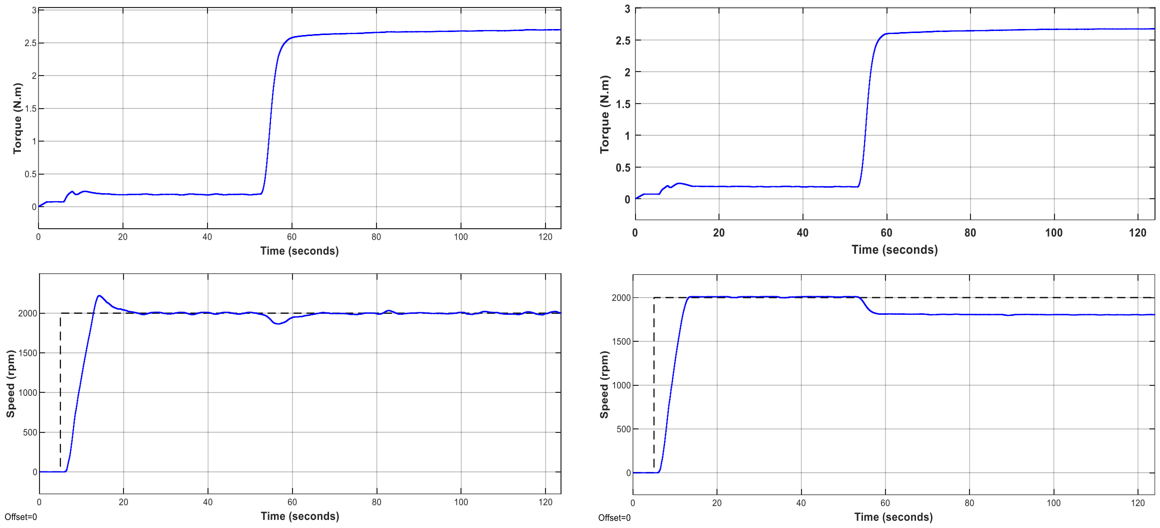


Figure 7. Experimental setup for speed control of induction motor using OAGRNN



(a) (b)

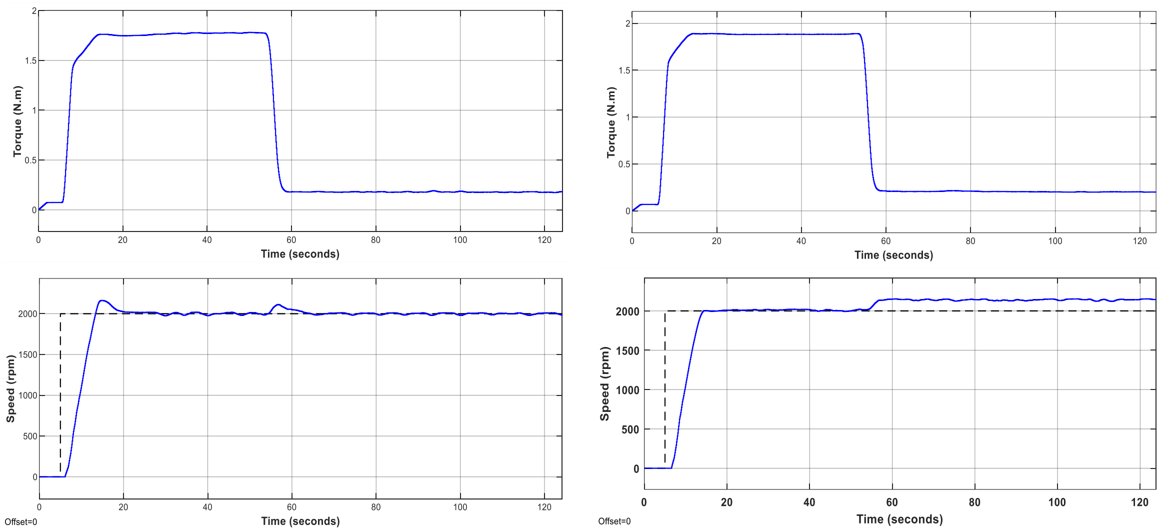
Figure 8. Response for speed with a torque change from 0.2 to 1.67 N.m (a) Open-loop response and (b) OAGRNN controller response



(a) (b)

Figure 9. Response for speed with a torque change from 0.2 to 2.7 N.m (a) OAGRNN controller and (b) Open-loop response for speed

In Figure 11, the three-phase induction motor is subjected to a variable torque application. The results revealed that the OAGRNN controller can maintain the desired speed while applying a variable torque. Also, the PI controller is developed and used in the experimental setup given in Figure 7 for comparison reasons. Figure 12 displays the PI response after applying a braking torque to the induction motor (ranging from 0.2 to 1.7 N.m). The PI controller takes a very long time to reach the speed needed with ripples in comparison to the OAGRNN's response in Figure 9(b). For the experiment time range of 0 to 120 seconds, a variable load torque is provided to the three-phase induction motor, and the PI response is shown in Figure 13 for more comparison. Experimental results in Figure 11 and Figure 13 revealed that under these varied load situations, OAGRNN can follow the intended speed perfectly compared to the PI controller.



