

# Optimization of an ANN-based speed and position estimator for an FOC-controlled PMSM using genetic algorithm

Juan Paolo Quismundo<sup>1</sup>, Edwin Sybingco<sup>1</sup>, Maria Antonette Roque<sup>1</sup>, Alvin Chua<sup>2</sup>, Leonard Ambata<sup>1</sup>

<sup>1</sup>Department of Electronics and Computer Engineering, Gokongwei College of Engineering, De La Salle University, Manila, Philippines

<sup>2</sup>Mechanical Engineering Department, Gokongwei College of Engineering, De La Salle University, Manila, Philippines

---

## Article Info

### Article history:

Received Sep 10, 2022

Revised Jan 26, 2023

Accepted Feb 16, 2023

---

### Keywords:

Artificial neural networks

Genetic algorithm

PMSM

Sensorless field oriented control

---

## ABSTRACT

To further improve the performance of sensorless permanent magnet synchronous motor (PMSM) implementations, this study develops a neural network-based estimator for the speed and position estimation of a PMSM using field oriented control (FOC) as its control scheme. The proposed neural network's hyperparameters are optimized using genetic algorithm. The neural network is trained and optimized based on a training dataset obtained from the Simulink simulation of the motor control system. The hyperparameters optimized include the training algorithm parameters, batch size, and the number of hidden layers and the corresponding neurons. The proposed estimator performed with better estimation accuracy than conventional estimators such as the sliding mode observer (SMO), model reference adaptive system (MRAS), and two other neural network configurations. The qualifications were made on steady-state and dynamic conditions. In terms of efficiency, the proposed estimator has a relatively lower power consumption but still falls short of the power drawn when using an actual sensor. The qualification process verified that the optimization of the neural network's hyperparameters using genetic algorithm can provide a better performance in the estimation of motor parameters in sensorless motor applications.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

## Corresponding Author:

Juan Paolo Quismundo

Department of Electronics and Computer Engineering, Gokongwei College of Engineering

De La Salle University

2401 Taft Avenue, Malate, Manila, 1004, Philippines

Email: jpbquismundo@gmail.com

---

## 1. INTRODUCTION

In recent years, permanent magnet synchronous motors (PMSMs) have managed to gain an increased amount of research interest given its efficiency, flexibility in terms of speed, and cost-effectiveness [1]. Control methods for the PMSM execute efficiency and robustness but these methods require information such as the rotor's speed and/or position which are accurately retrieved using mechanical sensors in the shaft of the ac machine. But with the introduction of sensorless methods, there presents the advantage of reducing cost and increasing the motor control system's ruggedness and reliability [2]. In proposed models for PMSM speed and/or position estimation, there are model-based [3] and rotor-saliency-based techniques [2], [4].

Since model-based sensorless drives are very dependent on the motor parameters, variations or tolerances from the nominal value can affect the performance of the motor based on the decreased accuracy of the estimator which is an undesirable effect [5]. Certain methods have also been developed to address variations in the motor parameters including the use of model-reference adaptive models (MRAS) [2], [6], [7]. These MRAS-based solutions for parameter deviation are incorporated in the sensorless estimation

method being used. However, when the speed and position estimator is modeled based on artificial neural networks, the compensation for the deviation in motor parameters are embedded in the estimation model itself as demonstrated in [8]-[11].

The use of neural networks in position and/or speed estimation is another popular method used along with MRAS, sliding mode observer (SMO), Luenberger observers, among others [12]-[19]. Different network architectures have been implemented in rotor speed and/or position estimators such as feedforward [20]-[30], recurrent [20], [29], linear [29], and radial basis function [28] demonstrating advantages such as high estimation accuracy and good dynamic performance [29]. In [11] proposed a neural network-based speed and position estimator that replaces the sliding-mode observer in conventional applications, providing accurate estimations without dependence on the motor parameters. However, most of the research on neural network-based estimators in field oriented control (FOC) have relied upon a trial-and-error method in determining the best hyperparameters including the number of neurons and layers in their proposed neural network architectures, while others have failed to identify the methods used in determining these hyperparameters. Genetic algorithm in hyperparameter optimization in other applications have resulted to improvements in the neural-network's adaptive ability and generalization [31]. In this study, genetic algorithm is proposed to be used to optimize the hyperparameters of an artificial neural network that serves as a PMSM rotor's speed and position estimator.

## 2. MODELING THE ESTIMATOR

### 2.1. PMSM model

The PMSM is used in this study as the induction motor as it is known for its efficiency and higher torque density in terms of torque per unit volume [2]. The rotor consists of surface-mounted permanent magnets and its stator has a Y-connected winding for every 120°. The 3-phase PMSM is regarded as a nonlinear time-varying system that can be simplified into a two-phase voltage model in synchronous rotating  $d - q$  coordinates via the concepts of Clark and Park transformations. The model of the PMSM is shown in (1).

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R_s + \frac{d}{dt}L_q & -\omega_e L_q \\ \omega_e L_q & R_s + \frac{d}{dt}L_d \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e \lambda_f \end{bmatrix} \quad (1)$$

Where  $u_d, u_q$  represents the  $dq$ -axis stator voltages of the PMSM;  $i_d, i_q$ , representing the  $dq$ -axis stator currents;  $L_d, L_q$ , representing the  $dq$ -axis stator inductances;  $R_s$ , represents the stator winding resistance of the PMSM;  $\omega_e$ , represents the rotor's angular speed; and  $\lambda_f$ , represents the permanent magnets' flux-linkage. As mentioned, the  $d - q$  coordinate model can be transformed into an equivalent model in the stationary  $\alpha - \beta$  coordinate system. The inverse Park transforms on the  $d - q$  model is shown in (2).

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \begin{bmatrix} R_s + \frac{d}{dt}L_s & 0 \\ 0 & R_s + \frac{d}{dt}L_s \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} \quad (2)$$

Where  $u_\alpha, u_\beta$  represents the alpha and beta stator voltages;  $i_\alpha, i_\beta$  represents the alpha beta stator currents;  $R_s$  represents the stator winding resistance of the PMSM;  $L_s$  represents the stator winding inductance; and  $e_\alpha, e_\beta$  represents the alpha beta equivalent back electromotive force (back-EMF) voltages.

### 2.2. Sensorless field oriented control scheme

FOC or the vector control of a PMSM is a motor drive control scheme or approach that improved the performance of an induction motor driven by an inverter. This approach can obtain near-instantaneous response in torque by transferring from a steady-state condition to another. With the motor's phase currents serving as the inputs, the first transformation is the alpha-beta transformation or also called as the Clarke transformation wherein the three-phase system is projected into a two-dimension orthogonal system. The mathematics behind the Clarke transform isolates the common-mode component among the three vectors as shown in (3).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (3)$$

Where  $i_a, i_b, i_c$  represents the three-phase motor currents;  $i_\alpha, i_\beta$  represents the alpha and beta axis projections of the motor currents. The Park transform will deliver the 2-D orthogonal system into a rotating 2-D

orthogonal system. This will deliver almost direct current (DC) values to be used as feedback values for the control loop. This transform enables the ease of control by manipulating these DC values instead of sinusoidal ones. The formula used for Park Transform is given by (4).

