

ABSTRACT

SENSORLESS SPEED ESTIMATION OF AN AC INDUCTION MOTOR BY USING AN ARTIFICIAL NEURAL NETWORK APPROACH

Abdulelah Alkhoraif, M.S.
Department of Electrical Engineering
Northern Illinois University, 2015
Donald S. Zinger, Director

Sensorless speed detection of an induction motor is an attractive area for researchers to enhance the reliability of the system and to reduce the cost of the components. This paper presents a simple method of estimating a rotational speed by utilizing an artificial neural network (ANN) that would be fed by a set of stator current frequencies that contain some saliency harmonics. This approach allows operators to detect the speed in induction motors such an approach also provides reliability, low cost, and simplicity. First, the proposed method is based on converting the stator current signals to the frequency domain and then applying a tracking algorithm to the stator current spectrum in order to detect frequency peaks. Secondly, the ANN has to be trained by the detected peaks; the training data must be from very precise data to provide an accurate rotor speed. Moreover, the desired output of the training is the speed, which is measured by a tachometer simultaneously with the stator current signal. The databases were collected at many different speeds from two different types of AC induction motors, wound rotor and squirrel cage. They were trained and tested, so when the difference between the desired speed value and the ANN output value reached the wanted accuracy, the system does not need to use the tachometer anymore. Eventually, the experimental results show that in an optimal ANN

design, the speed of the wound rotor induction motor was estimated accurately, where the testing average error was 1 RPM. The proposed method has not succeeded to predict the rotor speed of the squirrel cage induction motor precisely, where the smallest testing-average error that was achieved was 5 RPM.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

DECEMBER 2015

SENSORLESS SPEED ESTIMATION OF AN AC INDUCTION MOTOR BY USING AN
ARTIFICIAL NEURAL NETWORK APPROACH

BY

ABDULELAH ALI ALKHORAIF
©2015 Abdulelah Ali Alkhoraif

A THESIS SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

Thesis Director:
Donald S. Zinger

ACKNOWLEDGEMENTS

First and foremost, I would like to give special acknowledgment to my God (الله) for giving me the strength and knowledge to continue studying electrical engineering.

I would like to express my sincere gratitude to my advisor Prof. Donald S. Zinger for the continuous support of my M.S. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M.S. study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Reza Hashemian, Prof. Veysel Demir, for their insightful comments and encouragement, but also for the hard question which incited me to widen my research from various perspectives.

My sincere thanks and appreciation to my lovely parents Ali Alkoraif and Shikah Almasalam, for giving me everything they could. I can never give enough gratitude. I would also like to thank all my brothers and sisters for supporting me in my study.

Last but not the least, I would like to thank my lovely wife Hajar. Her patience, understanding, and encouragement were the real motivation to accomplish this thesis.

DEDICATION

I dedicate this thesis to my beloved father and mother,

my brother Ibrahim,

and my darling wife

“You are all my happiness.”

TABLE OF CONTENTS

	Page
LIST OF TABLES	VII
LIST OF FIGURES	VIII
Chapter	
1. BACKGROUND	1
1.1 An Introduction of the Proposed Method	1
1.2 Thesis Objective	3
1.3 Thesis Overview	3
2. AC INDUCTION MACHINES	5
2.1 Speed Measurement of Induction Machines	5
2.2 Speed Measurement Using Transducer	7
2.3 Sensorless Speed Measurement	9
2.3.1 Adaptive Method	11
2.3.2 Tracking Saliency Harmonics	15
2.3.3 Artificial Intelligence	16
2.4 Delimitations and Limitations	17
2.5 Assumptions	18

Chapter		V Page
3. THE LITERATURE REVIEW OF SALIENCY HARMONICS	19	
3.1 Introduction	19	
3.2 Saliency Harmonics	19	
3.2.1 Rotor Eccentricity	20	
3.2.2 Slot Harmonics	25	
3.3 Fourier Fast Transform (FFT)	27	
3.4 Tracking Algorithms	27	
3.5 Using ANN for Sensorless Speed Estimation	28	
4. ARTIFICIAL NEURAL NETWORKS	29	
4.1 Introduction	29	
4.2 The Principle of Neuron Weights	31	
4.3 Why ANNs?	34	
4.4 Designing ANNs	36	
5. METHODOLOGY	39	
5.1 Introduction	39	
5.2 Procedures	39	
5.2.1 Stage 1: Monitoring the Stator Current	41	
5.2.2 Stage 2: Transfer the Data from Time Domain to Frequency Domain	44	
5.2.3 Stage 3: Tracking the Rotor Saliency Harmonics	46	
5.2.4 Stage 4: ANN Estimation	50	
6. RESULTS	56	

Chapter	VI Page
6.1 Extracting Harmonics	56
6.2 Results	60
6.2.1 Data A Results	61
6.2.1.1 One Hidden Layer	61
6.2.1.2 Two Hidden Layers	64
Neurons Variation	64
Width Variation	67
Peaks Variation	69
6.2.2 Data A+B Results	71
6.2.3 Data C Results	74
6.2.4 Data D Results	78
7. CONCLUSION	86
7.1 Discussion	86
7.2 Summary	89
7.3 Conclusion	90
7.4 Future Work	91
REFERENCES	92
APPENDIX: THE MATLAB CODE OF THE PROPOSED METHOD	96

LIST OF TABLES

Table	Page
4.1. One neuron input, weights, and output with threshold function for two inputs	32
5.1. The three motors descriptions	42
5.2. The collected data	45
5.3. The calculation of the maximum saliency harmonics based on 8% slip	48
5.4. The approximate width for each window	48
5.5. The worst accuracy	54
5.6. Accuracy scales	54
6.1. The approximate width for the second window of pattern 24 in Table 6.3	59
6.2. The peaks and differences for data A	81
6.3. The peaks and differences for data A+B	82
6.4. The peaks and differences for data C	83
6.5. The peaks and differences for motors A, data D	84

LIST OF FIGURES

Figure	Page
Figure 2.1: Shaft-mounted tachometer (rotary encoder)	8
Figure 2.2: Shaft encoder transducers	8
Figure 2.3: A type of shaft sensor called optical incremental encoder	8
Figure 2.3: Adaptive scheme	13
Figure 2.5: Model reference adaptive systems for an ACIM speed estimation	14
Figure 3.1: Air-gap eccentricities	21
Figure 3.2: Stator-rotor slot in an IM	25
Figure 4.1: Components of a biological neuron	30
Figure 4.2: The neuron model	30
Figure 4.3: One neuron structure with threshold function for two inputs	32
Figure 4.4: A net of neurons structure (ANN)	35
Figure 4.5: A linear activation function	37
Figure 4.6: The hyperbolic tangent activation function	37
Figure 5.1: The overall structure of ANN sensorless speed estimation	40
Figure 5.2: The general structure of the lab experimental setup	41
Figure 5.3: A & B (WRIMs) connections	43
Figure 5.4: C (SCIM) connections	44

Figure	Page
Figure 5.5: ANN structure	51
Figure 5.6: The flowchart of the proposed method	55
Figure 6.1: A quarter of the stator current signal at 1747 RPM, pattern 24	57
Figure 6.2: The stator current spectrum at 1747 RPM, pattern 24 in Table 6.3	58
Figure 6.3: The windows of the tracking algorithm	58
Figure 6.4: The 5th window of the tracking algorithm on pattern 24 in Table 6.3 ...	59
Figure 6.5: The prediction errors of data A, (AI1)	61
Figure 6.6: The prediction errors of data A, (AI2)	62
Figure 6.7: The prediction errors of data A, (AI3)	63
Figure 6.8: The prediction errors of data A, (AII1)	64
Figure 6.9: The prediction errors of data A, (AII2)	65
Figure 6.10: The prediction errors of data A, (AII3)	66
Figure 6.11 : The prediction errors of data A, (AII4)	67
Figure 6.12 : The prediction errors of data A, (AII5)	68
Figure 6.13 : The prediction errors of data A, (AII6)	69
Figure 6.14: The prediction errors of data A, (AII7)	70
Figure 6.15 : The prediction errors of data A+B, (A+BII1)	71
Figure 6.16: The prediction errors of data A+B, (A+BII2)	72
Figure 6.17: The prediction errors of data A+B, (A+BII3)	73
Figure 6.18 : The prediction errors of data C, (CII1)	74
Figure 6.19 : The prediction errors of data C, (CII2)	75

Figure		X Page
Figure 6.20 : The prediction errors of data C, (CII3)	76	
Figure 6.21 : The prediction errors of data C, (CII4)	77	
Figure 6.22: The prediction errors of data D, (DII1)	78	
Figure 6.23: The prediction errors of data D, (DII1)	79	

CHAPTER 1

BACKGROUND

1.1 An Introduction of the Proposed Method

Sensorless speed estimation is a method to measure the rotor speed of induction motors (IMs) without any attached device that can sense the rotational movement. The conventional speed sensors such as encoders or tachometers negatively affect the reliability of IMs because of associated drawbacks. Therefore, many researchers recently are trying to find a proper way not only to estimate the rotor speed but also to increase the reliability and performance of the system.

First of all, many speed transducers need extra cable and a voltage source to run, so their prices will be added the expensive cost of the device itself. Also, the conventional sensors are often attached to the rotor shaft, which in turn increase the inertia of the rotor and the size of the machine. Therefore, these sensors might get damaged if they are running in a tough environment, which means they will also require frequent maintenance. These disadvantages can be wiped out if a successful sensorless speed method is used instead.

Therefore, this paper will present an intelligent technique to estimate the rotor speed in IMs by utilizing the speed-dependent harmonic components, "saliency harmonics," that can be extracted from the stator current spectrum. The selected motors are two identical wound rotor induction motors (WRIMs) and a squirrel cage induction motor (SCIM). The artificial neural network (ANN) is the proposed intelligent technique in this research because of the high prediction ability. Thus, there will be some lab experiments to collect real data as much as possible from each motor in order to train and test the ANN. The ANN inputs will be a particular

number of stator current harmonic peaks that hopefully will contain some of the saliency harmonics of the IM. It should be mentioned that the WRIMs and the SCIM will be examined individually.

Monitoring the stator current signal is the first step of the alternative way to estimate the rotor speed. Then, the second step is to transfer the stator current signal to the frequency domain so that the certain peaks can be extracted based on a tracking algorithm. Finally, these peaks will be the ANN inputs while the output is the rotor speed that has to be measured simultaneously with the stator current signal. The final design of each ANN for each data will be evaluated by the testing results of ANNs accuracy which will also prove whether or not the proposed method is workable.

If the desired accuracy is achieved, the system can get rid of the unreliable device, which is the speed transducer. The primary benefit of removing the speed sensor is increasing the system performance and reliability. However, there is an arguable point about the advantage of the omitted speed sensor cost in case the proposed method costs more than a speed sensor. This issue needs to make a price survey of the proposed method implementation and the speed sensor, but in some cases the proposed method does not need an extra cost. For instance, the speed value of IMs in industrial fields are urgently required in control applications, and the stator current or voltage signal are always monitored either by a personal computer (PC) or a control board. Thus, the purpose of getting rid of speed transducers in this research is not only to estimate the speed but also to obtain the benefit of the existent PC or control board.

Eventually, this paper undertakes to evaluate a new sensorless speed estimation method that can be applied to IMs by utilizing an artificial neural network approach. Making the system

smart and reliable is a big motivation for many researchers around the world, and that might be accomplished in IMs by implementing the proposed method.

1.2 Thesis Objective

The primary objective of this research is to determine the rotor speed of an induction motor by utilizing an artificial neural network that is fed from the rotor saliency harmonics. Eventually, after training process, the system will get rid of the speed sensor in order to enhance the reliability and performance of the ACIM.

1.3 Thesis Overview

The thesis is composed of seven chapters that are organized in the following way. Chapter 1 introduced the sensorless speed estimation. Chapter 2 covers the problems associated with speed measurement devices and states several sensorless speed methods that have been recently used to enhance the performance and reliability of ACIMs.

Chapter 3 presents a review of tracking saliency methods, discussing their causes and benefits, which can provide the speed information. Also, Chapter 3 proposes the ANN sensorless speed approach in order to measure the ACIMs rotor speed instead of transducers. Chapter 4 gives a brief introduction and some applications of ANNs , and the chapter explains ANNs procedures through an example.

Chapter 5 covers the methodology for each stage of the proposed method. The tasks that have to be followed are addressed intensively, where the sensorless speed accuracy of the system can be measured. demonstrates the results of the proposed method, and point out an investigation

with several factors that can change the speed accuracy. Chapter 6 shows the results of the proposed method affected by several factors that can change the speed accuracy. Finally, Chapter 7 contains the overall discussion and conclusion of the proposed method; also suggests a possibility for future work.

CHAPTER 2

AC INDUCTION MACHINES

2.1 Speed Measurement of Induction Machines

AC induction motors (ACIMs) are the most widely used machine in consumer and industrial applications. ACIMs have played a significant role in the high productivity of modern industry. In fact, ACIMs are dominating 85% of electrical machines because of many advantages, some of which are the simple and rugged construction, low cost and minimum maintenance, and high reliability and sufficiently high efficiency. Also, one of the main features is the ability to connect ACIMs directly to an AC power source without any converter. Furthermore, more than half of total industrial electricity consumption is caused by ACIMs [1]. Therefore, enhancing the reliability of ACIM would certainly help in saving energy.

ACIMs are generally categorized based on rotor types: squirrel cage or wound rotor. The stationary part of squirrel cage and wound rotor IMs is the same while the rotor part is different. In wound rotor, insulated windings will be placed into the rotor slots but there will be copper or aluminum bars installed into the rotor slots in squirrel cage. For both types, there is a rotating magnetic field generated in the stator when it is supplied by polyphase AC. The speed of the stator's magnetic field rotation is known as the synchronous speed (N_s). N_s is calculated by multiplying the electrical supply frequency (f_e) times the constant 60, then dividing the result by the number of pole pairs p in the motor. The actual speed at which the rotor actually runs is the rotor speed (N_r). Because of the fact that the rotor runs at a speed less than synchronous speed,

these types of motors are called asynchronous motors or "induction motors," where the normalized difference between rotor speed and synchronous speed is called "slip."

$$N_s = \frac{60 * f_e}{p} \quad (2.1)$$

$$\text{Slip } (s) = \frac{Ns - Nr}{Ns} \quad (2.2)$$

$$f_r = \frac{p * N_r}{60} \quad (2.3)$$

$$f_r = f_e * (1 - s) = f_e - f_s \quad (2.4)$$

where:

$$f_s = f_e * s \quad (2.5)$$

fs : slip frequency

fe : rotor frequency

In motor control applications, the designers consider the rotor speed as the key element to control the system because the primary goal in AC motor control is to make the rotor rotate at a desired speed despite load variations. Thus, monitoring the speed is extremely important in industrial and engineering applications of IMs. Ultimately, many industrial processes will inevitably rely on ACIMs because of their advantages.

2.2 Speed Measurement Using Transducer

The conventional device that can track the shaft rotational speed for an IM is a transducer. It is known as a sensor device that converts energy from one form to another. Examples of speed measurement transducer are resolvers, encoders, or tachometers, where they are classified based on their function of sensing the rotor's speed, direction, or position. The most popular used device to measure an ACIM speed is a shaft-mounted tachometer or a rotary encoder as depicted in Figure 2.1. For industrial applications, rotary encoders that are coupled to the shaft are preferred in order to ensure precise speed and positioning. In addition, they do not require an analog to digital converter; they convert rotary movement into digital signals, which can be sent directly back as a feedback signal to the control panel.

An example of a rotary encoder is shown in Figure 2.2, and it is an optical incremental encoder which is considered the most popular shaft speed sensor. It consists of a rotating wheel, a light source, a light sensor (receiver), and an information cable. The disk, which is coupled to the rotating shaft has a certain number of slots as illustrated in Figure 2.3. As the shaft rotates, these slots interrupt the light emitted into the light sensor, generating a digital or pulse signal output. These pulses will be counted by a counter per second. Consequently, as the number of slots and pulses are known, the rotational speed can be measured by using equation in Figure 2.3. All other types of the rotational transducer have the same principle to measure the speed, regardless of the slight technical difference between each type.

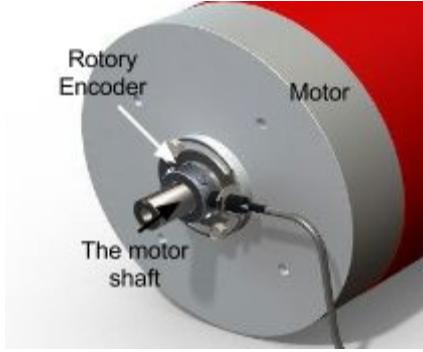
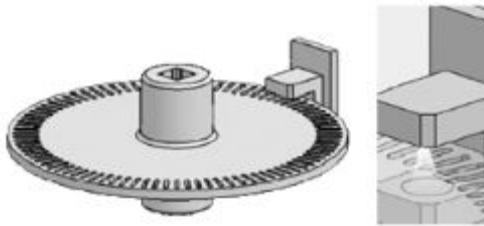


Figure 2.1: Shaft-mounted tachometer (rotary encoder).



Figure 2.2: Shaft encoder transducers.



$$\text{Speed (Nr)} = \frac{\text{Pulses per second}}{\text{Number of slots}} \text{ r.p.s}$$

Figure 2.3: A type of shaft sensor called optical incremental encoder.