(a) (b)

Figure 10. Response for speed with a torque change from 1.67 to 0.2 N.m (a) OAGRNN controller response and (b) open-loop response

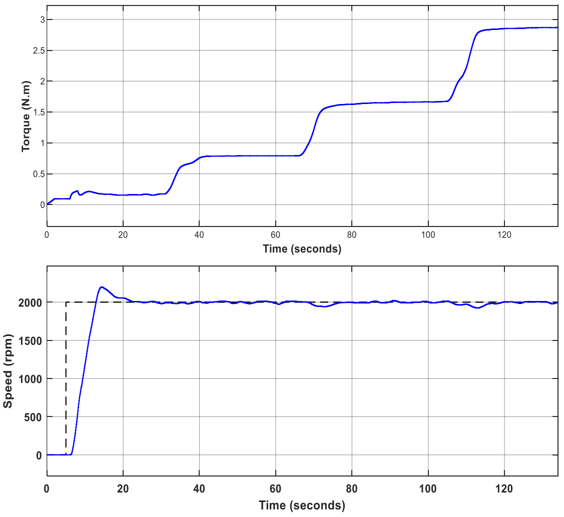


Figure 11. OAGRNN tracking for the desired speed under variable torque

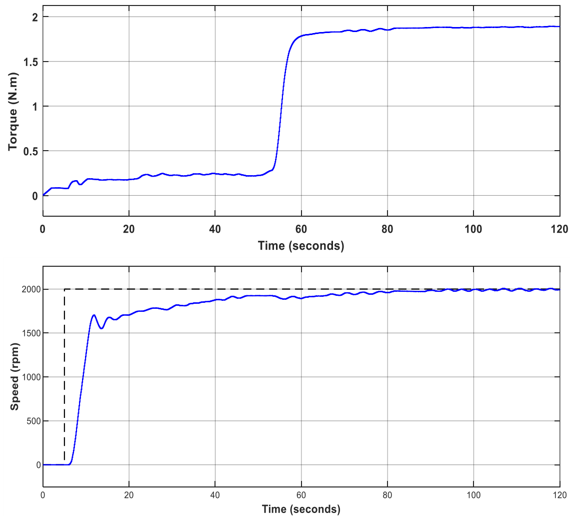


Figure 12. Speed response for PI controller with a torque change from 0.2 to 2.7 N.m

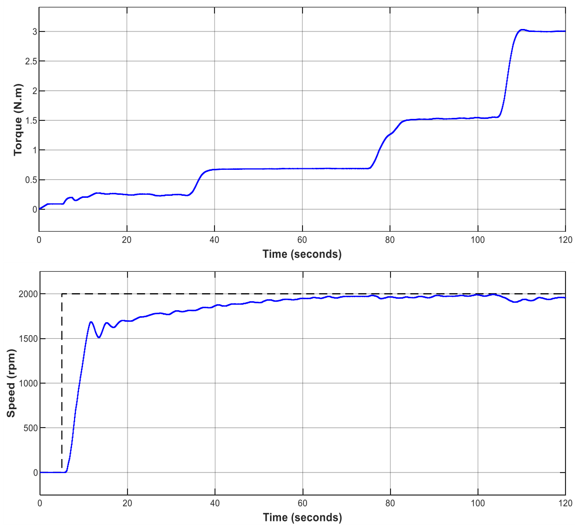


Figure 13. PI tracking for the desired speed under variable torque

5. CONCLUSION

In this work, a three-phase induction motor is utilized in the simulation and experimental models to test the proposed OAGRNN algorithm for optimal speed control. The speed of a three-phase induction motor is directly online controlled using a code that was created using OAGRNN. The OAGRNN controller provides the inverter with the control signal it needs to produce the proper frequency and voltage for the induction motor instantly. A PI controller has been developed as well for comparison with the OAGRNN controller. The proposed OAGRNN controller is tested using simulated and actual case studies of load torques delivered to the induction motor. Based on the simulation and experimental results, the OAGRNN controller allowed for the smooth and reliable control of induction motor speed, and it performed much better than the PI controller at tracking the intended motor speed.