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \sin \theta & -\cos \theta \\ \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{4}$$

Where  $i_d, i_q$  represents the direct and quadrature projections of the motor currents;  $i_\alpha, i_\beta$  represents the alpha and beta axis projections of the motor currents. Proportional-integral controllers are used in the proposed control system to reduce certain error values to zero. Reference values for flux, speed, and torque are introduced. Reference flux should be zero since this follows the concept of FOC. The reference speed is user input. The reference torque value is simply obtained from the output of the proportional-integral (PI) controller for speed. Space vector modulation (SVM) is the control method to be used in FOC in driving the 3-phase PMSM in this proposal. The SVM is utilized due to its ability to make instantaneous changes in its output phasor’s angular position which is advantageous when used for field-oriented control or vector control of a 3-phase motor. The block diagram for the entire field-oriented control scheme is shown in Figure 1. This includes the PMSM, the Clarke and Park transforms, the PI controllers, the observer, the space vector modulator, and the 3-phase inverter. This will be the same block diagram used in the proposed genetic algorithm (GA)-optimized neural network estimator which is to be represented by the observer block.

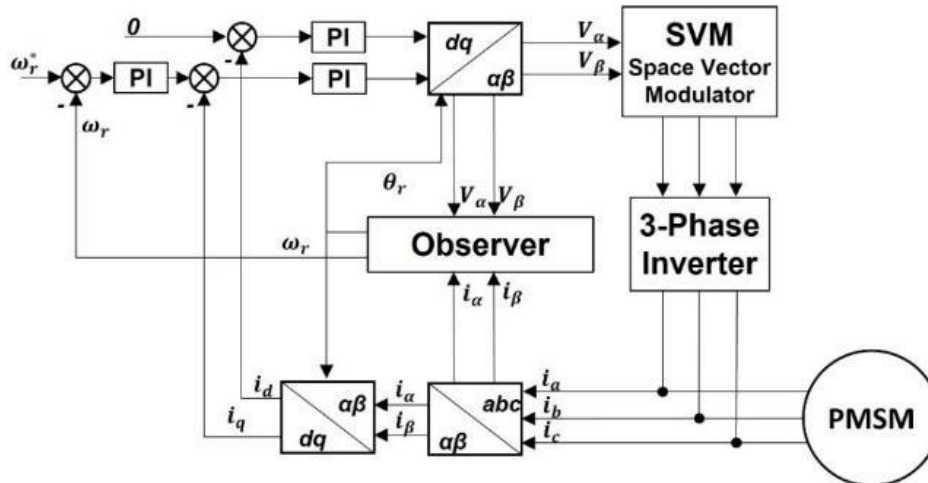


Figure 1. Sensorless field oriented control block diagram

### 2.3. Neural network estimator model

In this study, the estimator block used takes the alpha and beta voltages and currents as the inputs to the neural network. The outputs come in the form of  $\sin(\theta_{NN})$  and  $\cos(\theta_{NN})$  that is a representation of the motor’s equivalent alpha and beta back-EMF as shown in (5).

$$\begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = \lambda_f \omega_e \begin{bmatrix} -\sin \theta_e \\ \cos \theta_e \end{bmatrix} \tag{5}$$

where  $\lambda_f$  is the flux linkage of the permanent magnets;  $\omega_e$  is the rotor’s angular speed;  $e_\alpha$  and  $e_\beta$  represents the back-EMF voltages; and  $\theta_e$  is the electrical position of the motor.

$$\hat{\theta} = -\text{atan}^{-1}\left(\frac{\hat{e}_\alpha}{\hat{e}_\beta}\right) \tag{6}$$

where  $e_\alpha$  and  $e_\beta$  represents the back-EMF voltages obtained from (5), and  $\hat{\theta}$  is the estimated angle. To retrieve the estimated speed, the change in position is obtained using (7).

$$\theta_{err} = \hat{\theta} - \tilde{\theta} \approx \sin(\tilde{\theta} - \hat{\theta}) = \sin(\tilde{\theta}) \cdot \cos(\hat{\theta}) - \cos(\tilde{\theta}) \cdot \sin(\hat{\theta}) \tag{7}$$

where  $\hat{\theta}$  is the position value determined from equation 6,  $\tilde{\theta}$  is the position obtained from integrating the rotor speed in the phase-locked loop (PLL), and  $\theta_{err}$  is the position error that serves as the input of the phase-locked loop as shown in Figure 2.

In Figure 2, the position error  $\theta_{err}$  is fed into a PI controller, the output of the PI controller is the estimated rotor speed  $\hat{\omega}$ . Integrating further, the estimated position  $\tilde{\theta}$  is obtained. The scheme, when using sine and cosine values highly reduces spikes from  $-\pi$  to  $\pi$  [32]. The proposed neural network estimator for optimization is shown in Figure 3.

In Li and Zhao [33], the PLL used the calculated alpha and beta back-EMF to estimate the speed and position. In this case, the SMO calculates the back-EMF which serves as the PLL input. The use of the PLL was proven to be effective in reducing estimation error and filtering high frequency components. By replacing the back-EMF with an artificial neural network (ANN), the proposed model will make use of the advantages of the PLL but without the disadvantages brought by conventional estimators like the inherent chattering when using the SMO. This performance will be validated by comparing the speed and position estimation errors.

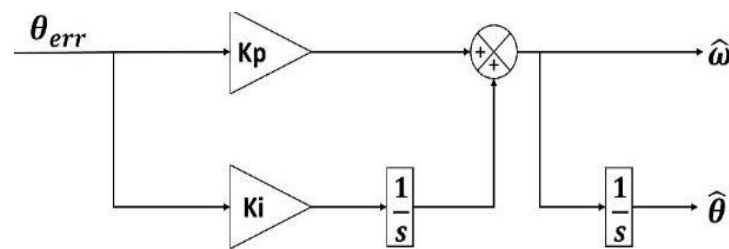


Figure 2. Phase locked loop block diagram

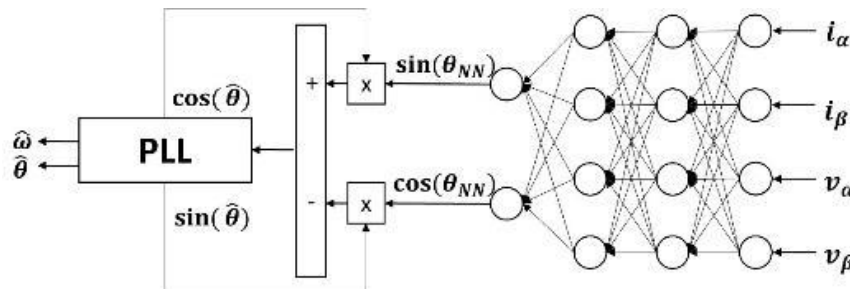


Figure 3. Neural network estimator block diagram

### 2.4. Genetic algorithm optimization

As stated in this study, previous research involving the use of neural network estimators in field-oriented control of PMSMs make use of trial and error or using the previous experimental data in determining certain hyperparameters. This study will introduce the use of genetic algorithm in optimizing the network hyperparameters for this application. The process for genetic algorithm to be used in this study is shown in Figure 4. The algorithm will start off with initializing a population. Each individual of the population will have its fitness evaluated. From the data of fitness values, the termination criterion is checked if already met. If not, the population enters parent selection wherein individuals are paired based on their individual fitness to generate an offspring. This offspring is developed by the crossover operator. The mutation operator is also used to help avoid convergence on a local minimum. This will ensure the effectiveness of the optimization process. In developing the genetic algorithm for the speed and position estimator, the termination criteria should be set. These criteria can come in the form of a maximum number of generations and/or in the form of a target mean-squared error (MSE). As for the hyperparameters to be optimized, a valid range of values should be declared from which the genetic algorithm will limit itself to. This range can be realized during the linear mapping process. The initial population is used as the initial values for the hyperparameters in the neural network design. The target hyperparameters to be optimized are the following: number of hidden layers, number of neurons for the hidden layers, activation function, batch size, initial mu, mu decrease factor, and the mu increase factor.