In spite of the fact that rotational transducers have been highly developed, all these types of sensors have been causing many obstacles. There is absolute agreement on the negativity of the speed sensor device, where these articles [1-11][20][22] address that rotational transducers are associated with many drawbacks. First of all, an additional device of transducers needs cable and a DC power supply (12-24VDC) to run; and it increases the cost of the systems. For instance, the rotary encoder costs the same price as a low power IM (2-5Kw). In addition, the average cost of a speed sensor at 50 kW IM is 25% of the machine price [6].

Rotary encoders are sensitive to harsh environmental conditions such as humidity and high temperatures. Since ACIMs are more likely to be running in a hostile or aggressive environment, it is inappropriate to install the speed sensor on the motor shaft because it might be the weakest part of the system [12]. On the other hand, attaching a rotary sensor to the motor shaft not only increases the size but will also increase the inertia of the rotor. In consequence of all previous disadvantages, the system will require frequent maintenance because the installation of speed sensors will reduce the reliability and robustness of the system[1-11][22].

Accordingly, increasing attempts have been made to operate ACIM "sensorless," without rotational transducer. Therefore, many researchers are proposing a method to eliminate the encoder mounted to the motor shaft without affecting the performance. Sensorless speed measurement of ACIMs has been extensively researched in the last few decades due to its various advantages.

2.3 Sensorless Speed Measurement

Most industrial applications that use ACIMs urgently require speed information. Because of the fact that ACIMs with rotational transducers suffer from weaknesses in both their reliability and performance as mentioned earlier, the sensorless method is the proper technical solution to overcome the rotational transducer's problems. The sensorless speed measurement approach achieves high objectives, some of which are [3] [5] [12-14]:

1. Reducing the size of the machine
2. Reducing the cost of the system
3. Reducing hardware complexity

4. Decreasing maintenance requirements
5. Increasing the reliability

The main purpose of using sensorless speed estimation of ACIMs is to achieve most of the previous advantages. Sensorless speed measurement can basically be divided into two groups: mathematical technique and signal injection technique [1] [15]. Recently, many researchers have tried a new method in which one of the previous techniques is combined with an artificial intelligence (AI) such as artificial neural network (ANN) as used in [13] [16-18], despite the fact that not all of them built ANNs to estimate the rotor speed directly. However, using AI can be considered as the third technique to estimate the rotor speed.

The mathematical technique depends on an adjustable model that uses the dynamic equation of IM while the signal injection is preferred when the mathematical technique is failed at very low speed. On the other hand, the artificial intelligence can be a fascinating technique where it can work successfully without any mathematical equation and an injected signal. By way of illustration, there are several approaches of sensorless speed measurement that have been proposed in the last decade. There is no precise agreed classification of the sensorless speed measurement types among researchers in [1][5][19-21] . However, the following categories are the most researched subjects:

1. Adaptive or observer method
2. Tracking saliency harmonics by signal injection
3. Artificial Intelligence

2.3.1 Adaptive Method

Adaptive methods are also called observer methods, and they are considered the most common technique of sensorless speed measurement. The adaptation mechanism of these methods has the ability to adapt itself to the changes in the process in order to achieve a desired output. Most adaptive methods are derived from space vectors equations as written in Equations 2.6 and 2.7 [1][8] [11] [20] [22]. There are different types of adaptive methods: model reference adaptive systems, sliding mode observer, Luenberger observer, Kalman filter, and full order & reduced order closed loop observers [1] [8] [20].

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \quad (2.6)$$

$$\underline{y} = \mathbf{C}\underline{x}$$

$$\dot{\hat{\underline{x}}} = \hat{\mathbf{A}}\hat{\underline{x}} + \mathbf{B}\underline{u} + \mathbf{K}(\underline{y} - \hat{\underline{y}}) \quad (2.7)$$

$$\hat{\underline{y}} = \mathbf{C}\hat{\underline{x}}$$

x: the state vector

y: the output vector

u: the system input vector

A,B, and C: the matrix parameters

K: a gain

In order to estimate the rotor speed (w_r) of IM, all these methods must depend on the mathematical model of the IM. Therefore, the dynamic behavior of ACIM in the stationary reference frame is shown below in d-q coordinate [11] [20][23].

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} i_s \\ \phi_r \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} i_s \\ \phi_r \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} V_s \quad (2.8) \\ \frac{d}{dt} \begin{bmatrix} i_s \\ \phi_r \end{bmatrix} &= A x + B v_s \\ i_s &= C x \end{aligned}$$

where:

$$i_s = [i_{ds} \quad i_{qs}]^T : \text{Stator Current}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\phi_r = [\phi_{dr} \quad \phi_{qr}]^T : \text{Rotor Flux}$$

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad C = [I \quad 0]$$

$$v_s = [v_{ds} \quad v_{qs}]^T : \text{Stator Voltage}$$

$$A_{11} = -\{R_s/(\sigma L_s) + (1-\sigma)/(\sigma \tau_r)\} I$$

$R_s \ R_r$: Stator and Rotor Resistance

$$A_{12} = M/(\sigma L_s L_r) \{(1/\tau_r)I - \omega_r J\}$$

M : Mutual Inductance

$$A_{21} = (M/\tau_r)I$$

ω_r : Motor Angular speed

$$A_{22} = -(1/\tau_r)I + \omega_r J$$

$L_s \ L_r$: Stator and Rotor Self Inductance

$$B_1 = (1/\sigma L_s)I$$

τ_r : Rotor Time Constant, $\tau_r = L_r/R_r$,

σ : Leakage Coefficient, $\sigma = 1 - M^2/(l$

A single type of the adaptive methods will be presented to clarify their principal. Model reference adaptive systems is the most commonly used between all adaptive methods. The

general structure of model reference adaptive systems consists of a reference model, an adjustable model and adaptive scheme or observer as illustrated in Figure 2.4 [1][11][13][22] [24][25]. The reference model uses Equation 2.6, while the adjustable model uses the adaptive, "the state observer," Equation 2.7, which is derived from Equation 2.6 [11][20]. Model reference adaptive systems can estimate the rotor speed based on rotor flux, back e.m.f., reactive power, or artificial neural network [17][24].

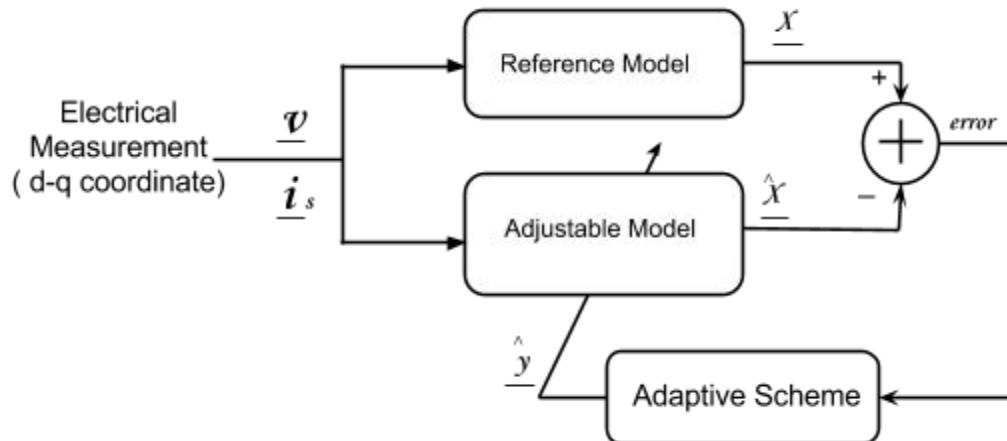


Figure 2.4: Adaptive scheme.

Since the ACIM can be presented in state equations, the adaptive methods can estimate the state variables such as stator current, rotor current, stator flux, rotor flux, and rotor speed. Therefore, model reference adaptive systems has been widely used with speed sensorless field oriented control (FOC) of ACIMs because they are based on the dynamic equation of IM where the voltages, currents and fluxes are expressed in space vector forms [1][11][13][22][24][25]. For speed measurement of ACIMs, the model reference adaptive systems can be constructed as

shown in Figure 2.5 [1]. Based on the dynamic equation of ACIM, Equation 2.8, the model reference adaptive systems will be fed by the stator voltages and the stator currents. In addition, this method will consider the rotor speed (w_r) as an unknown parameter and it will use the error as a feedback to modify the adjustable model in order to estimate the rotor speed (w_r). It should be noted that the error is the difference between the actual state variables s and the estimated state variables \hat{s} .

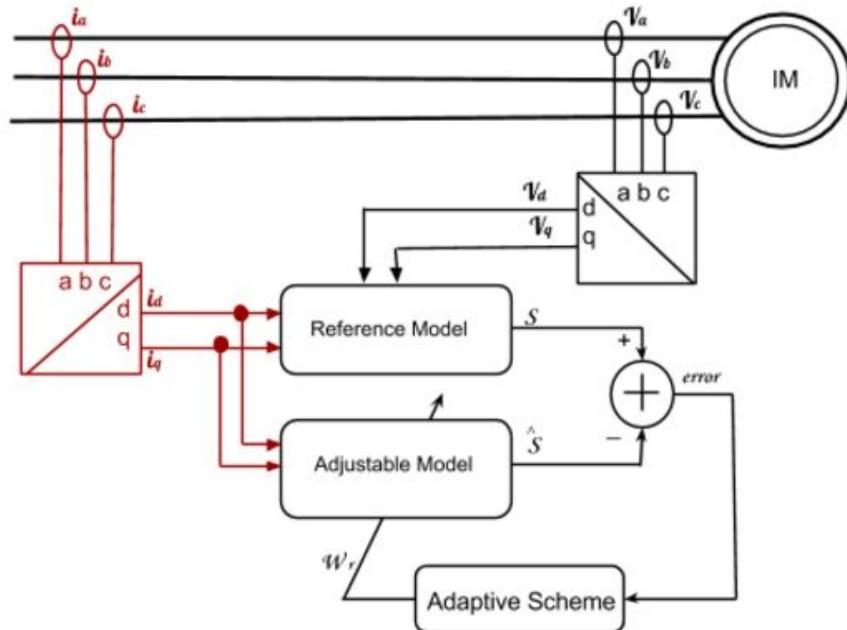


Figure 2.5: Model reference adaptive systems for an ACIM speed estimation.

The previous estimation methods, adaptive or observer methods, are sensitive to parameter variations such as stator and rotor resistance in induction motors because these methods are based on the state equation of IMs. Consequently, an unacceptable error will occur in different operating points [16]. In addition, because of the fact that any method that is

dependent of time-varying motor parameters is going to fail at zero speed, the adaptive method will be a worthless approach at zero speed. This type of technique has a long time delay that will negatively affect speed estimation during a transient [26].

2.3.2 Tracking Saliency Harmonics

Saliency methods that are independent of time-varying motor parameters have gained attention due to their ability to overcome the problem of parameter variations where they vary in various load conditions [26]. For example, signal injection is a good solution to estimate unobservable rotor speed at low or even zero stator frequency where adaptive methods that depend on speed dependent phenomenon (back EMF) totally fail [1][6][27][28]. Therefore, another way to classify sensorless approaches has been proposed in [14] depending on the speed range. For low and zero speed, carrier signal is preferred. For medium to high speed, a mathematical model is selected to adapt the IM's equations based on the electromotive force (EMF).

Signal injection is one of the techniques that is used to extract the spatial harmonics of the rotor. Signal injection is more likely to be used for rotor position estimation than speed estimation [27][28]. However, to obtain speed information with this method, a high-frequency signal needs to be injected into either the stator or the rotor side of the machine. The injected signal will interact with the saliency and produce a carrier signal current that contains the speed and position information that can be extracted from the negative-sequence component [1][27]. In other words, the high-frequency voltage excites the stator windings, so saliency-dependent

currents are generated. Thus, the stator current will definitely contain the saliency harmonics that would provide the rotor speed information if they were extracted accurately.

In this method, the magnitude and frequency of the injected carrier signals has to be selected precisely in order to not negatively affect the sensorless performance. Otherwise, it might create torque ripple and harmonic noise if the injected signal is at a high frequency. However, if its frequency is low, it would be difficult to extract the carrier signals from the line frequency. Thus, tracking saliency harmonics without any signal injection is considered a good technique to provide robust speed estimation [26].

In addition, the speed sensorless methods that have been proposed lately, among all their structures in [1][11][13][22][24][25], it's obvious to see that there must be at least two current transducers and two voltage transducers to be able to estimate the ACIM speed. These methods have also a disadvantage which is that IM's parameters should be known such as number of poles, number of slots, stator and rotor resistance, etc., in order to apply one of their methods. Here, artificial intelligence plays a significant and decisive role where artificial neural network provides the ability to compute the dynamic equation of the ACIM without any information about any parameter of the motor.

2.3.3 Artificial Intelligence

Artificial intelligence is a significant approach that is utilized to make the industrial applications work and react intelligently. Some of its types are fuzzy systems neural networks. The most important properties are: they do not usually require a mathematical model of the

system, they can reject noise, and they can easily be extended and developed [20]. The artificial neural network (ANN) is the selected method of this paper.

It has been demonstrated that speed measurement cannot be avoided in any industrial applications that use ACIMs, so the sensorless technique must provide the value of rotor speed. Thus, the sensorless speed estimation became a mandatory mission in sensorless industrial applications, and it is needed to meet the desirable performance level. Because rotational transducers are facing many obstacles, researchers are encouraged to investigate, develop, or even design a sensorless speed estimation method. As illustrated in this chapter, most sensorless methods have their limitations, advantages and disadvantages. However, tracking saliency harmonics using ANN is the chosen method of this paper.

The rotor saliency harmonics can be detected by monitoring the stator current or voltage of one phase of the three-phase IM. Some harmonics components of stator current in the machine are caused by the rotor saliency that can provide the ability to detect not only the speed [2][5] [8] [9] [14] [19] [26] [27][30][31], but also the position and the fault [19][27-29] [32-35]. Thus, rotor saliency harmonics are the key element of the sensorless speed estimation in this paper. They will be used as inputs of ANN.

2.4 Delimitations and Limitations

There is a delimitation using this method. The acquisition time of the most data is selected to be one second so that the delay might be a big concern. Also, the value of the acquisition time will affect the FFT resolution that is just 1 Hz. The probability of decreasing the acquisition time will be discussed in Chapters 5-7. Finally, the operation is chosen to be in the

steady state conditions to examine the proposed method. If it succeeds, there will be a future work to include low-speed condition.

The proposed method has some limitations: the number of sampling data, and the source frequency. They were connected directly to the voltage source of 60 Hz. The reason is that the frequency converter device in the lab creates a high noise compared to the saliency harmonics. Therefore, it would be so difficult for the tracking algorithm to extract the saliency harmonics from the high noise level.

2.5 Assumptions

It is assumed that all three-phase harmonic voltage systems are balanced. Also, the source voltage and the motor winding are balanced and healthy. This assumption is just for using the equation of the balanced condition to find the window width of the tracking algorithm. The other assumption is that saliency harmonics are located in the left sided of the range for each odd-order harmonics (3rd, 5th, etc.).

CHAPTER 3

THE LITERATURE REVIEW OF SALIENCY HARMONICS

3.1 Introduction

Since there is no sensor to measure the rotor speed, there must be a way to detect the speed information. The most widely used solution associated with IMs is tracking the saliency harmonics. These harmonics have been found to be the best tool that can provide the speed information. In addition, the stator current or voltage is the most common way that has been used in [2][5][8][9][14][19][26-36] to obtain the saliency harmonics that are related to the rotor speed. In other words, in order to detect the rotor speed from the stator current or voltage signal spectrum, there must be some harmonics depending on the rotor speed. Therefore, this section will focus on determining some inevitable harmonics that are influenced by the rotor speed and investigating sensorless methods that use these harmonics [19]. This type of technique is an extended approach of tracking saliency harmonics to estimate the rotational speed, which is discussed in Section 2.3.2. This chapter will present the phenomenon of saliency harmonics in ACIMs and how it can benefit the sensorless speed method for estimating the rotor speed.

3.2 Saliency Harmonics

There are two major causes that create saliency harmonics in IMs. The first one is generated by rotor eccentricity, while the second is by rotor slot harmonics [26][31][32][37].