ACKNOWLEDGEMENTS

The authors express their gratitude to Philadelphia University and The University of Jordan for their support.

REFERENCES

[1] A. Z. A. Firdaus, "Design and simulation of fuzzy logic controller for variable frequency drive of three phase AC induction motor," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 12-Special Issue, pp. 834–842, Dec. 2019, doi: 10.5373/jardcs/v11sp12/20193283.

[2] K. Y. Parikh and M. Jain, "The principle of PLC-VFD & its usage in automation industry," *International Journal of Recent Trends in Engineering and Research*, vol. 2, no. 11, 2016.

[3] P. M. Menghal and A. J. Laxmi, "Real time control of induction motor using neural network," in *2018 International Conference on Communication information and Computing Technology (ICCICT)*, Feb. 2018, pp. 1–6, doi: 10.1109/iccict.2018.8325873.

[4] T. A. Hussein, "V/F control of three-phase induction motor driven by VSI based on SVPWM," in *2021 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*, Oct. 2021, pp. 49–53, doi: 10.1109/icasea53739.2021.9733068.

[5] S. E. Farhi, D. Sakri, and N. Golea, "High-performance induction motor drive based on adaptive super-twisting sliding mode control approach," *Archives of Electrical Engineering*, vol. 71, no. 1, pp. 245–263, 2022, doi: 10.24425/aee.2022.140208.

[6] N. Yüksel, H. R. Börklü, H. K. Sezer, and O. E. Canyurt, "Review of artificial intelligence applications in engineering design perspective," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105697, 2023, doi: 10.1016/j.engappai.2022.105697.

[7] S. Bianchini, M. Müller, and P. Pelletier, "Artificial intelligence in science: An emerging general method of invention," *Research Policy*, vol. 51, no. 10, p. 104604, Dec. 2022, doi: 10.1016/j.respol.2022.104604.

[8] M. Alkayyali and J. Ghaeb, "Hybrid PSO–ANN algorithm to control TCR for voltage balancing," *IET Generation, Transmission & Distribution*, vol. 14, no. 5, pp. 863–872, Jan. 2020, doi: 10.1049/iet-gtd.2019.1246.

[9] T. Hagendorff, "A virtue-based framework to support putting AI ethics into practice," *Philosophy & Technology*, vol. 35, no. 3, p. 55, Sep. 2022, doi: 10.1007/s13347-022-00553-z.

[10] D. M. Ragab, J. A. Ghaeb, and I. Al-Naimi, "Enhancing the response of thyristor-controlled reactor using neural network," *International Transactions on Electrical Energy Systems*, vol. 29, no. 12, Jul. 2019, doi: 10.1002/2050-7038.12137.

[11] S. Mahfoud, A. Derouich, A. Iqbal, and N. El Ouanjli, "ANT-colony optimization-direct torque control for a doubly fed induction motor : An experimental validation," *Energy Reports*, vol. 8, pp. 81–98, Nov. 2022, doi: 10.1016/j.egyr.2021.11.239.

[12] A. Rubaai, R. Kotaru, and M. D. Kankam, "Online training of parallel neural network estimators for control of induction motors," *IEEE Transactions on Industry Applications*, vol. 37, no. 5, pp. 1512–1521, 2001, doi: 10.1109/28.952529.

[13] B. Karanayil, M. F. Rahman, and C. Grantham, "Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 167–176, Feb. 2007, doi: 10.1109/tie.2006.888778.

[14] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicevic, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output *LC* filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019, doi: 10.1109/access.2019.2938220.

[15] N. Pimkumwong and M.-S. Wang, "Online speed estimation using artificial neural network for speed sensorless direct torque control of induction motor based on constant V/F control technique," *Energies*, vol. 11, no. 8, p. 2176, Aug. 2018, doi: 10.3390/en11082176.

[16] S. Mahfoud, A. Derouich, N. El Ouanjli, N. V. Quynh, and M. A. Mossa, "A new hybrid ant colony optimization-based PID of the direct torque control for a doubly fed induction motor," *World Electric Vehicle Journal*, vol. 13, no. 5, p. 78, Apr. 2022, doi: 10.3390/wevj13050078.

[17] S. Mahfoud, A. Derouich, N. EL Ouanjli, M. EL Mahfoud, and M. Taoussi, "A new strategy-based PID controller optimized by genetic algorithm for DTC of the doubly fed induction motor," *Systems*, vol. 9, no. 2, p. 37, 2021, doi: 10.3390/systems9020037.