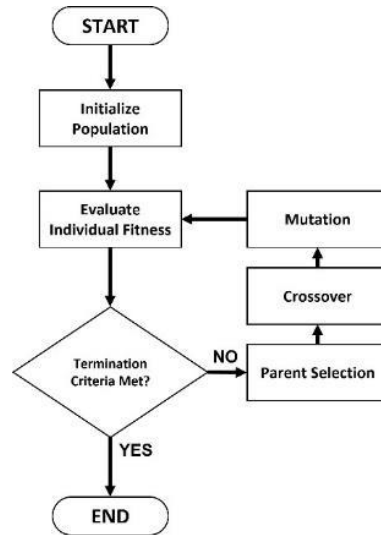


Figure 4. Genetic algorithm workflow

**2.5. GA-optimized neural network experimental setup**

The field-oriented control is modeled in Simulink as well as the neural network estimator. The dataset containing the alpha and beta voltages and currents as well as the target sine and cosine values are obtained from the simulation and is exported. The genetic algorithm optimization is done externally via python to obtain the set of hyperparameters with the best estimation accuracy. The optimized neural network model is imported back into the Simulink model for qualification. The motor used in this study is the X2212 980 KV II whose parameters are shown in Table 1.

The qualification of the proposed GA-optimized neural network estimator is conducted against the conventional methods of using the model-reference adaptive system (MRAS) and the sliding-mode observer (SMO) as well as two other neural network configurations from [11], [17]. The performances of the estimators are also compared against when using sensors which is the reference performance. The Simulink model of the sensed FOC system is shown in Figure 5.

Table 1. Motor parameters

Parameter	Value
Rated power	300 W
Rated speed	9800 rpm
Rated torque	0.125 Nm
Pole pairs	7
Stator-winding resistance	0.39 Ω
Stator winding inductance	12.1 mH
Flux linkage	0.1 Wb
Inertia	0.001 kg m <sup>2</sup>

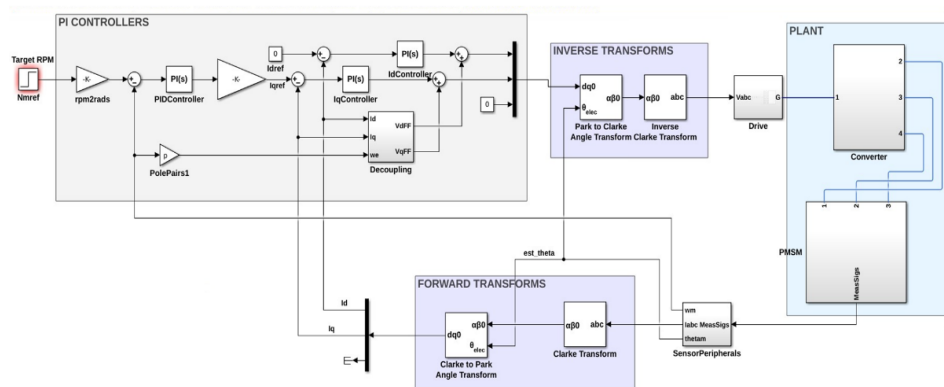


Figure 5. Simulink model of the FOC system

### 3. EXPERIMENTAL RESULTS

The performance of the proposed GA-optimized neural network estimator is verified using several comparative test cases against the sensor-based FOC, the conventional MRAS and SMO estimators, and two reference neural network configurations from [11], [17]. A sample of the training dataset is retrieved first. Then the genetic algorithm data is also to be presented containing the various iterations made on the hyperparameters. Then the optimized neural network is compared against the other methods in steady-state condition to validate the estimation accuracy improvements made. Followed by the dynamic response, the input power drawn when using the different estimators is measure as well. For the purpose of simplicity in the data that follows, only the electrical angle of the motor will be presented. The actual rotor position measured as the reference is obtained as the mechanical angle and is simply converted into the electrical angle for comparison since the estimator's output is in electrical degrees. The conversion is done using the following equation:

$$\theta_{elec} = \theta_{mech} \left( \frac{P}{2} \right) \quad (8)$$

where  $\theta_{elec}$  is the electrical angle,  $\theta_{mech}$  is the mechanical angle, and  $P$  is the number of poles.

#### 3.1. Training dataset

From the working model of the FOC system, the dataset to be used for the training of the neural network estimator was obtained. Due to the periodic behaviour of the motor, the dataset for a certain speed and load condition only covers one electric period. The dataset to be used are gathered at specific operating points of the motor. Conditions wherein the motor is not to be operated should not be included. The operating points in this study are from 1000 to 9000 rpm as the speed reference, and from 0.0 to 1.0 Nm as the motor load. A sample dataset for the 4000 rpm and 0.8 Nm Load operating point's voltages, currents, and sine-cosine components are shown in Figure 6(a), Figure 6(b), and Figure 6(c) respectively. It can be observed that all the parameters and targets take the form of sinusoidal waveforms.