These two types have been investigated separately or together for many reasons, some of which are:

- estimating or controlling rotor speed [2][5][8][9][14][19][26][27][30][31]
- estimating rotor position [27][28]
- diagnosing faults [19][29][32][33][35]
- estimating rotor and stator resistors [16]
- determining rotor slot number [38]

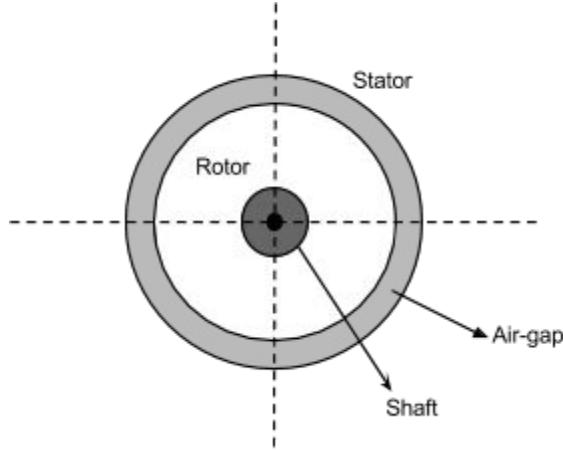
3.2.1 Rotor Eccentricity

Rotor eccentricity is also known as air-gap eccentricity [37]. The simple definition of "rotor eccentricity" is that the rotational movement does not have an ideal center with respect to the rotor itself or the stator or both, so the air gap will also be eccentric. There are many mechanical problems that might cause air-gap eccentricity, some of which are [32][37][39]:

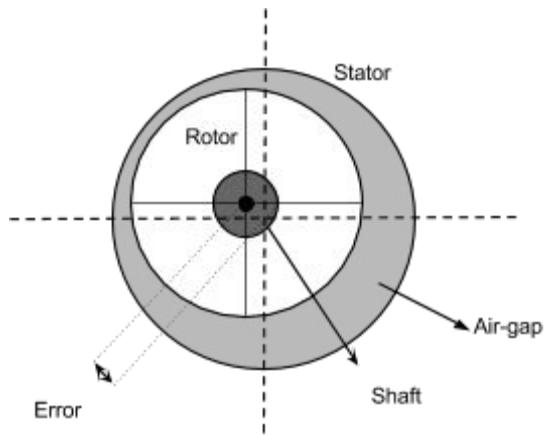
- shaft deflection
- worn bearings
- incorrect rotation center of the rotor with respect to the stator
- errors during assembly

In other words, rotor eccentricity means the air gap between the stator and the rotor is unequal (Figure 3.1). There are two forms of air-gap eccentricity [32][37][39]:

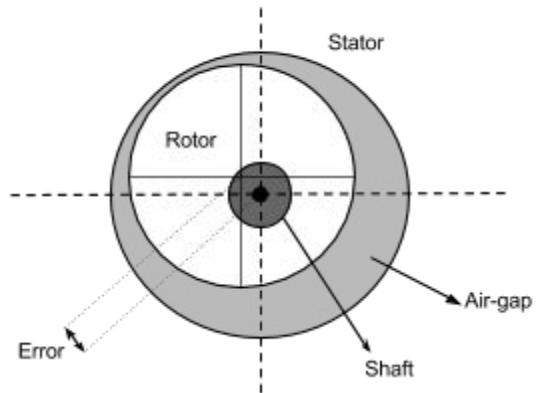
1. static eccentricity
2. dynamic eccentricity



a. Ideal



b. Static



c. Dynamic

Figure 3.1: Air-gap eccentricities.

First of all, if an IM were ideal, the rotor's center of rotation would be perfectly located at the center of the stator as illustrated in Figure 3.1.a. This assumption cannot happen in reality because there is no single machine that has an ideal design [32]. Thus, either static or dynamic

eccentricities or both definitely will exist. Therefore, rotor eccentricity plays a very important role for speed sensorless estimation.

In the case of a static eccentricity, the non-uniformity of the air gap produces asymmetric radial force where the position of the minimum radial air-gap length is fixed in space as illustrated in Figure 3.1.b [32][37][39]. This type of eccentricity is generated by an oval stator or the incorrect positioning of the rotor or the stator during manufacturing stage [32][37][39]. Static eccentricity is likely to cause dynamic eccentricity [32].

In addition, dynamic eccentricity might happen because of a bent shaft, misalignment of bearing, mechanical resonances, and bearing wear [32][37][39]. The position of minimum radial length of the air gap will rotate with the rotor as can be seen in Figure c, which illustrates that a dynamic eccentricity is a function of space and time. Static and dynamic eccentricity might be combined together, and this can be called mixed eccentricity as a third type.

All these types of eccentricity create asymmetric magnetic fields due to the non uniform air gap. Therefore, the certain spatial position resembles a sinusoidal signal that is related to the rotational speed. Therefore, by doing FFT the spatial signal will be represented as a saliency harmonic. Static and dynamic eccentricity can be found by using the following equation [2][10] [19][26][29][33]:

$$f_{ecc-dynamic} = [(KR \pm n_d) \left(\frac{1-s}{p} \right) \pm n_w] f_e \quad (3.1)$$

$$f_{ecc-static} = [(KR) \left(\frac{1-s}{p} \right) \pm n_w] f_e \quad (3.2)$$

Where:

$n_d = 0$: in case of static eccentricity
$n_d = 1, 2, 3 \dots$: in case of dynamic eccentricity
f	: fundamental supply frequency
n_w	: is the stator MMF harmonic order
p	: number of fundamental pole pairs
s	: slip
K	: any positive integer
R	: the number of rotor slots

$$f_{mix-ecc} = f_s \pm f_r \quad (\text{mixed eccentricity}) \quad (3.3)$$

The rotor speed equation can be derived from (3.2) where $n_d = 0$ and $n_w = 1$, as shown below. Thus, the rotor speed can be found only if the eccentric frequency (f_{ecc}) has been extracted and the number of the rotor slots is known. Consequently, Equation 3.9 is used in [26] to estimate the rotor speed considering that $k=1$.

$$f_{ecc} = [(KR)\left(\frac{1-s}{p}\right) \pm 1] f_e \quad (3.4)$$

$$f_{ecc} = (KR) \frac{(1-s)f_e}{p} \pm f_e \quad (3.5)$$

$$f_{ecc} \pm f_e = (KR) \frac{(1-s)f_e}{p} \quad (3.6)$$

$$f_{ecc} \pm f_e = (KR) \frac{f_r}{p} \quad (3.7)$$

$$f_r = \frac{p}{KR} (f_{ecc} \pm f_e) \quad (3.8)$$

$$N_r = \frac{60}{KR} (f_{ecc} \pm f_e) \quad (3.9)$$

$$R = \frac{60}{KN_r} (f_{ecc} \pm f_e) \quad (3.10)$$

The speed-dependent harmonic components of WRIM in the steady state can appear at frequencies given by [31][35][36][40].

$$f_h = | 6K(1-s) \pm 1 | f_e \quad (3.11)$$

There are unbalanced-supply and winding-operation conditions for WRIM, so the recent equation might not work for unbalanced operations [35]. However, one of the assumptions of the proposed method is that the motor is balanced and healthy and the source is balanced. Therefore, Equation 3.11 is going to be used in Chapter 4 for calculating the width of the windows. As all the previous equations depend on the slip, they inevitably rely on the rotor speed. Eventually, Equations 3.1 is the essential form that can be also derived to find rotor slot harmonics (RSH).

3.2.2 Slot Harmonics

Slot harmonics are the second type of the harmonics component that can be seen in the stator current or voltage spectrum of ACIMs. These types of harmonics are not really defects. They are built into the structure of the machine as they are caused by the need for conductors to pass through the machine, see Figure 3.2. Slot harmonics have been intensively studied because they have the ability to provide speed information in induction mechanics. For instance, the sensorless speed in [5] was measured from the difference Δf between the 5th harmonic and the rotor slot harmonic. Essentially, slot harmonics can be subdivided into: rotor slot harmonics and stator slot harmonics.

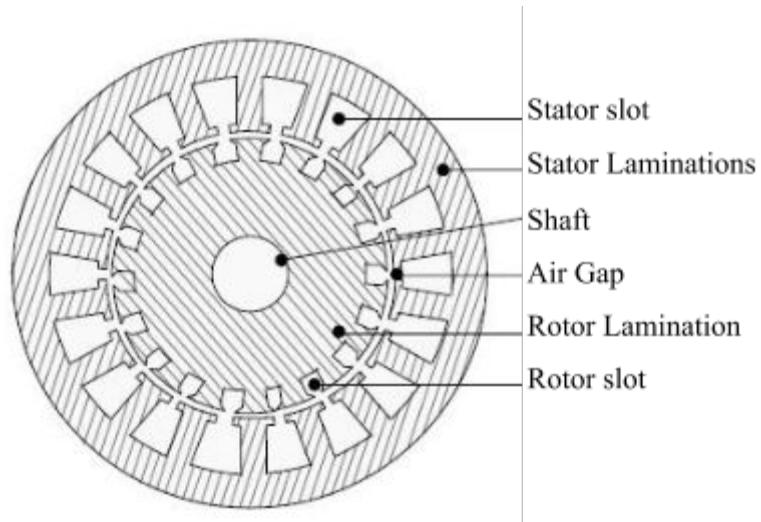


Figure 3.2: Stator-rotor slot in an IM.

The stator and rotor parts comprise a specific number of slots as illustrated in Figure 3.2. The number of stator and the rotor slots might vary from each other and they are commonly chosen based on certain conventions to achieve desired motor performance. Interestingly, these

slots can create detectable harmonics, so the effect of these slots has been widely studied in [2] [5][8][19][26][33][38]. In addition, the most important thing in these studies is that this type of harmonics can be found by the same equations of air-gap eccentricity, Equations 3.1, 3.2 and 3.4. The left side of the odd harmonics (3rd, 5th, 7th etc.) are seen to have saliency harmonics more than in the right side. Therefore, the window of the tracking algorithm will be placed in the left side of the odd harmonics.

Rotor slot harmonics (RSH) are usually found in most SCIM, so the expiration of RSH is commonly used for the SCIM saliency harmonics. However, the essential issue is extracting a group of speed-dependent harmonics from the stator current spectrum, whether these are eccentricity or slot harmonics. Therefore, "saliency harmonics" is the only expression that will be used for both types in this paper.

Saliency harmonics are proved to be mostly related to the machine pole-pair number and the operating speed [36]. Thus, most of the saliency harmonics studies require knowing the number of the rotor slots and other parameters such as stator and rotor resistor. Besides, researchers just consider that the saliency harmonic is the highest peak while it might be the second or third peak which is close to the odd harmonics. Even though a steady-state operation is considered for the proposed method, there are some cases in which the saliency harmonics might not be high enough to be extracted as the first peak. Also, because of the requirements that must be known to be able to estimate the rotor speed, ANN seems to be an outstanding solution. ANN can not only compute all the unknown parameters but also learns which of the peaks are the speed-dependent harmonics. Therefore, the sensorless speed estimation is based on a set of stator current harmonics that hopefully contain the saliency harmonics.

3.3 Fourier Fast Transform (FFT)

Converting the stator current signal from time domain to frequency domain is a mandatory step in order to detect the saliency harmonics. Fast Fourier transform (FFT) is the proper algorithm that has been successfully used in [5][29][31] [33][38][40] to transfer the stator current signal from time domain to frequency domain regardless of the utilization of the Goertzel algorithm in [2][30]. FFT provides an opportunity to see any signal as a function of frequency. Therefore, FFT algorithm as seen below, in Equation 3.12, is going to be used by MATLAB to convert the stator current signal to frequency domain.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}} \quad 0 \leq k \leq N - 1 \quad (3.12)$$

3.4 Tracking Algorithms

Various techniques are used to track saliency harmonics such as an adaptive tracking filter [14] and saliency tracking observer [27][28][30]. All the past studies were concerned with finding one or two saliency harmonics that might be used to calculate the rotor speed by using Equations 3.9 and 3.11. However, in this paper, the saliency harmonics will be demodulated by creating a particular tracking algorithm. The purpose of that is to select a group of harmonics as an input for ANN. The tracking algorithm will be set in MATLAB to extract two peaks at least from each window that is placed on the left side of the odd harmonics. This subject is going to be discussed more in Chapter 5.

3.5 Using ANN for Sensorless Speed Estimation

ANN is a well-known approach to predict any desired output for linear and nonlinear systems, so it has found wide application in the field of sensorless speed estimation [1][13] [16-18]. Therefore, some mathematical methods of sensorless speed estimation were combined with artificial intelligence in order to enhance the performance of the sensorless estimation [13] [16-18]. ANN approach is the main subject of this paper and it is going to be discussed in more detail in Chapters 4 and 5.

The previous studies reveal that the saliency harmonics successfully provide the rotor speed value without rotational sensor. The problem is most of these harmonics have small magnitude, so they are seen to be difficult to extract in some cases without very sensitive monitoring tools. Therefore, the lab experiment requires high efficiency tools; if such tools cannot be used, it might negatively affect the final result of ANN. The most difficult case is that the saliency harmonics are at the same level as the noise floor, which is the minimum level of a signal, or they are not the first or second peak in the windows. However, ANNs might be able to overcome this difficulty.

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS

4.1 Introduction

In a general overview, the principle of artificial neural networks (ANNs) is simply taken from the basic function of a biological neuron. In information technology, this approach is called ANN. Thus, it will be always referred as ANN in this paper. ANN approach will be illustrated in this chapter by giving definitions, some applications, the design process, and an example in order to give a brief introduction of ANNs. Substantially, ANNs offer a revolutionary new approach to science in terms of fast learning and accurate results.

A neural network is, in essence, one of the most important approaches in artificial intelligence that approximately imitate the way a human brain operates. Neural network theory is based on the principle of how biological neurons work, so ANNs are designed to simulate certain key properties of biological neurons. As seen in Figure 4.1, the three main parts of a biological neuron are dendrites, a cell's body, and an axon. These parts work as a mathematical function that can be implemented in an artificial model. Therefore, artificial neurons are designed to mimic aspects of their biological counterparts as shown in Figure 4.2.

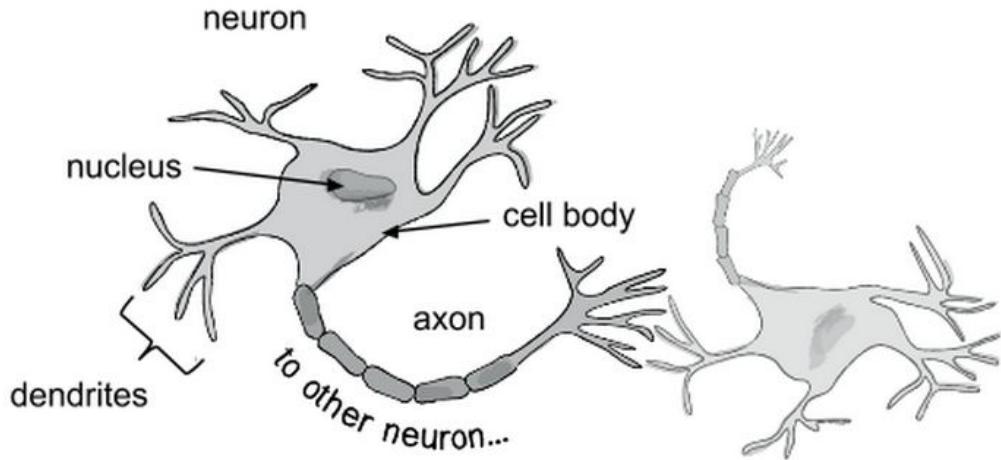


Figure 4.2:: Components of a biological neuron.

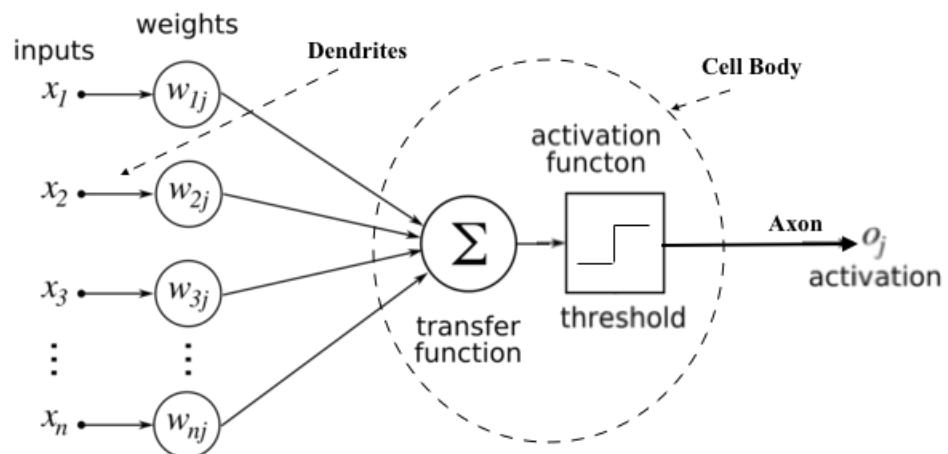


Figure 4.2: The neuron model.

In mathematical implementation, the dendrites allow the cell to receive signals, so it acts as the input vector. The cell's body acts as the summation function then the activation function. There are several types of activation functions in use with ANNs. The most common choice of activation functions are linear, hyperbolic tangent, or threshold function that are used in Figure 4.2. The threshold function works as a step function that gives an output equal 1 when the input receives a signal of more than zero [41]. For example, if the input is equal -1, the output will be equal 0, and if the input is equal 0.3, the output will be equal 1 and so on. The axon gets its value from the output of the cell's body that is either 0 or 1 based on the activation function [42].

In other words, mathematically, each input in a vector will be multiplied by its weight, and then the results will be added to each other. After that, the summation will pass through the activation function that will produce either 0 or 1 if the activation function is a threshold function. The most important tool in ANNs is the weight that can be adjusted by using a certain algorithm in order to get the desired output. Algorithms allow ANNs to learn from the errors, so it is called a learning algorithm, which will be explained later.

4.2 The Principle of Neuron Weights

The ANNs' mission is to achieve the desired output for each input signal by using a learning algorithm that is applied to adjust the weights of ANNs. To illustrate how adjusted weights can significantly change the output value, a simple example is explained. Two patterns for assumed data are shown below in Table 4.1. The input vector and desired output are given.

Table 4.1

One neuron input, weights, and output with threshold function for two inputs

pattern	input vector	desired output	weights (n=1)	output (n=1)	weights (n=2)	output (n=2)
1	[1 3]	0	[-2 1] ^T	1	[1 -1] ^T	0
2	[2 1]	1		0		1

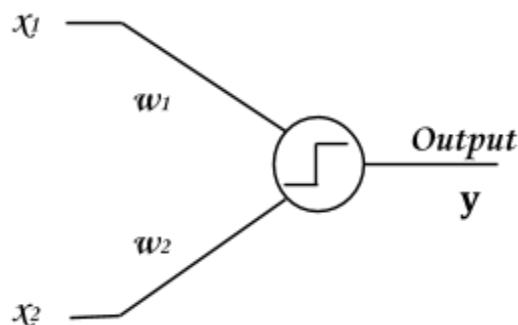


Figure 4.3: One neuron structure with threshold function for two inputs.