[18] A. Alias, "Overview of brushless DC motor: Construction and application," *International Journal for Technological Research in Engineering*, vol. 7, no. 8, pp. 6669–6675, 2020.

[19] M. Mahmud, S. M. A. Motakabber, A. H. M. Zahirul Alam, and A. N. Nordin, "Adaptive PID controller using for speed control of the BLDC motor," in *2020 IEEE International Conference on Semiconductor Electronics (ICSE)*, Jul. 2020, pp. 168–171, doi: 10.1109/icse49846.2020.9166883.

[20] K. Premkumar, T. Thamizhselvan, M. V. Priya, S. B. R. Carter, and L. P. Sivakumar, "Fuzzy anti-windup PID controlled induction motor," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 184–189, Oct. 2019, doi: 10.35940/ijeat.a1113.109119.

[21] G. Boukhalfa, S. Belkacem, A. Chikhi, and S. Benaggoune, "Genetic algorithm and particle swarm optimization tuned fuzzy PID controller on direct torque control of dual star induction motor," *Journal of Central South University*, vol. 26, no. 7, pp. 1886–1896, Jul. 2019, doi: 10.1007/s11771-019-4142-3.

[22] K. S. Yap, C. P. Lim, and I. Z. Abidi, "A hybrid ART-GRNN online learning neural network with a ε-insensitive loss function," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1641–1646, Sep. 2008, doi: 10.1109/tnn.2008.2000992.

[23] F. Allythi, *Three-phase asynchronous motors*. Noor Publishing, 2020.

[24] P. V Rao, *Power semiconductor drives*. BS Publications, 2017.

[25] L. Niu and L. Ye, "Adaptive tracking control of nonlinear systems using neural networks," in *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, Feb. 2009, pp. 12–15, doi: 10.1109/car.2009.15.

[26] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, Jul. 2020, doi: 10.1007/s40435-020-00665-4.

BIOGRAPHIES OF AUTHORS

Ahmad J. Al-Mahasneh received the B.Sc. and M.Sc. degrees in mechatronics/mechanical engineering from the Jordan University of Science and Technology in 2010 and 2014, respectively, and the Ph.D. degree in mechanical engineering from the University of New South Wales Canberra in 2020. He is an assistant professor at the Department of Mechatronics Engineering at Philadelphia University, Jordan. His research interests include data-driven modeling and control, adaptive and intelligent control, evolving systems and neural networks, nonlinear control, and unmanned aerial vehicle control. He is a member of the IEEE control systems society (CSS), IEEE robotics and automation society (RAS) and IEEE systems, man, and cybernetics society (SMC), he can be reached via email: amahasneh@philadelphia.edu.jo.

Samer Z. Salah holds an M.Sc. degree in electrical power and control engineering from Tafila Technical University, Jordan, conferred in 2021, and a B.Sc. degree in mechatronics engineering from the Hashemite University Jordan, completed in 2008. He has accumulated extensive experience, having worked at Philadelphia University for 14 years as a laboratory supervisor. During his tenure, he was responsible for teaching various mechatronics labs including control, automation, electrical machines, power electronics, and drives. Currently, he serves as a lecturer at the Department of Mechatronics Engineering within the School of Engineering at The University of Jordan, Amman, Jordan. His research interests encompass smart grids, renewable energy systems, power system stability, artificial intelligence, and machine learning. For further inquiries, he can be reached via email: samer.salah@ju.edu.jo.

Jasim A. Ghaeb is a professor of power electronics and control, and he is currently with the Department of Alternative Energy Technology, Faculty of Engineering and Technology, Philadelphia University, Jordan. He received a Ph.D. degree from the University of Bradford, the U.K, in 1989 in electrical power systems-control and power electronics. He is with 30 years of experience in teaching, research, and administration. Research areas include reactive power control, artificial intelligence, heuristic algorithms-based power system control, and embedded systems and data integrity. He can be contacted at email: jghaeb@philadelphia.edu.jo.

Mohammed Baniyounis was born in Jordan. He obtained his Ph.D. from the University of Kaiserslautern, Kaiserslautern, Germany in 2006. He has strong knowledge and experience of re-engineering and reverse engineering delegated through specifications and verification of software/hardware systems using different tools and logic. Besides his expertise in the distributed and concurrent systems approach for software modeling, design and development. He is interested in applying these methods introduced above along with software engineering technologies to solve automation and control-specific problems. A basic understanding of electrical machines also assists with this knowledge. Currently, he is the acting Dean of Engineering and Technology Faculty at the Philadelphia University in Amman. He can be contacted at email: mbaniyounis@philadelphia.edu.jo.