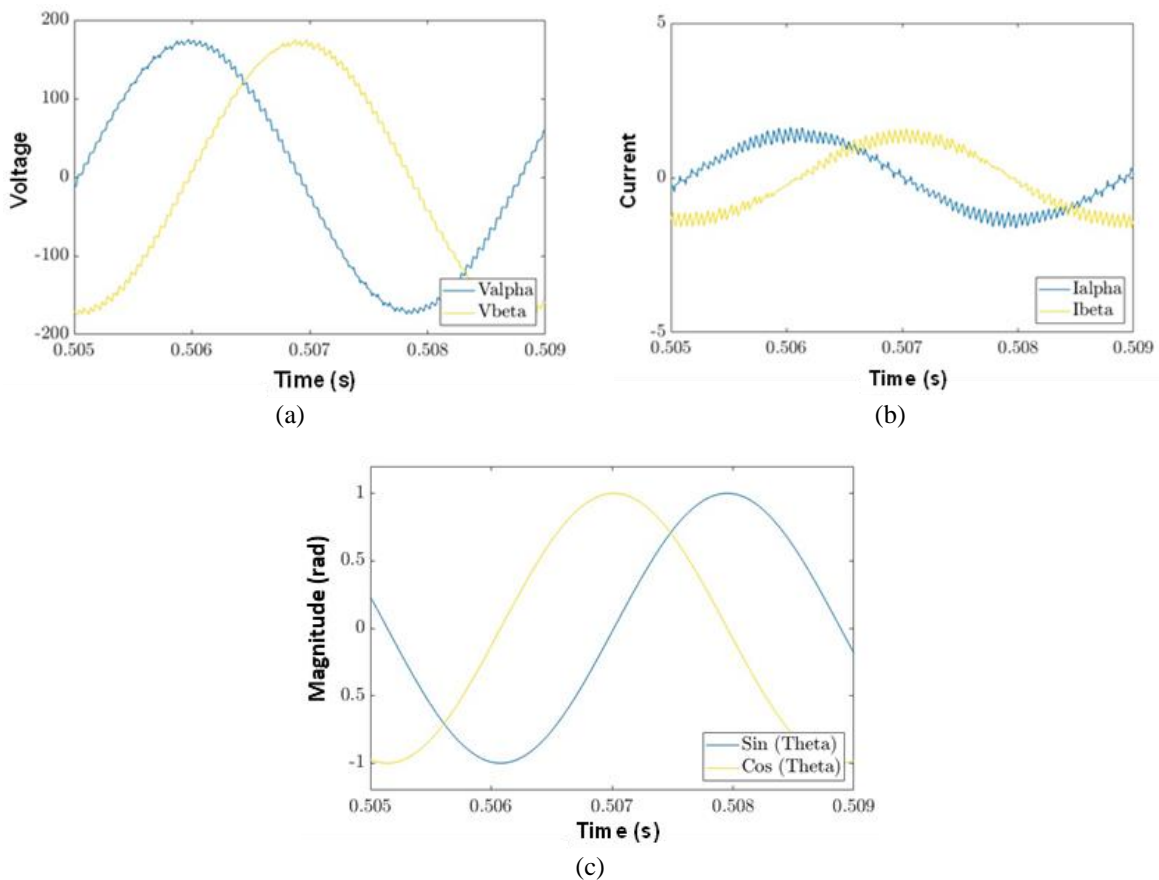


Figure 6. Dataset sample at 4000 rpm: (a)  $\alpha\beta$  voltages, (b)  $\alpha\beta$  currents, and (c) sine and cosine components

### 3.2. Genetic algorithm optimization

Initially, there is a population of 100 individuals with genes that are randomly selected and that represents the hyperparameters of the neural network. The performance of each individual is determined by training a neural network using the individual's hyperparameters on the collected dataset. The GA was run for 30 generations. The performance of each individual in terms of its neural network's loss over the entire run is shown in Figure 7(a).

In Figure 7(a), majority of the individuals scored a loss of less than 0.3. The individuals that scored a loss of 1 are those that were unable to converge during training resulting in an error and were given the highest loss. The data is presented in three indicators for the tanh, linear, and sigmoid activation functions of the hidden layers for every individual. As better observed in Figure 7(b), majority of the individuals that uses the linear activation function are encountering resistance to perform better than a loss of 0.07. It can also be observed that a lot of individuals retain the sigmoid activation function through generations indicating it is performing relatively better than the two other activation functions, and numerous individuals are performing with a loss of less than 0.001. In considering a minimum loss of 0.0001, the individuals meeting this criterion are shown in Figure 7(c). The loss of the individual is plotted against the total number of neurons that it requires. From the figure it can be shown that the number of neurons does not solely dictate the best performing configuration but still relies on the remaining hyperparameters as there are individuals above the 250-neuron count but is outperformed by individuals in the 100 to 150 neuron count range. Based on the genetic algorithm optimization, the hyperparameters for the individual with the fewest neurons with a loss of less than 0.0001 and the best performing individual are shown in table, along with the configuration used in the reference neural networks [11], [17].



Figure 7. Genetic algorithm optimization results (a) optimization scores, (b) zoomed-in version, and (c) neuron count vs loss

As shown in Table 2, the activation function, batch size and the Levenberg-Marquardt hyperparameters are the same, save for the number of neurons in the hidden layers between the least complex and the best performing configuration. For the remainder of the study, the best performing set of hyperparameters is to be used due to its relatively better performance sans the complexity. The total time for the genetic algorithm optimization process was 14 hours and 38 minutes. The optimization was done without the aid of the graphics processing unit (GPU).

Table 2. Neural network configurations

Hyperparameter	Loss	Activation function	Batch size	Initial mu	Mu decreases factor	Mu increase factor	Hidden layer 1 neurons	Hidden layer 2 neurons	Hidden layer 3 neurons
Least complex	0.00008956	Sigmoid	117	0.1	0.15	50	26	13	6
Best performance	0.00001140	Sigmoid	117	0.1	0.15	50	100	13	9
NN REF 1 [11]	-	Sigmoid	-	-	-	-	4	4	4
NN REF 2 [17]	-	Sigmoid	-	-	-	-	4	5	10

### 3.3. Steady-state performance

Qualifications for the estimators starts off with the speed and position estimation accuracy. The performances of the estimators for accuracy gives an insight into the expected response for the subsequent tests. Estimators with high accuracy would have an FOC performance comparable to when using a sensor-based system. The data for estimation accuracy is taken at steady state at a fixed speed reference of 4000 rpm with the rated load of 0.8 Nm.

The speed estimation is comparable for the estimators as shown in Figure 8, the variations manifest in the peak-to-peak and the average speed estimation error, but all estimators can average down on the actual 4000 rpm motor speed, with Figure 8(a) being the comparison between GA-NN and using a sensor. In comparing the performance of the GA-NN vs MRAS estimators, as shown in Figure 8(b), it is observed that the GA-NN easily outperforms the MRAS given the presence of a ripple in both the speed and position estimates. For the SMO comparison with the GA-NN, the SMO can better estimate the speed of the motor as shown in Figure 8(c). However, the inherent chattering in a SMO estimation is visible especially in the position estimation whereas the GA-NN has a more accurate and stable estimation relatively. The GA-NN position estimation is very comparable to that of the neural network reference 1 (NN REF 1) [11], there are some low frequency signatures on the NN REF 1 as compared to the GA-NN, as shown in Figure 8(d) but the signature of the estimation is still relatively similar. For the NN REF 2 performance however, the low frequency variations are more observable, as shown in Figure 8(e). The differences are also more profound when observing the transition from 2 pi radians to 0 radians in the vertical fall of the position value, the differences in terms of time for the transitions provide additional insights of how the estimates vary.

The steady state estimations of the proposed GA-optimized neural network against reference estimators provide an early insight as to the estimation accuracy performances. The peak-to-peak and the average estimation errors for both speed and position of the motor provides the metric in qualifying the different estimators, as shown in Table 3.

In Table 3, the proposed GA-NN estimator outperforms the other reference estimators in terms of speed estimation, for both *pk-to-pk* and average error. The SMO also has a very good estimation performance for the average position error, but the effect of the chattering is observable in the *pk-to-pk* error. For the speed estimation, the GA-NN has the smallest *pk-to-pk* error but the NN REF 1 [11] outperformed in terms of the average speed estimation error. The data presented gives an overview of the improvements made by optimizing the NN hyperparameters via genetic algorithm in terms of steady-state estimation.