First of all, the activation function must be chosen, and the initial value of the weights must be set. By noticing that the desired output value is either 1 or 0, the activation function can be the threshold. This task can be solved by just one neuron, as shown in Figure 4.3, but, first, the initial values of the weights, w_1 & w_2 , will be set arbitrarily at n=1, [-2 1]. The artificial neurons' output can be found by using Equations 4.1 and 4.2. For pattern 1, the output will be = $(1*-2)+(3*1) = 1$ that is more than 0, so the output of the threshold function will equal one, which is not the desired output. But for pattern 2, the output will be = $(2*-2)+(1*1) = -3$ that is less than 0, so the output of the threshold function will equal 0, which is not the desired output. Since the desired output is not achieved at n=1, the system will begin to learn from the error.

Now, the weights are the only components that can be changed to achieve the desired output $[0 1]^T$. In addition, the error will be the main factor that is used in the learning algorithm to adjust the weights. But for just this example, the weights will be set again arbitrarily, and the learning algorithm will be discussed later. The new assumed weights are $[1 -1]$, so the output of pattern 1&2 is $[-2 1]^T$. Thus, the output of the threshold function equals $[0 1]^T$, which is the desired output.

$$V_n = \begin{bmatrix} x_{p1} & x_{p2} \\ x_{p1} & x_{p1} \end{bmatrix} [w_1 \ w_2]^T \quad (4.1)$$

$$y_n = \text{thrsh}([V_1 \ V_2]^T) \quad (4.2)$$

at n=1

$$V_1 = \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} [-2 \ 1]^T = [1 -3]^T$$

$$y_1 = \text{thrsh}([1 \ -3]^T) = [1 \ 0]^T$$

at n=2

$$V_2 = \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} [1 \ -1]^T = [-2 \ 1]^T$$

$$y_2 = \text{thrsh}([-2 \ 1]^T) = [0 \ 1]^T$$

The previous technique is called training process where the weights are adjusted because the desired output is not achieved. The training process is the significant part of ANN; it is going

to be discussed more in the next section. In the recent example, the weights were arbitrarily adjusted to clarify the principle of weights in ANN where they must be adjusted by using a learning algorithm. There are several learning algorithms that are used to modify the weights in ANN. One of the most widely used training algorithms is called backpropagation; that is the only algorithm considered in this paper.

4.3 Why ANNs?

The single neuron that has been explained previously can solve a highly complicated function; if it is connected with other neurons as a network, that is called ANNs [41]. ANNs consist of layers in series, and each layer has either many neurons in parallel or just one neuron as shown in Figure 4.4. This structure is the first simple model of ANNs that was built by Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, in 1943. Since then, ANNs are increasingly popular as a new revolutionary approach. Many scientists have been developing this approach in order to design a system that can learn to make the right decision much like a human does [42].

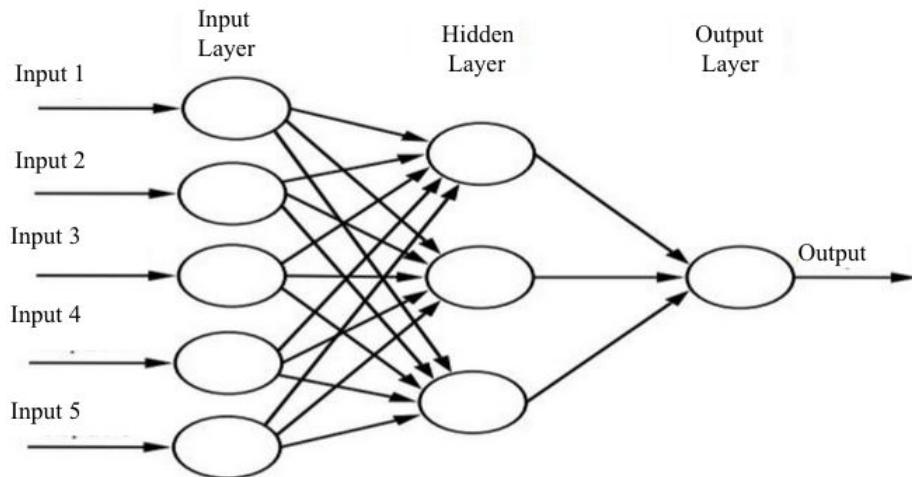


Figure 4.4: A net of neurons structure (ANN).

ANNs are trained to learn from existing data. For example, during the training, ANNs can learn to distinguish between images, voices, or shapes based on the data. Thus, after training, ANNs can be functionally used to make a decision on fingerprint recognition, speech recognition, character recognition, and so on.

ANNs have been applied in many different systems in order to make them more intelligent. In our daily lives, most phones have a system that can recognize speech and transfer it to a text like Siri. Other apps are able to turn images of documents and book pages into editable text. In the industrial field, many robotics are used instead of humans because of their speed, reliability, and efficiency. These robots are becoming smarter than in the previous decade. The reason is that robots are using an artificial intelligence approach like ANN's that can learn and make decisions accurately based on how they are trained. Therefore, ANNs' ability would encourage researchers to apply it to ACIM applications.

The previous examples rely on either a classification or predictive technique, which are the most important types of ANNs. Interestingly, the predictive technique is the nominated type of sensorless speed estimation of ACIMs. This type is also used in the prediction of energy consumption, weather forecasting, and stock market outlook. Even though the classification and the predictive technique are different, they share the same concept and most procedures. They are not capable of learning from nothing, which means ANN, in both types, must be trained from existing data before it can be used in any application.

4.4 Designing ANNs

Now, the most important question should be asked: how to design an ANN and get a benefit of it? To make everything easy and clear, there are two stages. The first stage begins before starting to design; the essential part is the data that allow the designer to decide which of them will be the input and the output. Then, it can be known how many inputs and outputs exist. Depending on the expertise, the designer can also decide to choose the number of neurons, the activation functions, and the number of hidden layers considering that the input layer and the output layer are not hidden (see Figure 4.4). The optimal structure consists of one or two hidden layers. Each layer has some neurons, at least one. These neurons work as the activation function that designers choose. Therefore, in this paper, the hyperbolic tangent activation function is the selected type for all the activation functions except the neuron of the output layer, which is a linear activation function. The graphs of these functions are illustrated in Figure 4.5 and 4.6.

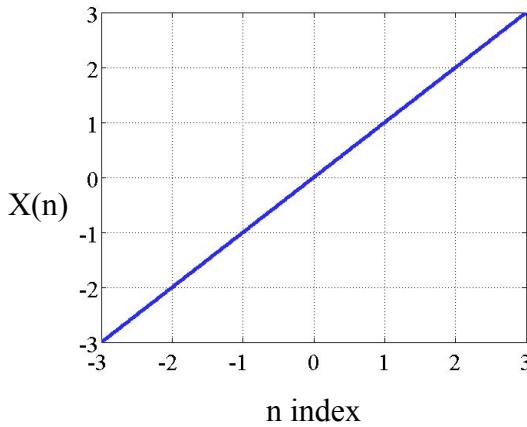


Figure 4.5: A linear activation

function.

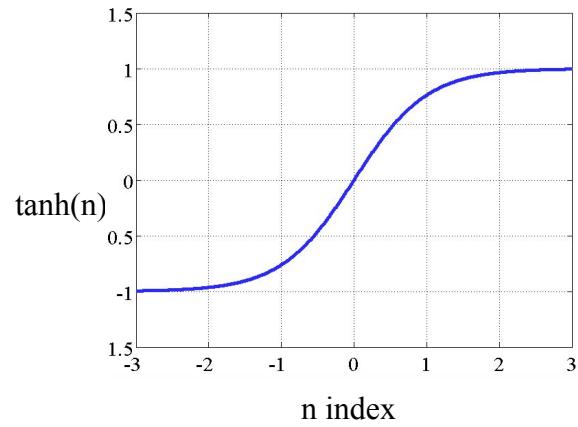


Figure 4.6: The hyperbolic tangent

activation function.

The last step before beginning to design is dividing the ANN data into 70% training data and 30% validation and testing data. Simply, the training data (70%) is used to update the weight, and the validation data (15%) is set during the training process to check the accuracy and avoid over-fitting. After the training process, the testing data is also used to check the ANN accuracy. These steps are used to determine how much accuracy ANNs offer, so the designer must verify if ANNs give accurate outputs as much as possible; otherwise, the design will be worthless.

In other words, the second stage, after beginning to design, is training ANN using the 70% of the data until the wanted accuracy is achieved. After the end of the training process, the eventual weights of the ANN will be ready to be examined by the testing data. If the ANN that was built gives a good accuracy for the desired output of the testing data, the design can be successfully used for implementation. It must be mentioned that the ANN inputs are going to be in a vector, $X(n)$, that is related to the extracted peaks; their relationship will be explained in

Chapter 5. Thus, the ANN data of the proposed method contains $X(n)$ as the input and the rotor speed value as the desired output. Also, the designing of ANNs for sensorless speed estimation is going to be extended and discussed more after the collected data analysis.

CHAPTER 5

METHODOLOGY

5.1 Introduction

After explaining the theoretical knowledge of saliency harmonics in Chapter 3, the experimental work is preferred in order to evaluate the proposed method. This chapter will describe the experimental framework for this thesis. The method of estimating the rotor speed will be addressed in detail.

5.2 Procedures

The overall structure of ANN sensorless speed estimation is depicted in the Figure 5.1. Since this method relies on an ANN, the sensorless speed detection cannot be done if ANN is not trained accurately. Therefore, the method has two stages; the first one is the training process that depends on 70% of the collected data of the stator current. After the ANN has been validated, the system can move to the second stage, the testing process, which is measuring the rotor speed without tachometer. The last stage is just verifying the final solution in order to confirm the actual predictive power of the ANN. Both stages have four steps, which are stator current monitoring, fast Fourier transform (FFT), rotor saliency tracking algorithm, and an artificial neural network (ANN) as shown in Figure 5.1. The slight difference between these stages is the training stage, requiring a speed sensor as illustrated in Figure 5.1, while the implementation of the final project will be the same structure but without the speed sensor.

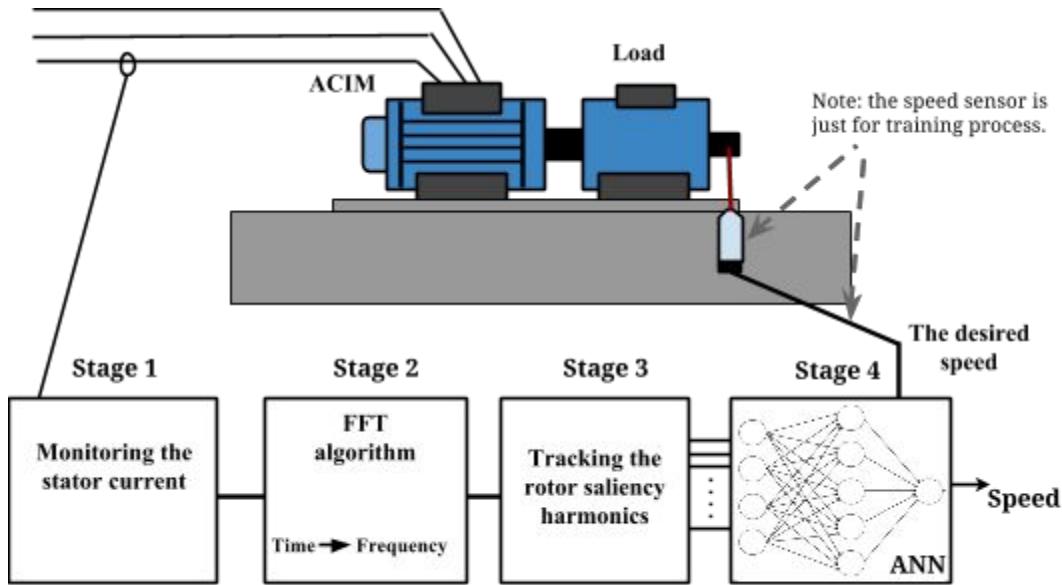


Figure 5.1: The overall structure of ANN sensorless speed estimation.

The proposed method has to follow specific tasks in order to reach the final results of the ANN accuracy. These tasks are as the following:

1. Collecting data of the stator current signals
2. Transferring the data from time domain to frequency domain
3. Analysing the transfer data and applying saliency tracking algorithm
4. Selecting the tracked saliency harmonics as the the input of ANN
5. Designing the artificial neural network (ANN)
6. Training and testing ANN

Each stage of the proposed method might contain one or two tasks, and these stages will be thoroughly discussed in much more detail.

5.2.1 Stage 1: Monitoring the Stator Current

The main goal of this stage is to monitor the stator current signal for a period of time (T_m) and send it as a digital signal to the FFT stage. The stator current data are obtained from real IMs. Three different IMs are chosen to examine the proposed method. The first and second motors are wound rotor induction motors (WRIMs), while the third is a squirrel cage induction motor (SCIM). Many components are used in the lab in order to acquire the stator current data, and these components are: a clamp current measurement, an oscilloscope, a tachometer, and a personal computer (PC).

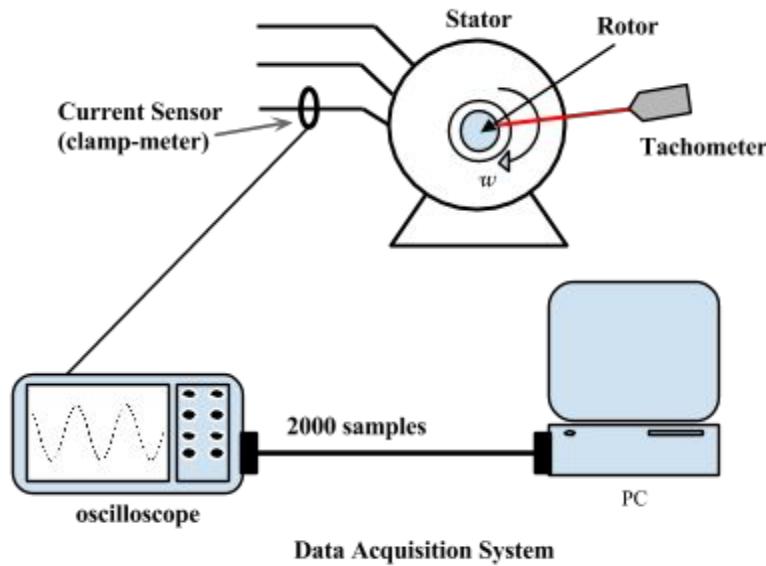


Figure 5.2: The general structure of the lab experimental setup.

The stator current was monitored by a clamp current connected to an oscilloscope that shows the signal of the current. The stator current signal is digitized and displayed on an oscilloscope as illustrated in Figure 5.2. Then, the signal is saved with the rotor speed that is measured simultaneously. This step is repeated many times in order to collect stator current signals at various speeds. These signals are sent in a digital form of 2000 samples for each speed that is measured by the tachometer. It should be mentioned that 2000 is the maximum number of samples that can be extracted from each signal. The following descriptions in Table 5.1 are for the three motors and the coupled DC motor (load).

Table 5.1

The three motors descriptions

Motor A & B	Motor C
Wound Rotor Voltage 208 AMPS 1.2 PH 3 H.P 0.25 RPM 1750	Squirrel Cage Voltage 208 AMPS 1.7 PH 3 H.P 1/3 R.P.M. 1725

DC motor (load)
Voltage 125 Amp 3 SH. FLD. AMP. 0.5 HP 0.33 RPM 1800

The WRIMs and SCIM (A,B, & C) are investigated separately in an experiment to obtain their stator current data several times with different speeds. In the experiment of the WRIMs, A & B, the stator windings are connected in a delta, and the rotor windings are connected short as shown in Figure 5.3. On the other hand, the stationary part of SCIM (C) is connected in a star as illustrated in Figure 5.4. The armature of the DC motor that is coupled with each motor (A, B, & C) is connected with a 25 ohm resistor in parallel, while the shunt field is connected with a variable voltage source (0-125 V.DC - 6 A). The variable voltage source is used to change the load torque which in turn changes the rotor speed. The rotor speed and the stator current data is measured simultaneously. The motors are supplied by a constant voltage of 208V and a constant frequency of 60Hz.

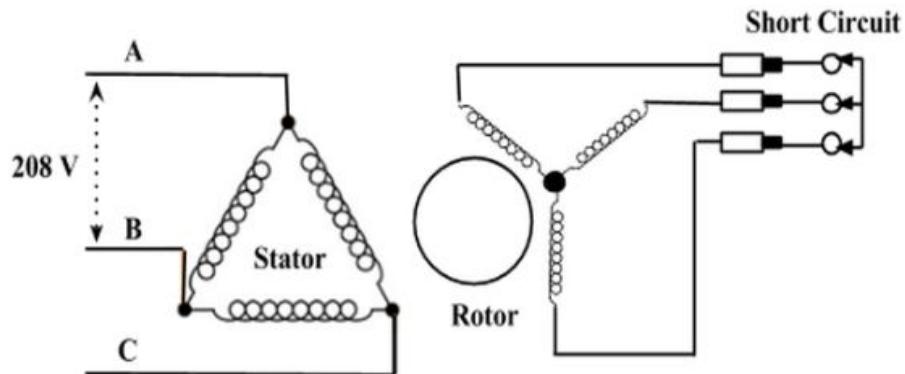


Figure 5.3: A & B (WRIMs) connections.