Table 3. Steady-state performance summary

Estimator	Position error ( <i>pk to pk</i> radians)	Average position error (radians)	Speed error ( <i>pk to pk</i> rpm)	Average speed error (rpm)
GA-NN	0.09015	0.02933	10.67	0.1396
SMO	0.2787	0.03658	12.16	0.2373
MRAS	0.129	0.0387	12.65	0.1448
NN REF 1 [11]	0.1147	0.07303	43.1	0.1214
NN REF 2 [17]	0.2404	0.08891	73.51	0.1725



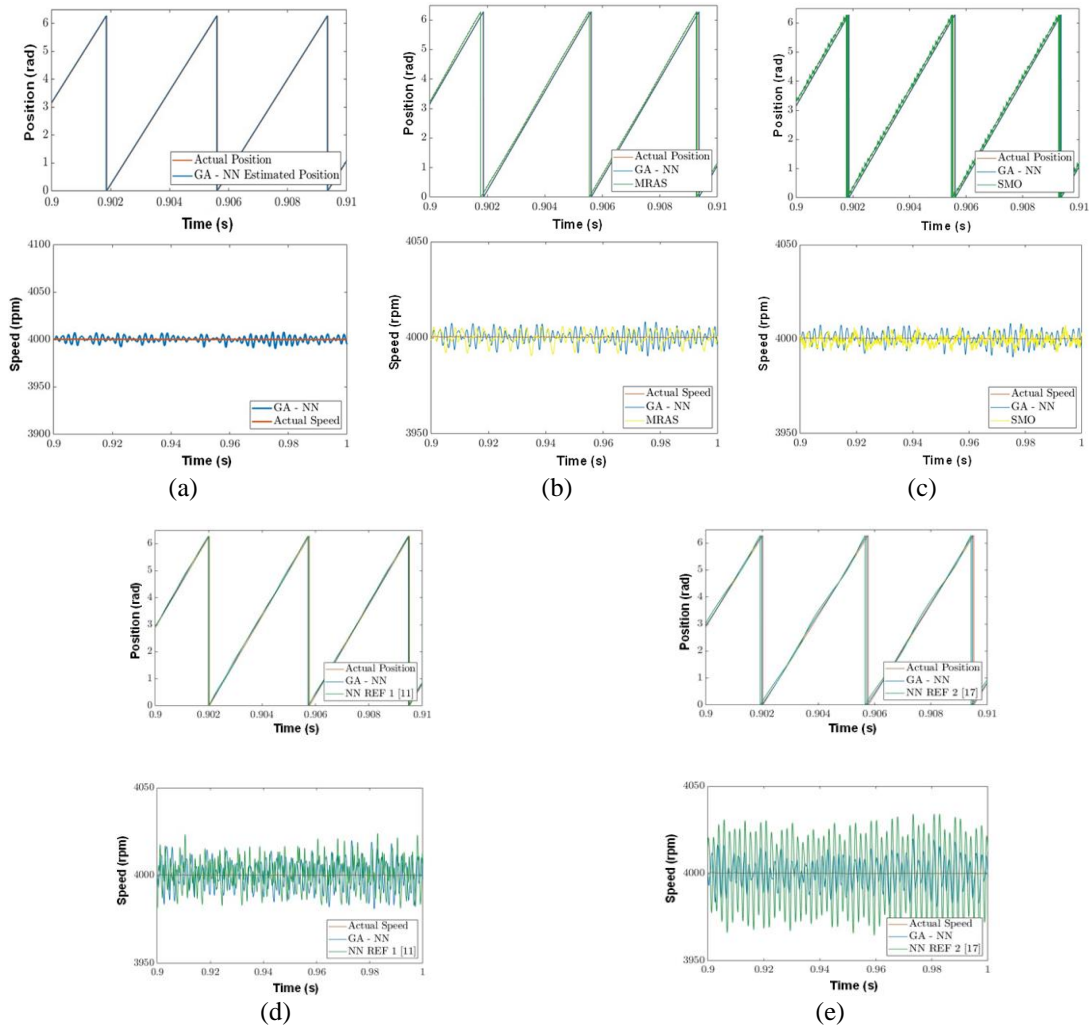


Figure 8. Stead-state position and speed estimation accuracy of GA-NN against (a) sensed and (b) MRAS, (c) SMO, (d) NN REF 1 [11] and (e) NN REF 2 [17]

### 3.4. Step-response performance

The step-response of the FOC using the estimators provides an outlook for the performance using the estimated values under dynamic conditions. The reference or target speed is stepped-up from 0 to 4000 rpm at 0.8 Nm load. The GA-NN estimation, as shown in Figure 9(a) had a leading response compared to a near vertical increase against the performance of the MRAS, as shown in Figure 9(b). There is also an observable overshoot against the GA-NN. An attempt to increase the speed was observed early for the MRAS response but was not able to reach the target speed initially. For the SMO response, as shown in Figure 9(c), it lagged the GA-NN response as well with no other abnormalities observed during the step input. The FOC system was able to reach its target speed for all three estimators used.

The signatures of the motor speed when using the GA-NN and the two reference neural network estimators are very similar as shown in Figure 9(d) and Figure 9(e). The GA-NN response is leading against the two reference NN estimators. The step-response comparisons show the effect of the difference in neural network configuration for a dynamic response. The GA-NN has performed relatively better than the two other neural network configurations.

The summary of the step-response measurements made on the GA-NN and the other reference estimators is shown in Table 4. As expected, the sensed FOC provides the best response to a step-input. The GA-NN however, outperforms all the other estimators used in terms of rise time and settling time. The MRAS has the lowest percent overshoot as compared to the other estimators. In terms of the steady-state error, the GA-NN has the same performance as the SMO both coming after the performance when using the sensor-based FOC. The data from Table 4 gives an overview on the effect of using a GA-optimized neural network estimator compared to the conventional estimators as well as other neural networks that are not optimized by genetic algorithm.

Table 4. Step response performance summary

Estimator	Rise time (ms)	Settling time (ms)	Steady-state error (rpm)	Overshoot (%)
Sensored	8.559	17	6	0.725
GA-NN	12.114	25	8	0.95
SMO	12.646	28	8	0.775
MRAS	12.774	35	9	0.675
NN REF 1 [11]	12.766	28	11	0.99
NN REF 2 [17]	13.085	60	10	1.15

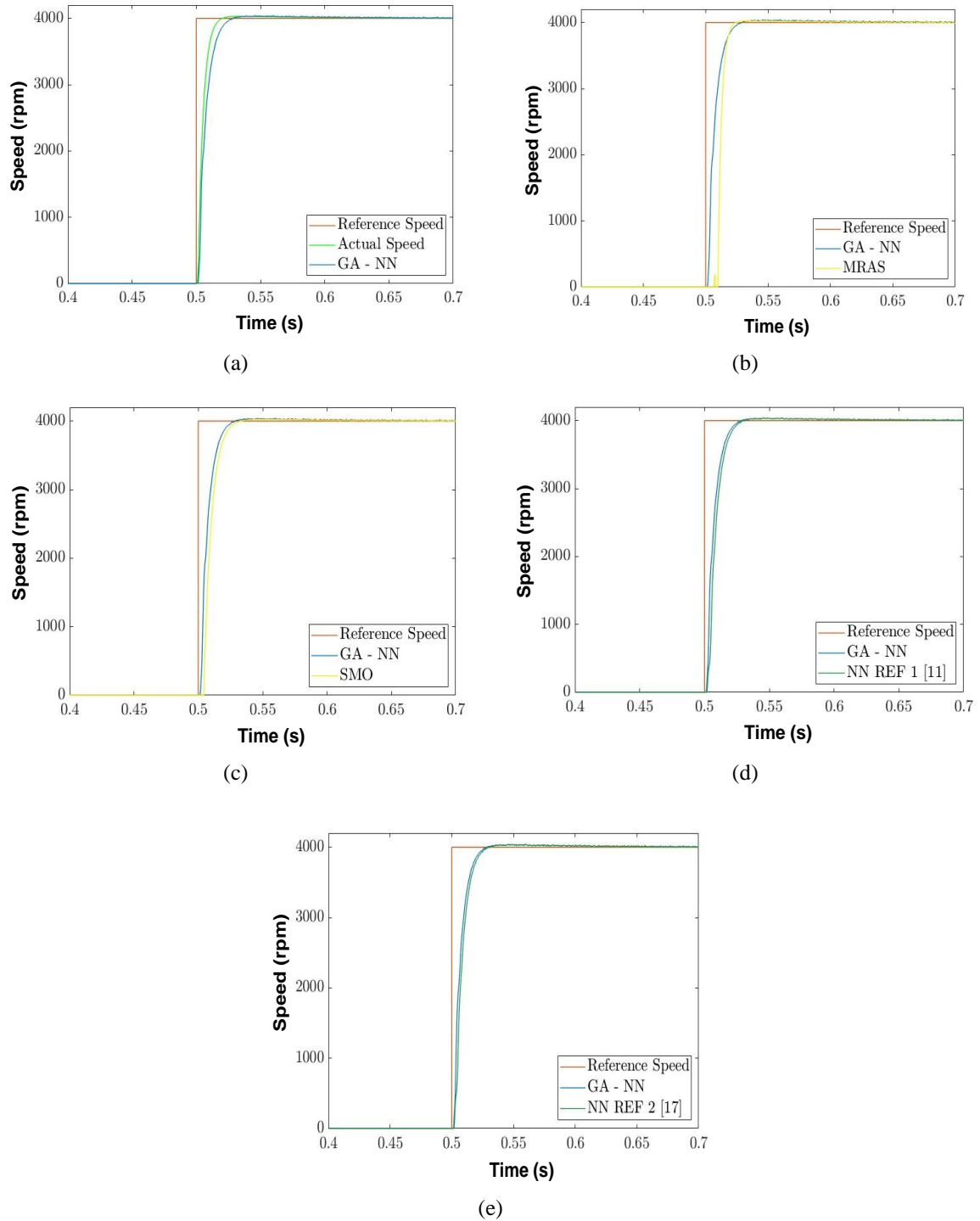


Figure 9. Step-response performance of the GA-NN against: (a) sensor (actual), (b) MRAS, (c) SMO, (d) NN REF 1 [11], and (e) NN REF 2 [17]

### 3.5. Performance at low speed

The performance of the estimators was also qualified for speed estimation at low speed. Some observers such as the back-EMF SMO are expected to perform poorly at lower speed due to the inherent nature of the model. As such, the estimators were evaluated at a 500-rpm target speed with 0.8 Nm load.

As can be observed in Figure 10, when comparing the performance of the GA-NN vs the reference estimators. There is an observable difference in terms of the peak-to-peak ripple of the estimation as shown in Figure 10(a), this makes it apparent that the inherent chattering in the SMO becomes more defined at low-speed operation that is not observable in other estimators. As such, there is no distinguishable feature in the low-speed operation with the MRAS, and the two reference NN estimators, as shown in Figure 10(b), Figure 10(c), and Figure 10(d), respectively.

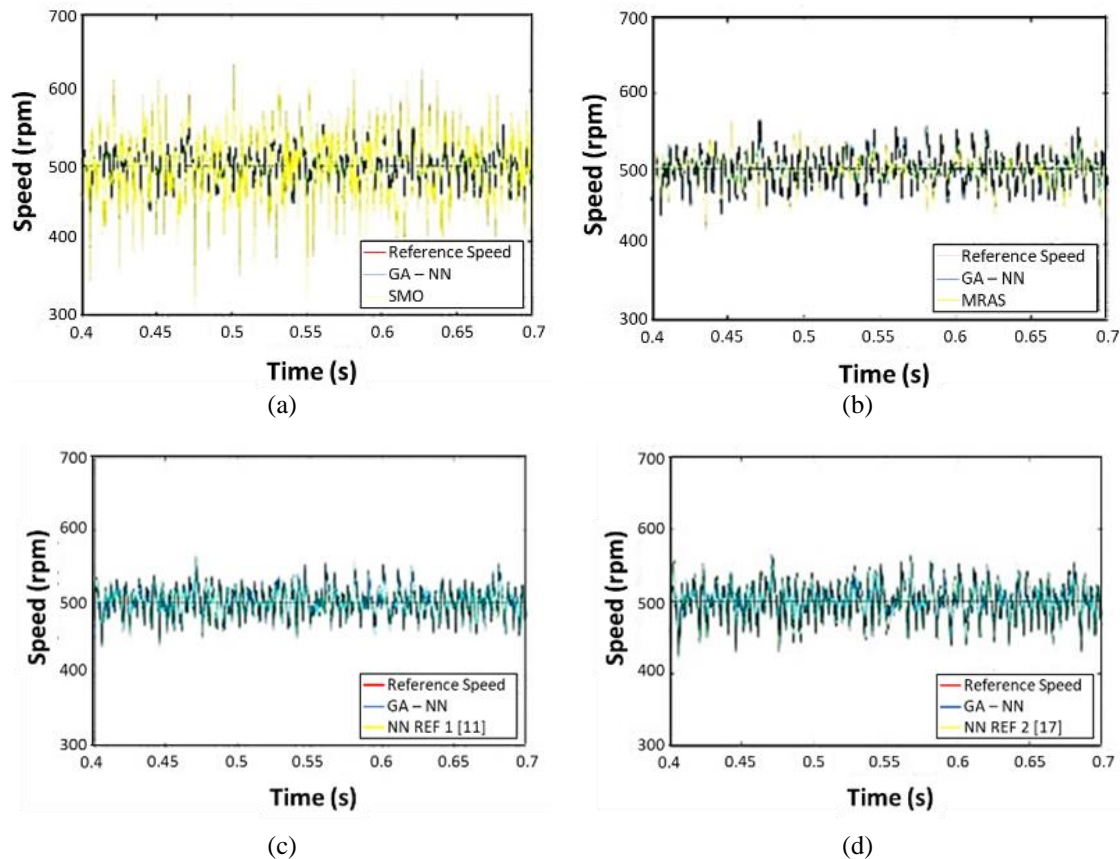


Figure 10. Low-speed estimation performance of the GA-NN against: (a) MRAS, (b) SMO, (c) NN REF 1 [11], and (d) NN REF 2 [17]

### 3.6. Efficiency

The efficiency of the FOC scheme using the estimators was compared to the efficiency when running on sensors. This was obtained by measuring the power drawn from the DC source or the battery at a steady state of 4000 rpm and 0.8 Nm. In this method, the load of the system is held constant while only updating the estimator being used, the difference in the power drawn will indicate the relative efficiency performance.