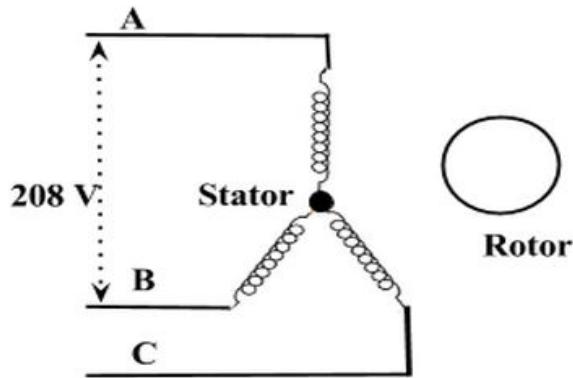


Figure 5.4: C (SCIM) connections.

5.2.2 Stage 2: Transfer the Data from Time Domain to Frequency Domain

As soon as the signal is received, it must be converted to a frequency domain. Therefore, some signal processing techniques are required for the proposed method. Thus, the primary purpose of this stage is to transfer the stator current data from time domain to frequency domain by utilizing FFT. The reason to convert is to be able to extract the most important frequency components of the stator current. MATLAB is used for all the signal digital analysing in this paper. There are some factors have to be mentioned in this chapter, and they are:

1. acquisition time (T_m)
2. FFT resolution
3. window widths
4. the number of windows (K)
5. the position of the windows
6. the number of peaks for each window (U)
7. the number of neurons in the hidden layer

Table 5.2 shows the general description of the collected data where each pattern has one second as the acquisition time except data D's patterns. A set of three patterns of each data were usually measured at the same speed, which would be very helpful in the training process.

Table 5.2

The collected data

Data types	Motor type	Number of patterns	Tm (second)	Speed range (RPM)
A	A	30	1	1784 to 1707
B	B	10	1	1788 to 1694
C	C	30	1	1791 to 1735
D	A	67	0.5	1783 to 1694

All data for all three motors have 2000 samples, which is the maximum amount of samples that can be taken from the oscilloscope to the computer. This small number of samples will limit the ability to extract higher saliency harmonics above 1000 Hz. The calculations below demonstrate the reason behind the limitation of the Nyquist frequency for all data except D.

$$F_s = \frac{N}{T_m}$$

$$F_s = \frac{2000 \text{ samples}}{1 \text{ second}} = 2000 \text{ Hz}$$

$$\text{Nyquist frequency } F_n = \frac{F_s}{2}$$

$$F_n = \frac{2000}{2} = 1000 \text{ Hz}$$

The number of samples (N) that are available is 2000 with a sampling frequency (F_s). If the acquisition time (T_m) of this data is known, the sampling time (T_s) can be calculated below with the FFT resolution.

$$T_s = \frac{T_m}{N} = \frac{1}{2000} \text{ or } T_s = \frac{1}{F_s} = \frac{1}{2000} = 0.5ms$$

$$\text{FFT resolution} = \frac{1}{T_m} = \frac{1}{1} = 1Hz \text{ or}$$

$$\text{FFT resolution} = \frac{F_s}{N} = \frac{2000}{2000} = 1Hz$$

Because the Nyquist frequency equals 1000 Hz in data A, B, and C, the highest odd harmonic of 60Hz is the 15th harmonic. Thus, the windows of the tracking algorithm that are going to be placed on the left side for each odd harmonic are from the 3rd to 15th harmonics. On the other hand, the Nyquist frequency of data D is 2000 Hz and the FFT resolution is 2 Hz, because T_m equals a half second. The main purpose of collecting data D is to decrease the time delay and see if the accuracy of ANN will be the same as data A. The resolution can be increased to one Hertz by adding 4000 zeros to the data D to complete one second. The results will be demonstrated in Chapter 6.

5.2.3 Stage 3: Tracking the Rotor Saliency Harmonics

The tracking algorithm is significantly important to any sensorless speed method that depends on saliency harmonics. Creating a cohesive tracking algorithm is the main objective in

this section. The factors associated with this stage are window width (Lw), the number of windows, and their positions. As illustrated earlier in Chapter 2, in steady state, the saliency harmonics are likely to be located slightly less than the odd-order harmonics (3rd, 5th, 7th, 9th, etc.). In addition, the speed for all motors was measured at a maximum of 1791 RPM and a minimum of 1694 RPM. Therefore, the range varies between 0.5% to 5.9% of the slip value as calculated below.

$$S_L = \frac{1800 - 1791}{1800} = 5m \quad S_H = \frac{1800 - 1694}{1800} = 59m$$

The slip range is considered to be larger than 5.9% in case saliency harmonics are slightly out of this range. Therefore, 8 % ($> 5.9\%$) is chosen to be S_H . The window width depends on the slip range (S_H), fundamental frequency (f_1) and the value of K, which is a positive integer. The width of each window can be obtained from Equation 2.11 (from Chapter 2). Therefore, Table 5.3 shows how the widths are found. Then, the difference (D) between the odd harmonics and the calculated saliency can be approximated to be the final value of the widths as illustrated in Table 5.4.

$$f_h = |6K(1-s) \pm 1| f_e$$

For example:

at $s = 8\%$ the f_h for the first row is

$$f_h = |6(1-0.08) - 1| 60 = 271.2 Hz$$

$$\text{so, } D = 300 - 271.2 = 28.8 \text{ Hz}$$

Table 5.3

The calculation of the maximum saliency harmonics based on 8% slip.

K	± 1	f_h at s = 0 (Hz)	odd harmonics	f_h at s = 8% (Hz)	D
1	- 1	300	5th	271.2	28.8
1	+ 1	420	7th	391.2	28.8
2	- 1	660	11th	602.4	57.6
2	+ 1	780	13th	722.4	57.6
3	- 1	1020	17th	933.6	86.4

Table 5.4

The approximate width for each window

odd-order harmonic	1st	3rd	5th	7th	9th	11th	13th	15th	17th
width (Hz)	5	-	29	29	-	58	58	-	87
the average width (Hz)	-	17	-	-	44	-	-	73	-

It is noticed that the 3rd, 9th, and 15th harmonics can not be obtained from the previous equation. Thus, the width of their windows will be assumed to be the average. However, these widths can be set larger than the recommended values as designers prefer. The width probably will affect the accuracy of ANN.

After identifying the widths, the tracking algorithm can extract a specific number of peaks from each window. There will be a maximum of seven windows as selected in Table 5.4 (see the green shadow). These peaks can be represented in a vector, F(P), that is demonstrated below. This number of peaks will definitely influence the accuracy of ANN. Finally, it must be mentioned that the frequency values of the peaks are the required data for the ANN input, not the

magnitude of the peaks. After extracting certain peaks from each window (3rdW-15thW), it is very helpful to subtract these peaks from the odd harmonic of their windows, as shown in Equation 5.2. Therefore, the new vector is represented in Equation 5.3.

$$F(P) = [P_{1(1)} P_{1(2)} \dots P_{1(U)} P_{2(1)} P_{2(2)} \dots P_{2(U)} P_{K(1)} P_{K(2)} \dots P_{K(U)}] \quad (5.1)$$

$$P_{k(u)} \in \mathbb{R}$$

$$D_{k(u)} = ((2*k+1)*f_1) - P_{k(u)} \quad (5.2)$$

$$G(D) = [D_{1(1)} D_{1(2)} \dots D_{1(U)} D_{2(1)} D_{2(2)} \dots D_{2(U)} D_{K(1)} D_{K(2)} \dots D_{K(U)}] \quad (5.3)$$

$$D_{k(u)} \in \mathbb{R}$$

$$k = 1, 2, 3, \dots, K \quad u = 1, 2, 3, \dots, U$$

f_1 : the fundamental frequency

The last required step for ANN input is the normalization, especially for the data that have values bigger than one. The reason behind this step is to make the ANN produce a faster and more accurate prediction. The optimal normalization that will work very well with the hyperbolic tangent function is from -1 to 1. Thus, G(D) is going to be normalized to be the ANN input vector, X(n), as seen below.

$$X(n) = [X_1 X_2 X_3 \dots X_{N-1} X_N]$$

The input vector $X(n)$ is added to a bias ($X_{bias} = 1$), which is very helpful in many cases. The bias value has the ability to shift the activation function to the left or right, which might be critical for successful learning. The new input vector associated with the bias is given below.

$$X_{new} = [X(n) \ X_{bias}] = [X_1 X_2 X_3 \dots X_{N-1} X_N \ 1]$$

5.2.4 Stage 4: ANN Estimation

The major point that is used to evaluate the entire system is the accuracy of the ANN output. All of the six factors that were mentioned in 5.2.2 will also affect the accuracy. Therefore, the optimal design that has been stated earlier is going to be explained in this section, and then the last task, training and testing ANN, is going to be applied. In Chapter 4, the backpropagation algorithm was selected to be the learning algorithm in this paper. Also, the hyperbolic tangent, Equation 5.4, is the activation function for all the neurons except the neuron of the output layer that is a linear activation function, Equation 5.6. The derivation of these functions, Equations 5.5 and 5.7, are needed for the neuron error gradients as illustrated in backpropagation algorithm.

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad z \in \mathbb{R} \quad (5.4)$$

$$\frac{df(z)}{dz} = 1 - f(z)^2 \quad (5.5)$$

$$g(z) = z \quad z \in \mathbb{R} \quad (5.6)$$

$$\frac{dg(z)}{gz} = 1 \quad (5.7)$$

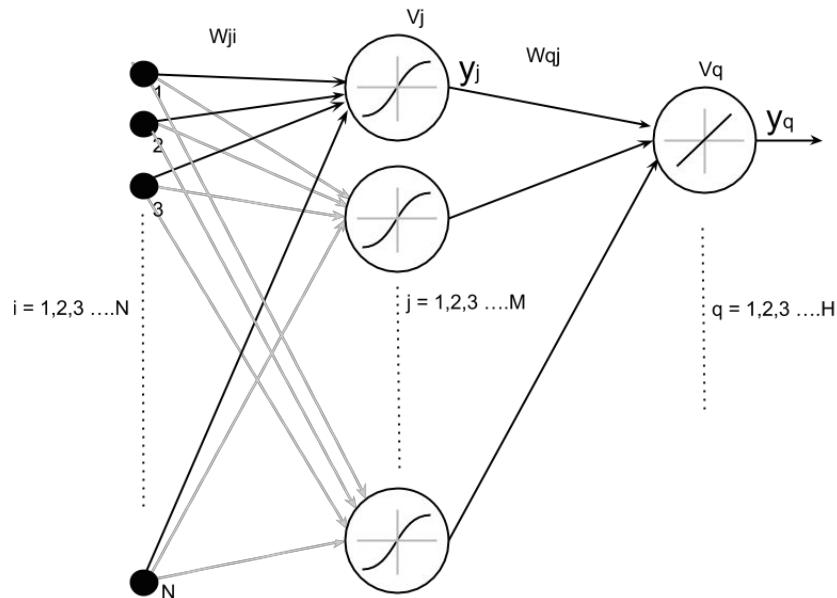


Figure 5.5: ANN structure.

Based on the ANN structure that is shown in Figure 5.5, the backpropagation algorithm is presented below.

The output of the hidden layer:

$$V_j = W_{ji} X^T$$

$$Y_j = \tanh(V_j)$$

The output of the output layer:

$$V_q = W_{qj} Y_j^T$$

$$Y_q = V_q$$

The difference between the desired output and the actual output:

$$Error = d_q - Y_q$$

The hidden layers' error gradients:

$$\delta_q = Error * (1)$$

$$\delta_j = (\sum \delta_q W_{qj}) * (1 - Y_j^2)$$

Updating weights:

$$W_{ji(k+1)} = W_{ji(k)} + u \delta_j X_i$$

$$W_{qj(k+1)} = W_{qj(k)} + u \delta_q Y_j$$

where:

x_i is the i th input.

Y_q is the actual output

d_q is the target output

Y_j is the first layer neuron's activation function

μ is a small constant called *learning rate*

It is time to apply the backpropagation algorithm in the training data to end up with final weights that are used to examine the ANN accuracy by the testing data. The accuracy is calculated by the following equation:

$$\text{the accuracy \%} = [100 - 100 * \frac{\text{ave. |Error|}}{\text{ave. desired speeds}}] \% \quad (5.8)$$

Eventually, the accuracy must be defined based on the available data. The worst case is that the ANN always gives a fixed number where the desired speed is varied. For example, the desired speed of data A varies from 1785 to 1707 RPM as listed in Table 5.5. By assuming the worst case, the ANN provides a specific number no matter what the desired speed is, so the ANN output might be one of the three cases as shown in Table 5.5. Therefore, by using Equation 5.8, the accuracy of each case is shown in the last row. Their accuracy seems to be good, but in reality, they are not. Thus, the acceptable accuracy is determined in Table 5.6, where the successful design must not go up 3 RPM as an average error.

The overall structure of all the steps that have been discussed in this chapter are shown below. Figure 5.6 concludes all the required phases, beginning from extracting each pattern, doing the FFT, and calculating the Nyquist and fundamental frequency. Then, Figure 5.6 shows the way to place each window in its position, and to obtain a certain number of peaks from each window. The arrangement of these peaks is critical as illustrated in the flowchart. Finally, ANN topology is represented as the last phase, where the rotor speed can be finally predicted after the training process, considering the red structure in Figure 5.6 is used for training.

Table 5.5

The worst accuracy

Desired Speed	ANN output case 1	ANN output case 2	ANN output case 3
1784	1784	1707	1745.5
1784	1784	1707	1745.5
1774	1784	1707	1745.5
1774	1784	1707	1745.5
1765	1784	1707	1745.5
1765	1784	1707	1745.5
1759	1784	1707	1745.5
1759	1784	1707	1745.5
1751	1784	1707	1745.5
1751	1784	1707	1745.5
1742	1784	1707	1745.5
1742	1784	1707	1745.5
1742	1784	1707	1745.5
1734	1784	1707	1745.5
1734	1784	1707	1745.5
1727	1784	1707	1745.5
1727	1784	1707	1745.5
1715	1784	1707	1745.5
1715	1784	1707	1745.5
1715	1784	1707	1745.5
1707	1784	1707	1745.5
1707	1784	1707	1745.5
Diff.	39.8	37.2	20.5
Accuracy	97.7	97.7	98.83

Table 5.6

Accuracy scales

Ave. E	Accuracy %	scales
1	99.94	Excellent
2	99.88	very good
3	99.82	good
4	99.77	average
5	99.71	poor
6	99.65	very poor

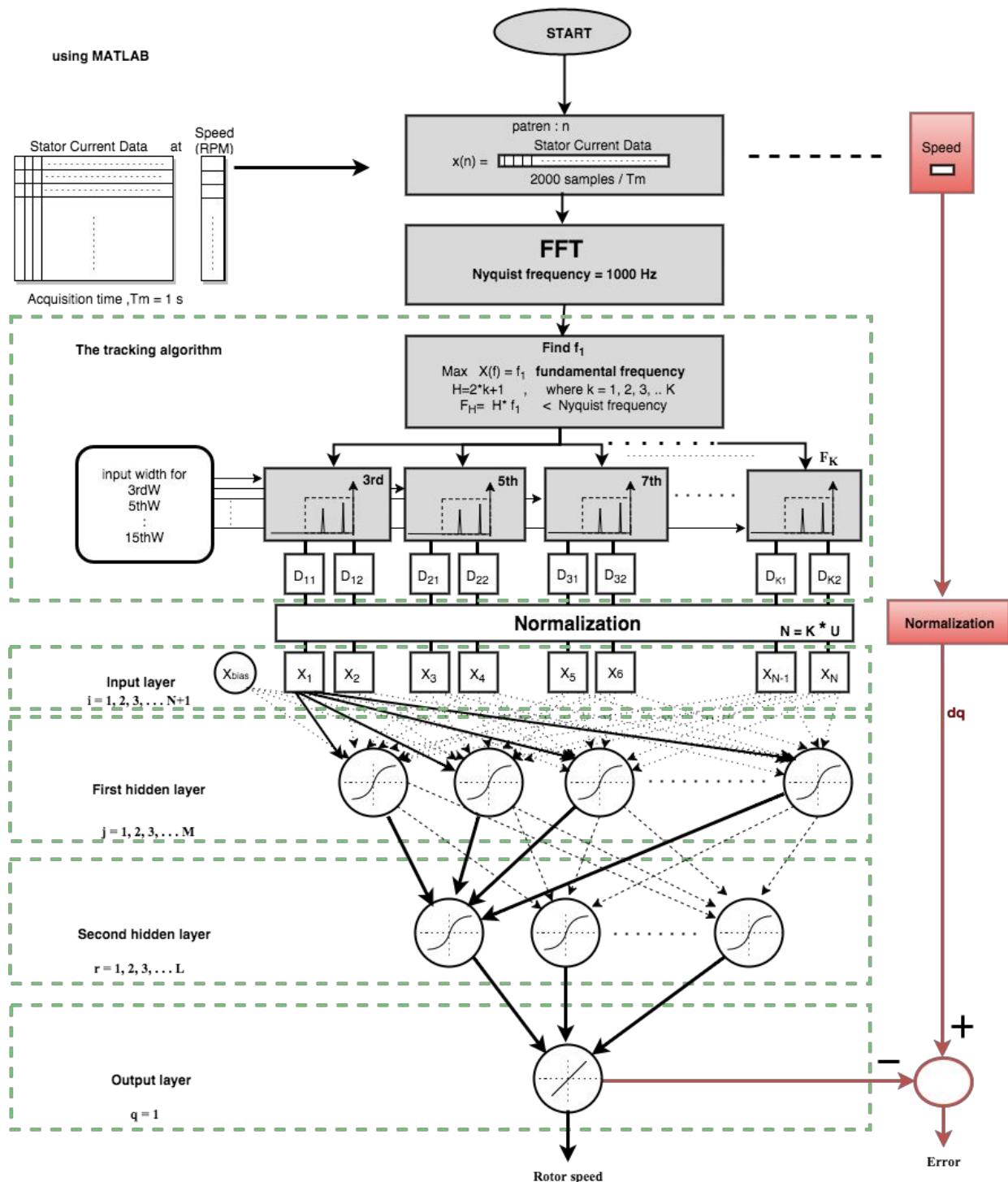


Figure 5.6: The flowchart of the proposed method.

CHAPTER 6

RESULTS

6.1 Extracting Harmonics

Previously, all the steps of sensorless speed estimation of ACIMs have been clearly explained in Chapter 5. The flowchart of the proposed method depicts these steps serially. Thus, in this chapter, the primary aim is to evaluate the accuracy of the ANN prediction for each data type.

All the ACIMs data that were collected from the lab were investigated by using MATLAB, and all the codes that are required to apply the proposed method are attached in the Appendices. First of all, there are four types of the collected data from different motors; these data are A, B, C, and D data depending on the type of ACIMs that were mentioned in Table 5.2.

In this paper, the ANNs predictions were divided based on the collected data into data A, A+B, C, and D prediction. All the collected data were measured in time domain, and each pattern had 2000 samples for one second except each pattern of data D had 4000 samples. The next example is presented to illustrate the idea of extracting the peaks of each pattern. Figure 6.1 shows just a quarter of the stator current signal where it was monitored at speed 1747 RPM, pattern 24 in Table 6.3.

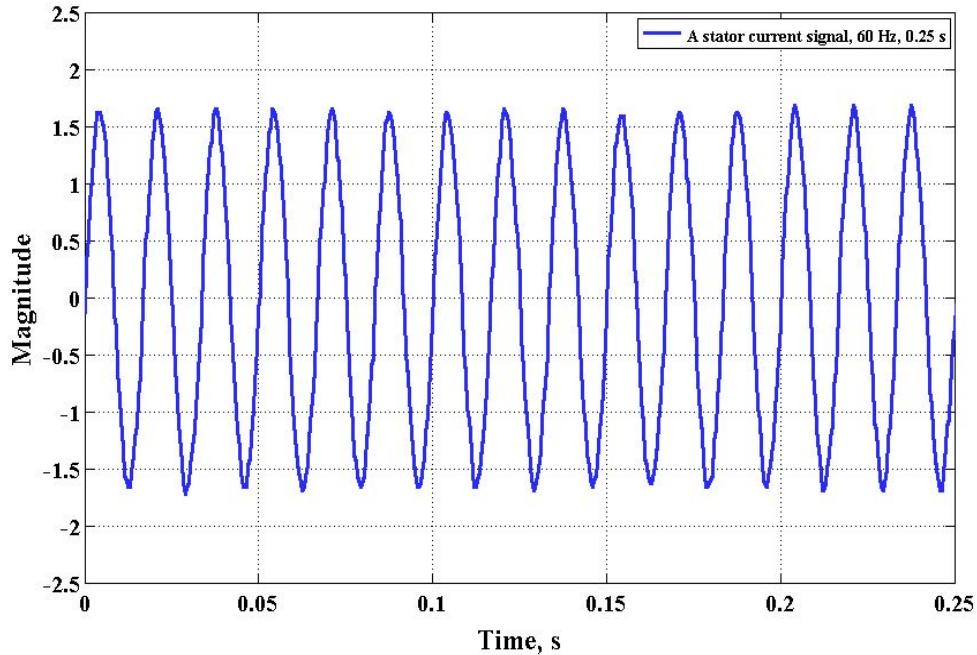


Figure 6.1: A quarter of the stator current signal at 1747 RPM, pattern 24.

The next stage was to convert the stator current signal to frequency domain by using the FFT algorithm. Figure 6.2 depicts the stator current spectrum of pattern 24. The goal was to extract the saliency harmonics by extracting a number of peaks from each pattern. The assumption is that saliency harmonics are in the left side of the range for each odd-order harmonics. In order to demonstrate how the tracking algorithm is applied, the FFT of the previous example is shown with a certain number of windows (K) that were implemented to extract a specific number of peaks (U), as shown in Figure 6.3.

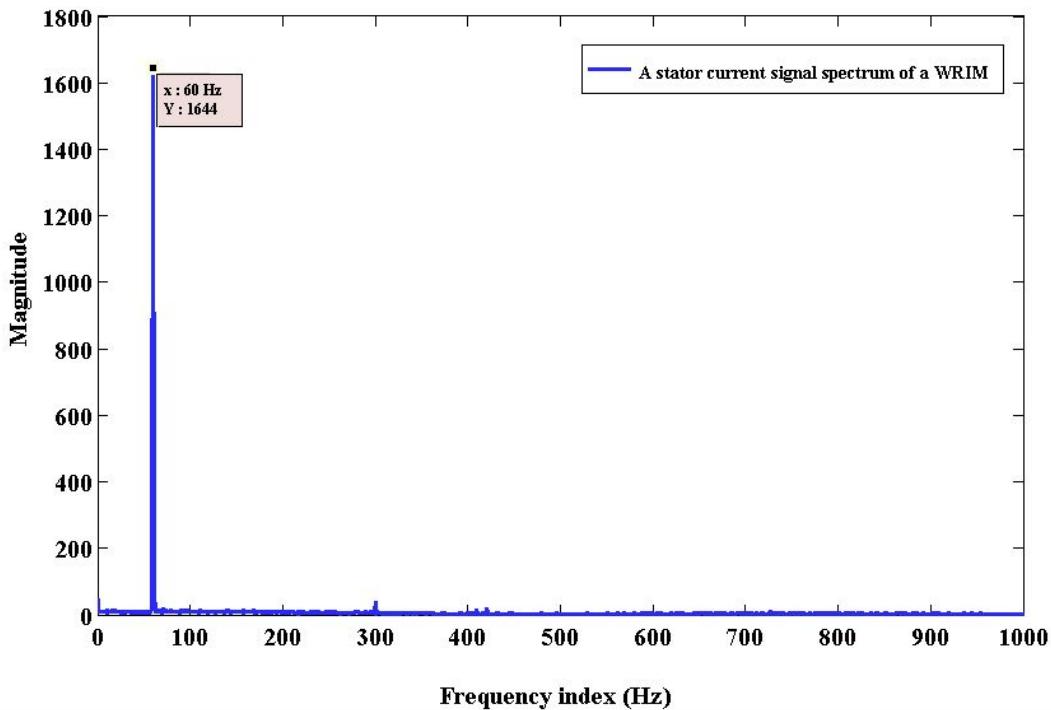


Figure 6.2: The stator current spectrum at 1747 RPM, pattern 24 in Table 6.3.

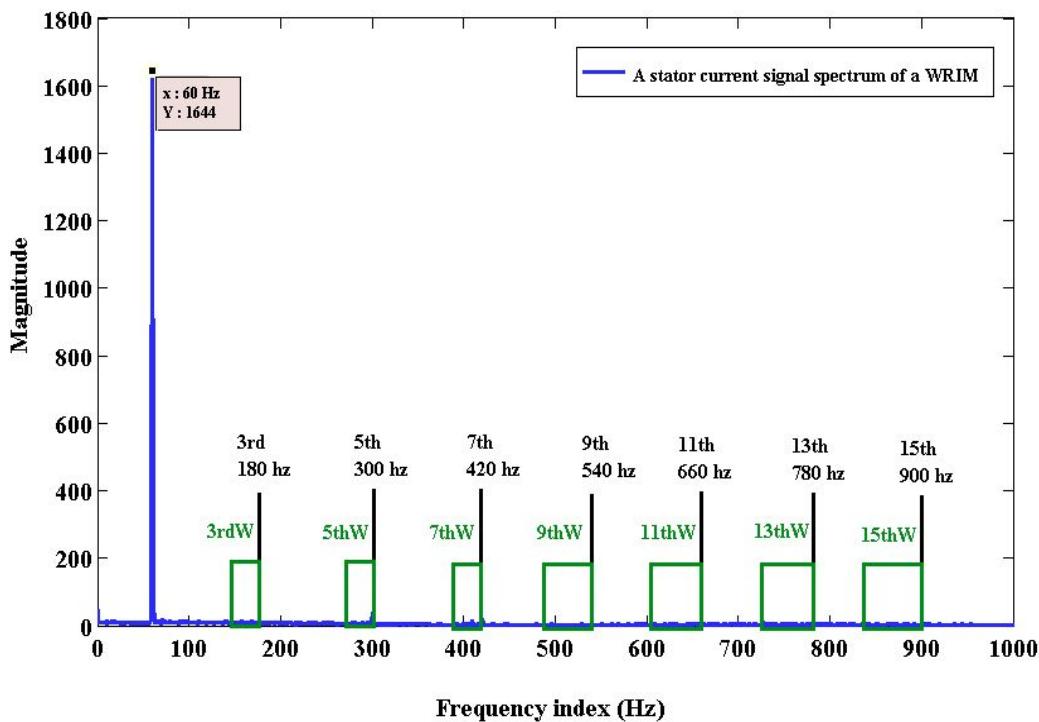


Figure 6.3: The windows of the tracking algorithm.

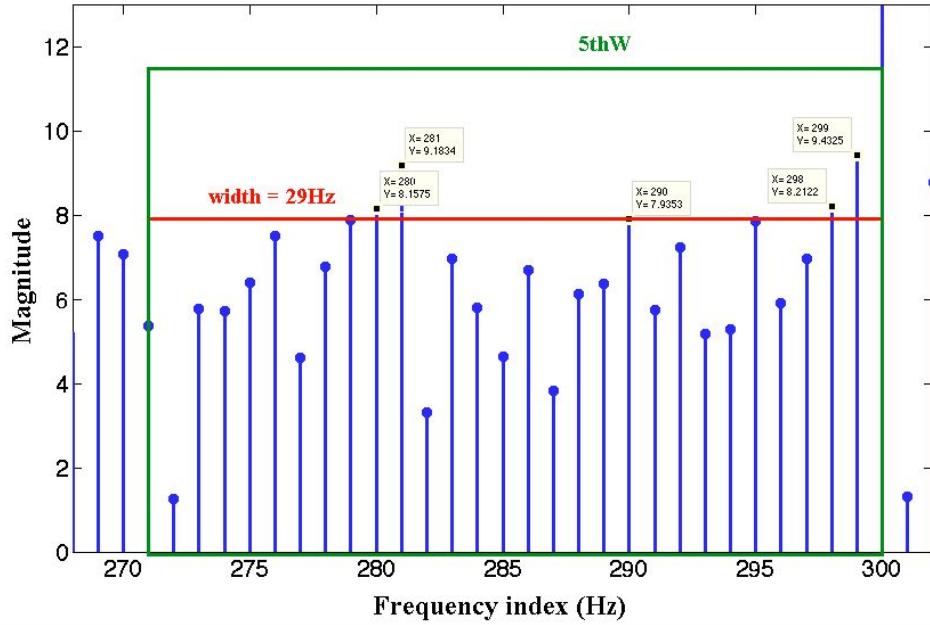


Figure 6.4: The 5th window of the tracking algorithm on pattern 24 in Table 6.3.

Each window extracts U number of peaks. For instance, the second window (5thW) of the previous example is shown in Figure 6.4, and the first five extracted peaks ($P_{K(U)}$) of this window are listed in Table 6.1, where $K = 2$ and $U = 5$. Therefore, Equation 5.2 was applied to find the peak differences of the second window.

Table 6.1

The approximate width for the second window of pattern 24 in Table 6.3

peaks	1st	2nd	3rd	4th	5th
$P_{K(U)}$ (Hz)	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
	299	281	298	280	290
Apply Equation 5.2, $D_{k(u)}$ (Hz)	D_{21}	D_{22}	D_{23}	D_{24}	D_{25}
	1	19	2	20	10

The previous table shows how the ANN inputs are extracted and selected based on several factors which are U and K. Therefore, Tables 6.2, 6.3, 6.4, and 6.5 represent the training and testing data of two peaks for all the four data, A, A+B, C, and D respectively. Then, the differences of all the windows are presented in a vector $D_{k(u)}$ which is the input vector to ANN stage.

6.2 Results

First of all, it must be mentioned that the graphs of all the data types below show the difference (Error) between the desired speed and the ANN output in y-axis while the x-axis displays the desired speed. These graphs have two different results: training-data represented by blue dots, and testing-data results represented by red dots. In addition, the comparison of all ANN outputs is based on the testing-data accuracy which was considered as the overall accuracy; and the unit of the average error is RPM. Beginning with data A, the ANNs were examined with one hidden layer or two hidden layers in order to see which layer system had a better accuracy. It might be helpful to mention that the learning rates of the first and the second hidden layer that had been used with every training were 1 and 0.02 respectively, where the epochs was set to be 2000 iterations.

6.2.1 Data A Results

6.2.1.1 One Hidden Layer

Figures 6.5, 6.6, and 6.7 show the result of single hidden layer ANN that had 23 neurons, where each window supplied two peaks. These graphs were taken in a different number of windows: Figure 6.5 at two windows (5thW-7thW), Figure 6.6 at four windows (5thW-11thW), Figure 6.6 at all the windows (3rdW-15thW).

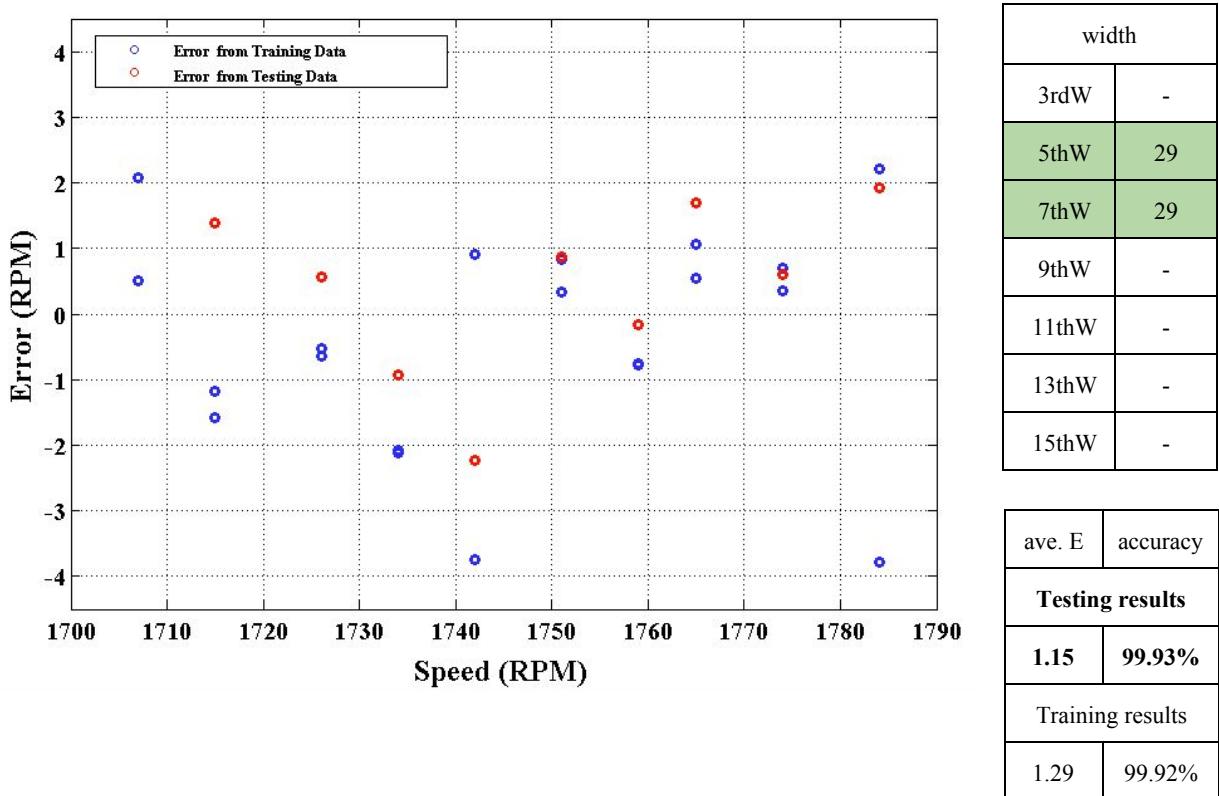
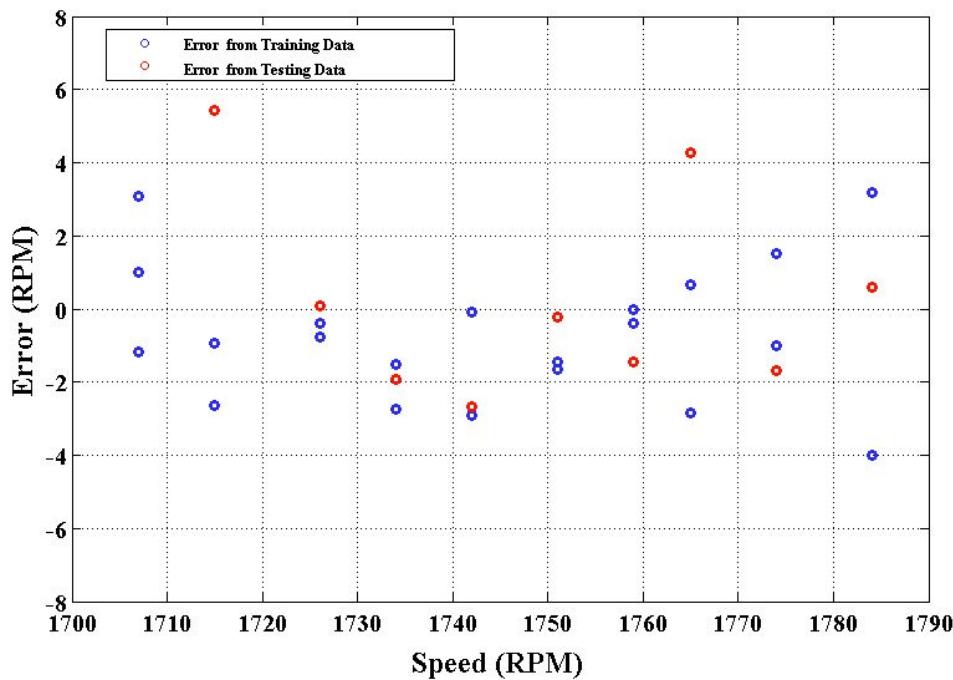


Figure 6.5: The prediction errors of data A, (AI1).



width	
3rdW	-
5thW	29
7thW	29
9thW	44
11thW	58
13thW	-
15thW	-

ave. E	accuracy
Testing results	
2	99.88%
Training results	
1.6	99.9%

No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
1	First HL: 23	Second HL: -	4	2

Figure 6.6: The prediction errors of data A, (AI2).