As shown in Table 5, when using the sensor, the FOC system draws the least average power compared to the FOC systems using an estimator which is expected. However, in comparison between the estimators themselves, the GA-optimized neural network managed to draw the least amount of power at 362.6 W. The performance of the GA-NN was followed closely by the conventional SMO estimator and by the NN REF 1 [11] by a margin of 0.4 W and 0.6 W respectively. These findings would indicate that the improvements in estimation in a GA-NN still comes short of the sensed performance but is a step in the right direction as it outperforms the rest of the estimators verified. The data shows that there is an improvement in efficiency when optimizing the hyperparameters of the neural network estimator as demonstrated by the performance of the proposed GA-NN.

Table 5. Efficiency summary

Estimator	Input power drawn (W)
Sensored	361.4
GA-NN	362.6
SMO	363
MRAS	363.9
NN REF 1 [11]	363.2
NN REF 2 [17]	363.8

#### 4. CONCLUSION

The design and simulation of a neural network-based speed and position estimator for a field-oriented control PMSM was successfully done in the MATLAB/Simulink environment with the neural network hyperparameters optimized via genetic algorithm. The genetic algorithm optimized the following hyperparameters: batch size, Levenberg-Marquardt training parameters (initial  $\mu$ ,  $\mu$  increase factor,  $\mu$  decrease factor), activation function for the hidden layer/s, and the number of hidden layers and its corresponding number of neurons. The trained neural network based on the genetic algorithm-optimized hyperparameters managed to attain a mean-square error loss of 0.00001140 on the validation data. The proposed GA-optimized neural network has outperformed the other estimators during steady-state condition, speed range performance, step-response performance and efficiency. In comparing the proposed network against other neural network estimators, the optimization of the hyperparameters provide significant improvement in the performance of the estimation process leading to a better steady-state and dynamic response of the FOC. During qualification, some conditions provide room for improvement for all estimators including the performance during very low speeds of below 500 rpm, and the margin between the efficiency of the system when using sensors and when using estimators.

#### ACKNOWLEDGEMENTS

The authors would like to thank everyone who was part of the ESC Project under the UAV Laboratory of De la Salle University for all the support for the conduct of this study.




#### REFERENCES

- [1] K. -W. Lee, S. Park, and S. Jeong, "A seamless transition control of sensorless PMSM compressor drives for improving efficiency based on a dual-mode operation," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1446–1456, 2015, doi: 10.1109/TPEL.2014.2316198.
- [2] M. Rashed, P. F. A. MacConnell, A. F. Stronach, and P. Acarnley, "Sensorless indirect-rotor-field-orientation speed control of a permanent-magnet synchronous motor with stator-resistance estimation," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1664–1675, 2007, doi: 10.1109/TIE.2007.895136.
- [3] T. O. -Kowalska and M. Dybkowski, "Stator-current-based MRAS estimator for a wide range speed-sensorless induction-motor drive," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1296–1308, 2010, doi: 10.1109/TIE.2009.2031134.
- [4] M. S. Zaky, M. M. Khater, S. S. Shokralla, and H. A. Yasin, "Wide-speed-range estimation with online parameter identification schemes of sensorless induction motor drives," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1699–1707, 2009, doi: 10.1109/TIE.2008.2009519.
- [5] 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, 2007, doi: 10.1109/TIE.2006.888778.
- [6] A. Dominic D. and T. R. Chelliah, "Analysis of field-oriented controlled induction motor drives under sensor faults and an overview of sensorless schemes," *ISA Transactions*, vol. 53, no. 5, pp. 1680–1694, 2014, doi: 10.1016/j.isatra.2014.04.008.
- [7] H. Kim, J. Son, and J. Lee, "A high-speed sliding-mode observer for the sensorless speed control of a PMSM," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4069–4077, 2011, doi: 10.1109/TIE.2010.2098357.
- [8] A. Iqbal and M. R. Khan, "Sensorless control of a vector controlled three-phase induction motor drive using artificial neural network," *2010 Joint International Conference on Power Electronics, Drives and Energy Systems & 2010 Power India*, 2010, pp. 1-5, doi: 10.1109/PEDES.2010.5712474.
- [9] P. Girovský, J. Timko, J. Žilková, and V. Fedák, "Neural estimators for shaft sensorless FOC control of induction motor," in *Proc. of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*, 2010, doi: 10.1109/EPEPEMC.2010.5606907.
- [10] M. A. Rafiq, M. Habibullah, and B. C. Ghosh, "Artificial neural network based speed tracking of a field oriented induction motor drive," in *2012 7th International Conference on Electrical and Computer Engineering*, 2012, pp. 315-318, doi: 10.1109/ICECE.2012.6471549.
- [11] W. Zine, Z. Makni, E. Monmasson, L. Idkhajine, and B. Condamin, "Interests and limits of machine learning-based neural networks for rotor position estimation in EV traction drives," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 1942-1951, 2018, doi: 10.1109/TII.2017.2765398.
- [12] J. -W. Song, K. -C. Lee, K. -B. Cho, and J. -S. Won, "An adaptive learning current controller for field-oriented controlled induction motor by neural network," in *Proc. IECON '91: 1991 International Conference on Industrial Electronics, Control and Instrumentation*, 1991, vol. 1, pp. 469-474, doi: 10.1109/IECON.1991.239340.
- [13] K. -B. Cho and K. -C. Lee, "An adaptive fuzzy current controller with neural network for a field-oriented controller induction machine," *International Journal of Approximate Reasoning*, vol. 10, no. 1, pp. 45–61, 1994, doi: 10.1016/0888-613X(94)90008-6.
- [14] D. Fodor *et al.*, "Neural networks applied for induction motor speed sensorless estimation," in *1995 Proceedings of the IEEE International Symposium on Industrial Electronics*, 1995, vol. 1, pp. 181-186, doi: 10.1109/ISIE.1995.496623.