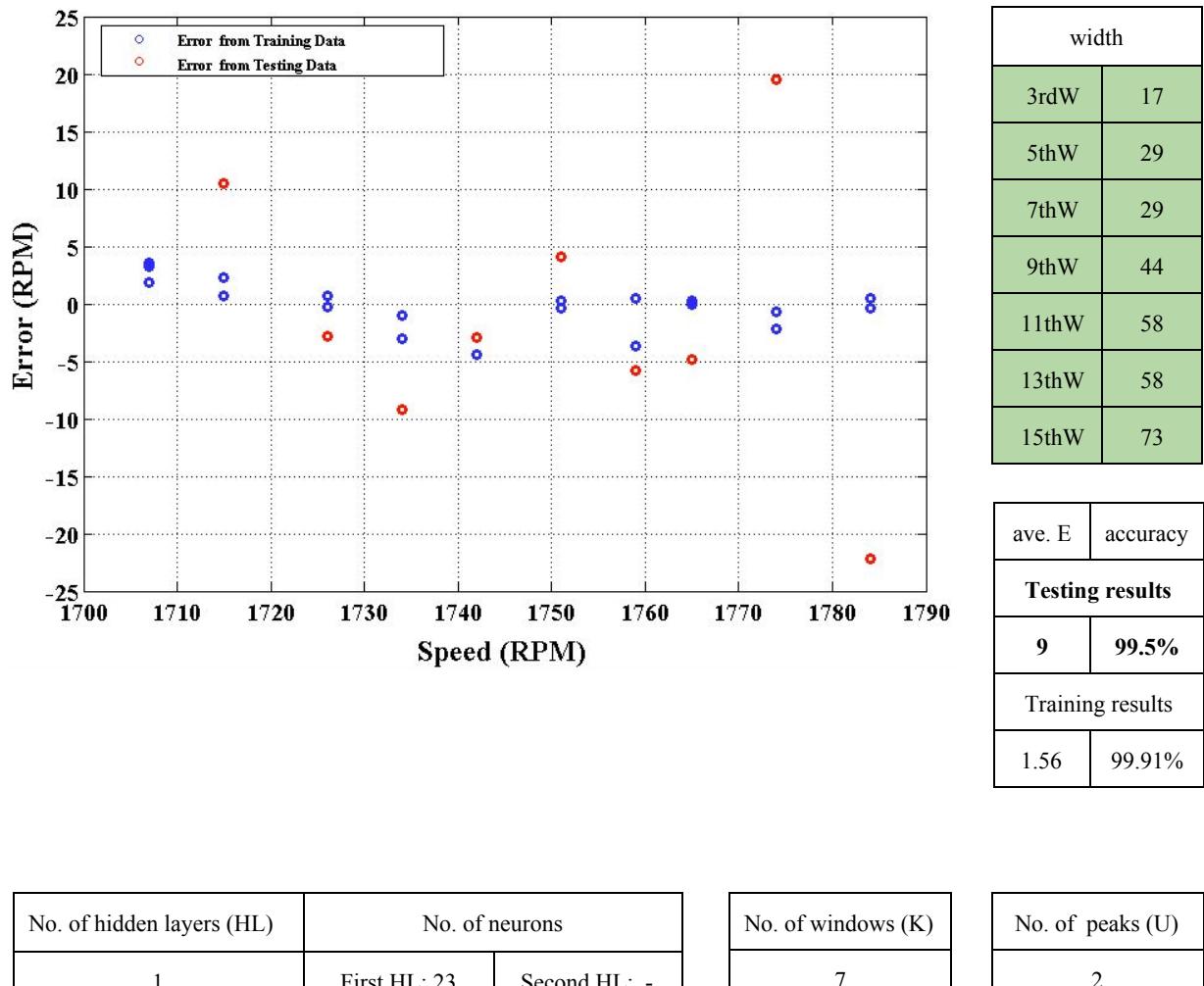


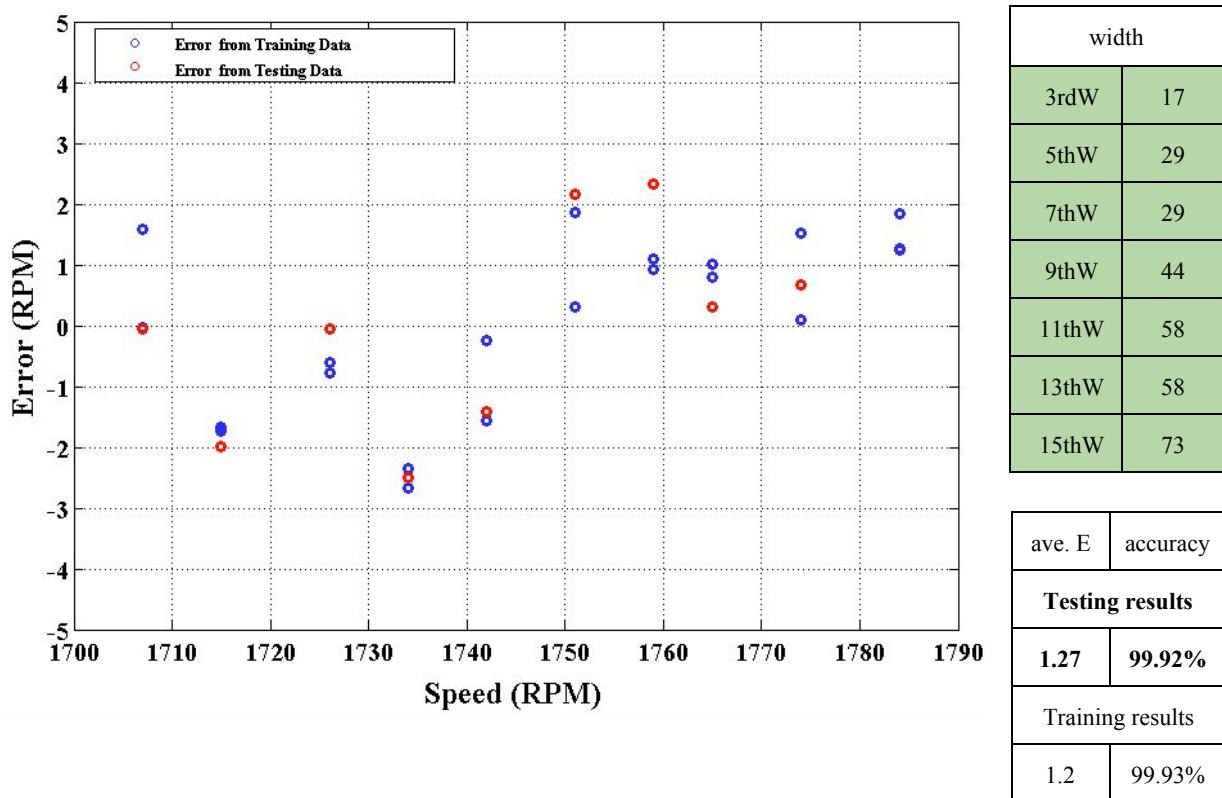
Figure 6.7: The prediction errors of data A, (AI3).

In Figure 6.5, the accuracy is very good where there are just two windows (5thW and 7thW). Comparing this result with the accuracy in Figures 6.6 and 6.7, the single hidden layer failed in a case that two peaks were extracted from each window. The worst case is seen in Figure 6.7 where four out of nine patterns have an error that is almost over 10 RPM. In other words, the effect of increasing the number of windows in single hidden layer negatively increases the average error, which in turn decreases the accuracy clearly seen in Figure 6.7.

6.2.1.2 Two Hidden Layers

Neurons Variation:

The second examination that was applied to data A was using ANNs with two hidden layers where all the windows (3rdW-15thW) were chosen to supply two peaks from each one. Thus, Figures 6.8, 6.9, and 6.10 illustrate the result of two hidden layers ANN where the neurons of these graphs were varied.



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 33	Second HL: 7	7	2

Figure 6.8: The prediction errors of data A, (AII1).

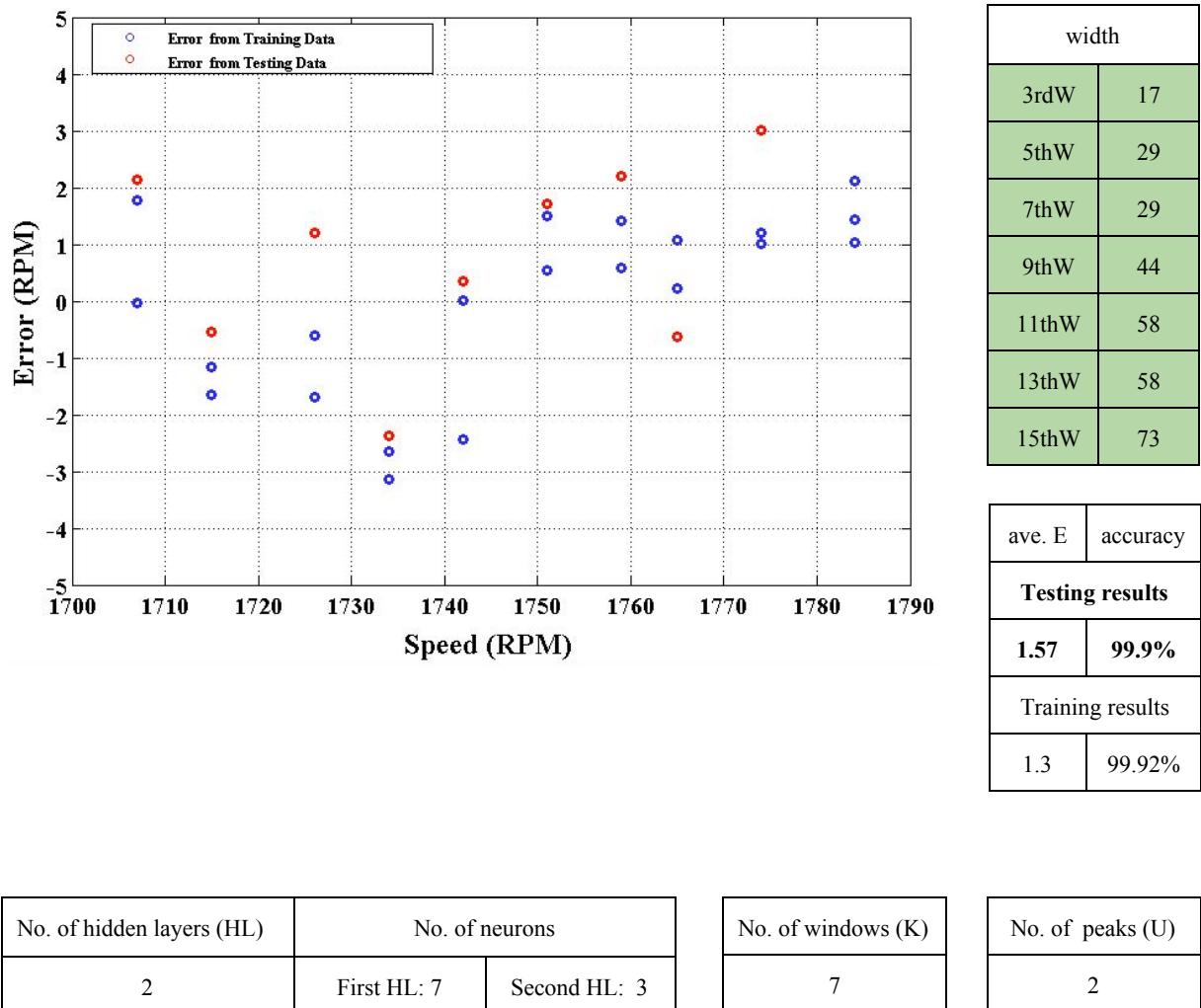


Figure 6.9: The prediction errors of data A, (AII2).

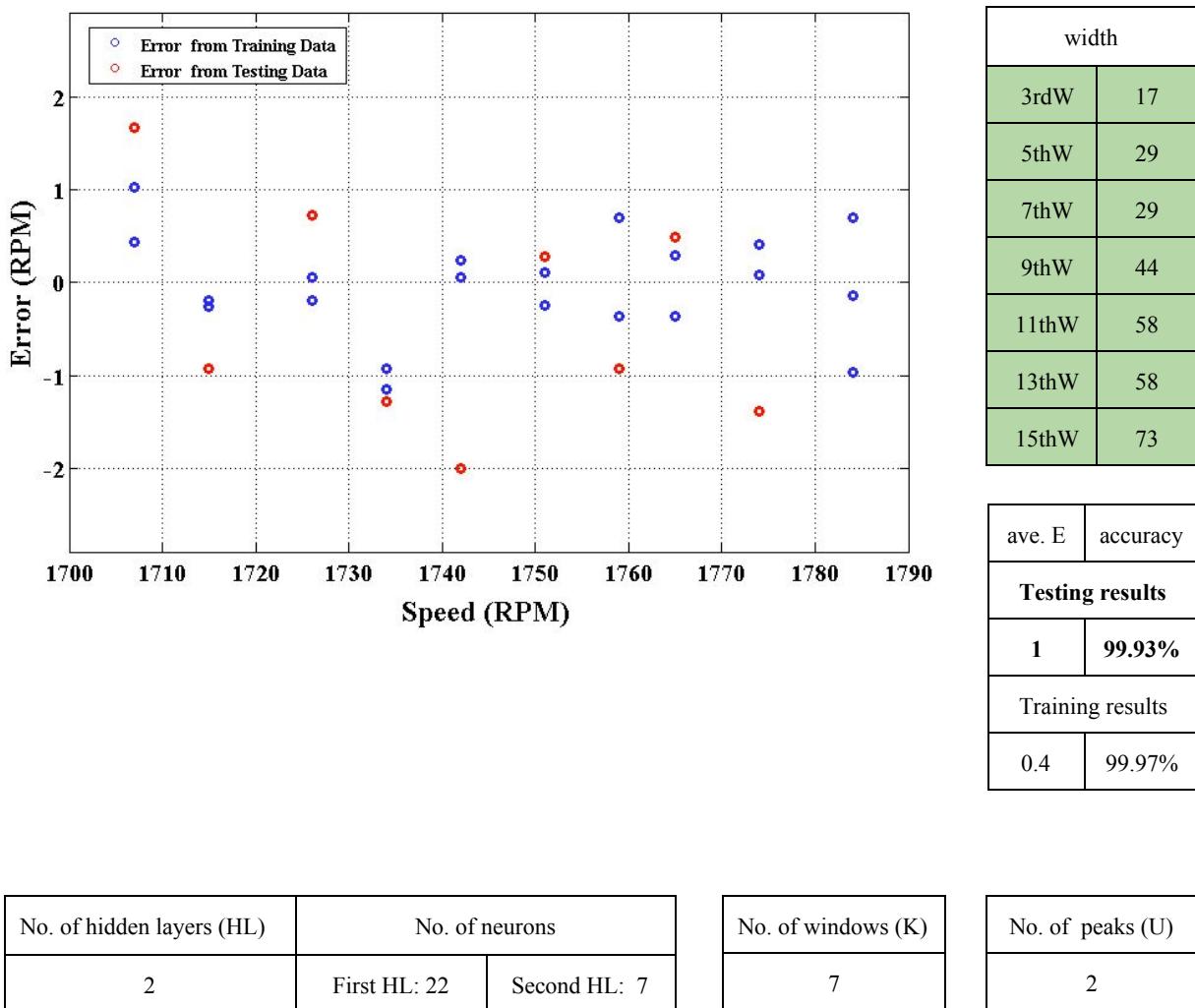
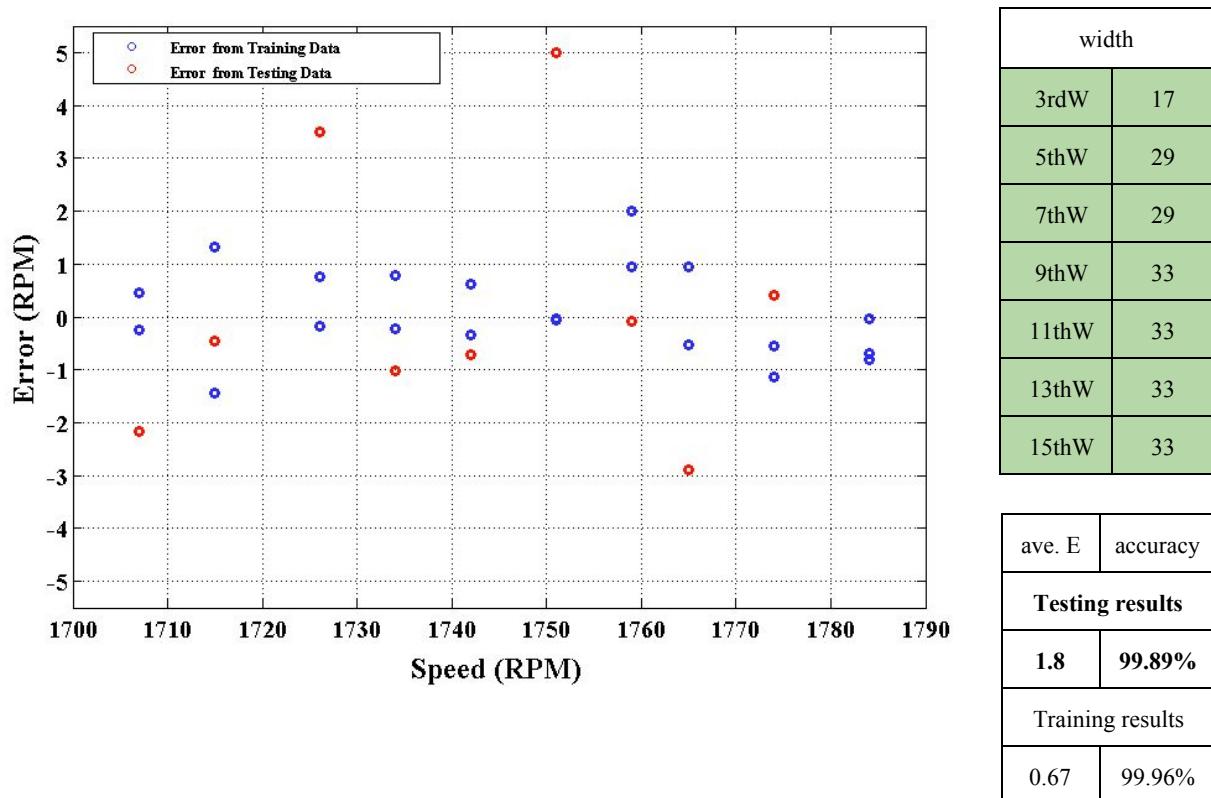