- [15] G. M. -Aguilar, J. M. M. -Eguilaz, B. Pryymak, J. Peracaula, and J. A. Beristain, "A comparative analysis of two neural-network-based estimators for efficiency optimization of an induction motor drive," in *2006 IEEE International Power Electronics Congress*, 2006, pp. 1-6, doi: 10.1109/CIEP.2006.312164.
- [16] A. Goedel, I. N. Silva, P. J. A. Serni, and M. Suetake, "Estimation of electrical machine speed using sensorless technology and neural networks," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, 2008, pp. 1-6, doi: 10.1109/TDC-LA.2008.4641832.
- [17] A. Goedel and I. N. Da Silva, "Torque and speed estimator for sensorless induction motor drive using cascade neural networks," in *2006 IST IEEE Conference on Industrial Electronics and Applications*, 2006, pp. 1-6, doi: 10.1109/ICIEA.2006.257321.
- [18] A. Goedel, C. Graciola, S. A. O. Silva, C. F. Nascimento and M. Suetake, "A comparative study for single and multilayer neural networks applied to speed estimation in induction motors," *The XIX International Conference on Electrical Machines - ICEM 2010*, 2010, pp. 1-6, doi: 10.1109/ICELMACH.2010.5607949.
- [19] Y. B. Zbde, S. M. Gadoue and D. J. Atkinson, "Model predictive MRAS estimator for sensorless induction motor drives," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3511-3521, 2016, doi: 10.1109/TIE.2016.2521721.
- [20] A. K. P. Toh, E. P. Nowicki and F. Ashrafzadeh, "A flux estimator for field oriented control of an induction motor using an artificial neural network," in *Proc. of 1994 IEEE Industry Applications Society Annual Meeting*, 1994, vol. 1, pp. 585-592, doi: 10.1109/IAS.1994.345461.
- [21] T. O. -Kowalska and C. T. Kowalski, "Neural network application for flux and speed estimation in the sensorless induction motor drive," in *ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal, 1997, pp. 1253-1258 vol.3, doi: 10.1109/ISIE.1997.648923
- [22] W. W. L. Keerthipala, M. H. Chun, and B. R. Duggal, "Microprocessor implementation of field-oriented control of induction motors using ANN observers," *Microprocessors and Microsystems*, vol. 21, no. 2, pp. 105-112, 1997, doi: 10.1016/S0141-9331(97)00010-0.
- [23] J. M. G. -Villalobos, J. R. -Resendiz, E. A. R. -Araiza, and V. H. Mucino, "A review of parameter estimators and controllers for induction motors based on artificial neural networks," *Neurocomputing*, vol. 118, pp. 87-100, 2013, doi: 10.1016/j.neucom.2013.02.018.
- [24] C. -H. Tsai, "Neural network application for flux and speed estimation in the sensorless decoupling induction motor drive," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 5297-5303, doi: 10.1109/ICSMC.2006.385150.
- [25] D. D. Neema, R. N. Patel, and A. S. Thoke, "Rotor flux and torque estimator for vector controlled induction drive using ANN," in *2009 International Joint Conference on Neural Networks*, 2009, pp. 2215-2220, doi: 10.1109/IJCNN.2009.5178881.
- [26] Y. Oguz and M. Dede, "Speed estimation of vector controlled squirrel cage asynchronous motor with artificial neural networks," *Energy Conversion and Management*, vol. 52, no. 1, pp. 675-686, 2011, doi: 10.1016/j.enconman.2010.07.046.
- [27] P. Brandstetter and M. Kuchar, "Sensorless control of variable speed induction motor drive using RBF neural network," *Journal of Applied Logic*, vol. 24, pp. 97-108, 2017, doi: 10.1016/j.jal.2016.11.017.
- [28] Y. Yang, "A neural network model reference adaptive system for speed estimation of sensorless induction motor," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 6864-6868, doi: 10.1109/CCDC.2017.7978416.
- [29] J. Yang, L. Wang, D. Xu and B. Xue, "Sensorless Speed Estimation for Line-connected Induction Motor Based on Recurrent Multilayer Neural Network," in *2007 IEEE International Conference on Automation and Logistics*, 2007, pp. 2013-2018, doi: 10.1109/ICAL.2007.4338905.
- [30] A. Accetta, M. Cirrincione, and M. Pucci, "Sensorless control of PMSM by a linear neural network: TLS EXIN neuron," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, 2010, pp. 974-978, doi: 10.1109/IECON.2010.5675500.
- [31] J. Yang, "Indoor space compositions based on genetic algorithms to optimize neural networks," *Physical Communication*, vol. 42, 2020, doi: 10.1016/j.phycom.2020.101167.
- [32] M. Marchesoni, M. Passalacqua, L. Vaccaro, M. Calvini, and M. Venturini, "Performance improvement in a sensorless surface-mounted PMSM drive based on rotor flux observer," *Control Engineering Practice*, vol. 96, 2020, doi: 10.1016/j.conengprac.2019.104276.
- [33] R. Li and G. Zhao, "Position sensorless control for PMSM using sliding mode observer and phase-locked loop," in *2009 IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 1867-1870, doi: 10.1109/IPEMC.2009.5157699.

## BIOGRAPHIES OF AUTHORS






**Juan Paolo Quismundo**    is currently an applications firmware engineer at Power Integrations, previously as a firmware design engineer and electrical design engineer at Advanced Energy. He earned his M.Sc. in Electronics Engineering from De La Salle University, his post-graduate diploma in Power Electronics and his B.S. in Electronics Engineering from Mapua University. His research interests are intelligent systems, artificial neural networks, motor control, and power electronics. He can be contacted at email: jpbquismundo@gmail.com.






**Edwin Sybingco**    is a full professor of the DLSU ECE Department. He received the B.S., M.S., and PhD in Electronics and Communications Engineering from De La Salle University in 1990, 1993, and 2018. He is currently a faculty member in Electronics and Computer Engineering Department, De La Salle University, where he teaches courses related to signal processing, machine learning, computer vision, and control systems. His principal research interest includes various topics in signal processing, intelligent transport system, robotics, computer vision, and computational intelligence. He can be contacted at email: edwin.sybingco@dlsu.edu.ph.






**Maria Antonette Roque**    is an assistant professor of the Department of Electronics and Computer Engineering. She is currently the associate dean of the Gokongwei College of Engineering. She holds a degree in BS Electronics and Communications Engineering and MS in Electronics and Communications Engineering from De La Salle University - Manila. Her research areas include digital design, networking and interfacing, database systems, electronic circuit design, unmanned aerial vehicle, mobile applications development, and computing and information technology. She can be contacted at email: antonette.roque@dlsu.edu.ph.



**Alvin Chua**    is a full professor of the Mechanical Engineering Department of De La Salle University, Philippines. He earned his BSME, MSME, and Ph.D. in ME at De La Salle University-Manila. As a scholar under the Department of Science and Technology-Engineering and Science Education Project (DOST-ESEP), he conducted his dissertation research at the University of New South Wales, Australia. He received an award as fellow in the 2020 Leaders of Innovation Fellowship, sponsored by the Royal Academy of Engineering, UK, Asian Institute of Management and Department of Science and Technology. He has published several papers in international journals and conferences. His research interests are on mechatronics, drones, robotics, optimal estimation, and controls. He can be contacted at email: alvin.chua@dlsu.edu.ph.



**Leonard Ambata**    is a licensed Electronics Engineer and a graduated of De La Salle University - Manila with a degree in Electronics & Communications Engineering last 2003, and MS-ECE in 2007. He is currently employed by DLSU-Manila as an academic Service Faculty of the Electronics & Communications Engineering Department and is working on dissertation for PHD-ECE. He can be contacted at email: leonard.ambata@dlsu.edu.ph.