Figure 6.10: The prediction errors of data A, (AII3).

Figures 6.8, 6.9, and 6.10 show a significant improvement compared with the previous approach, which was one hidden layer ANN. Even though the numbers of the neurons were different from each Figure, 6.8, 6.9, and 6.10, the accuracy was very sophisticated. Therefore, the two hidden layers approach was chosen for the rest of data, A+B, C, and D.

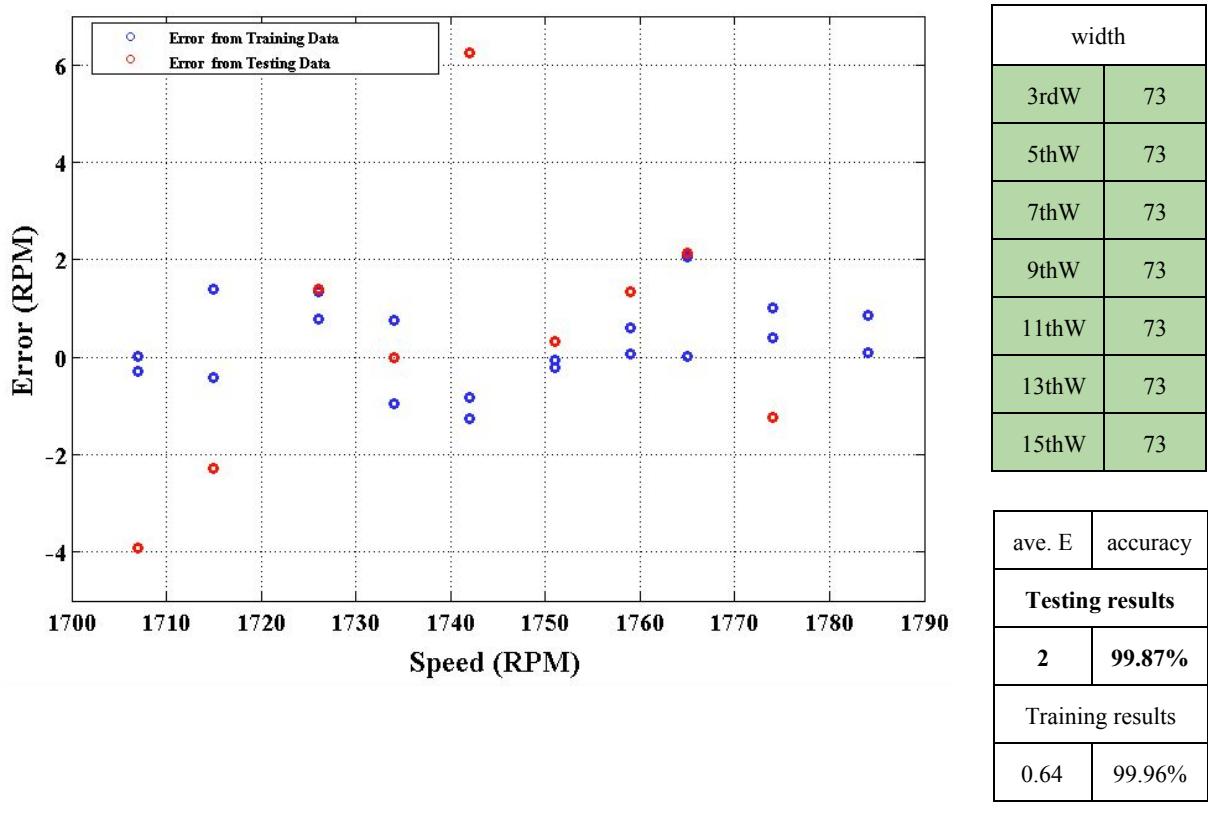
Width Variation:

Because the ANN design in Figure 6.10: Bc2 has achieved the best accuracy over all the past cases, it is chosen to see the effect of changing the width of the windows. Therefore, Figures 6.11 and 6.12 show the width variation, where the width values are listed beside each graph.



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	2

Figure 6.11 : The prediction errors of data A, (AII4).



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	2

Figure 6.12 : The prediction errors of data A, (AIIS).

Although some of the widths in Figure 6.11 are below the recommended values as illustrated in Table 5.4, the testing accuracy is still very good. On the other hand, the testing accuracy in Figure 6.12 is 2 RPM which is also a very good outcome despite the fact that all the widths were above the recommended values.

Peaks Variation:

In Figures 6.13 and 6.14, the number of peaks (U) was increased to examine its influence on the accuracy, so the ANN design that was used in Figure 6.10 is selected but with different U.

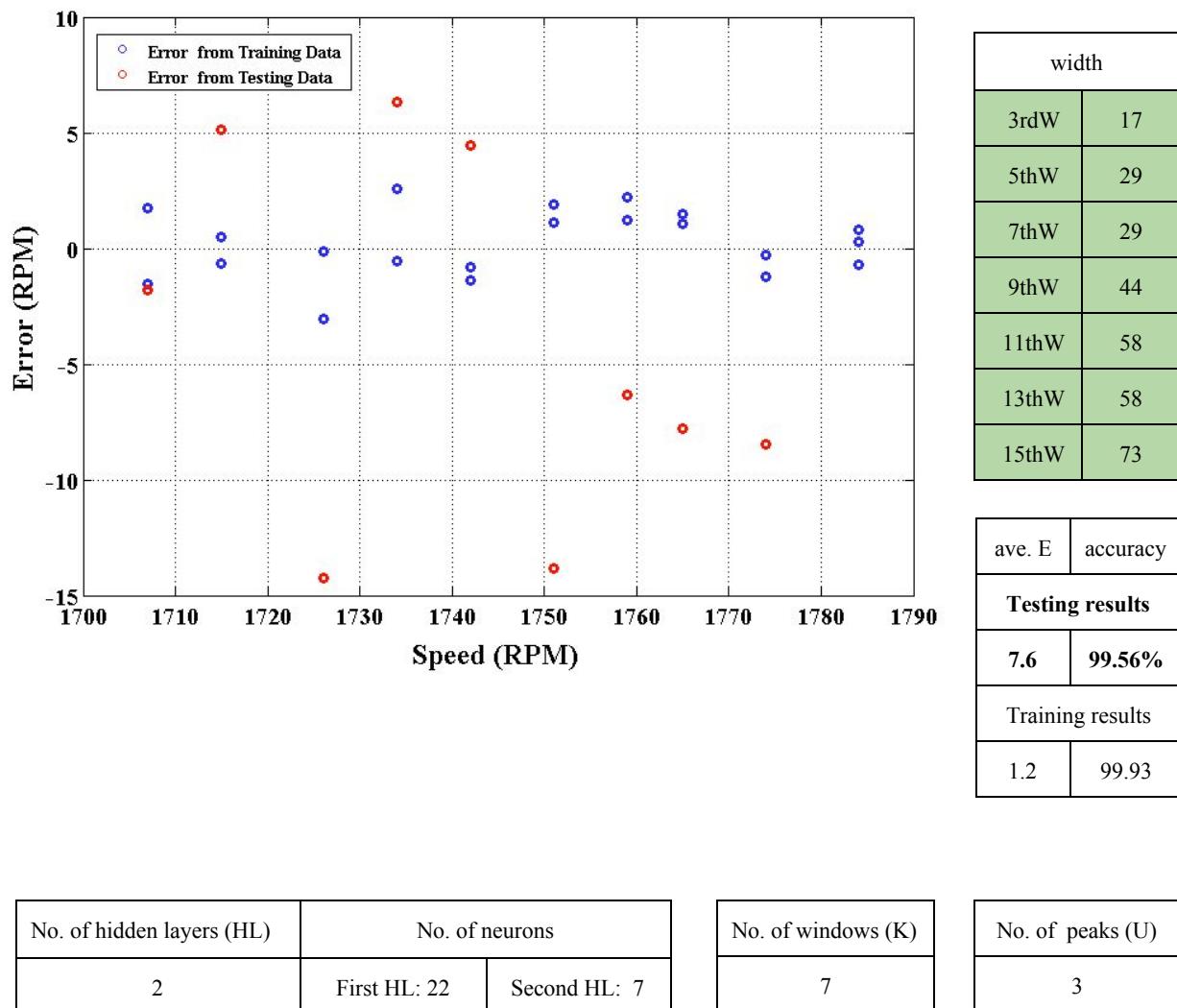
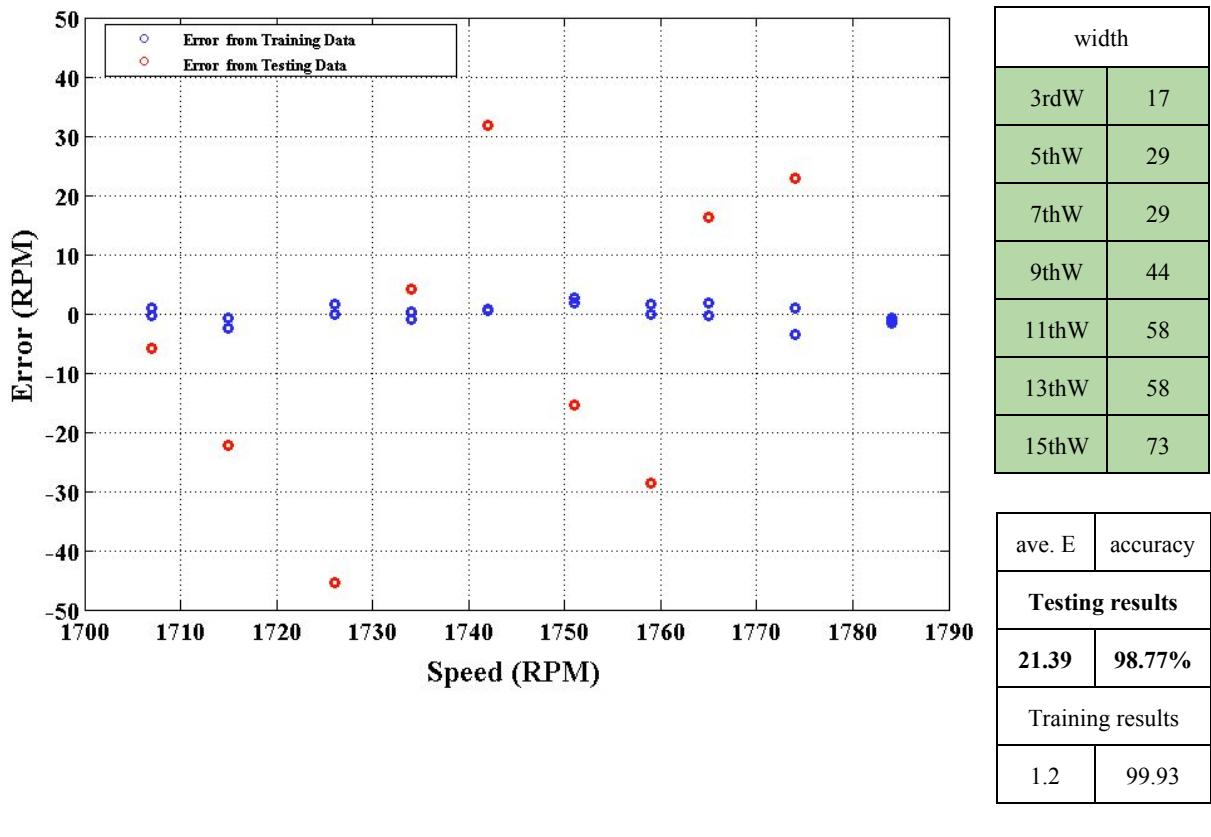


Figure 6.13 : The prediction errors of data A, (AII6).



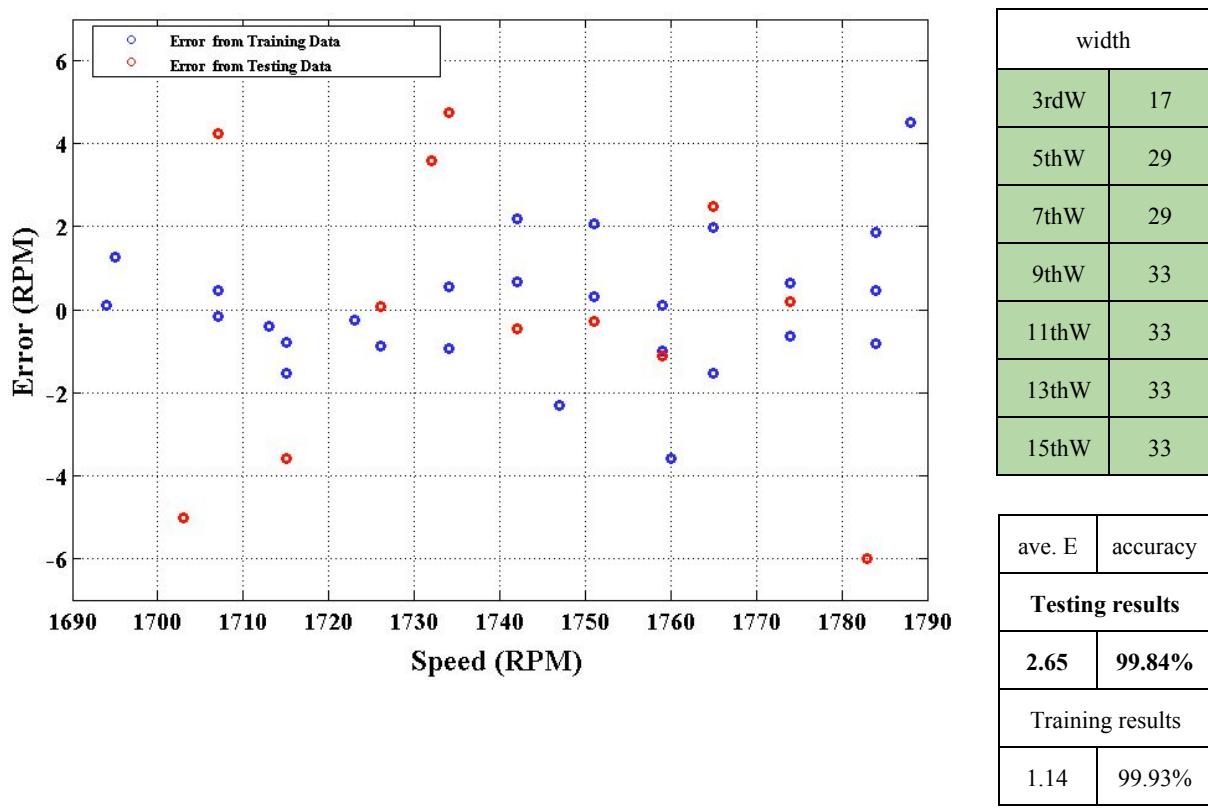
No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	5

Figure 6.14: The prediction errors of data A, (AII7).

It is clearly noticed that increasing the number of peaks negatively affects the accuracy as seen in Figure 6.13 and 6.14. Extracting more than two peaks would raise the average error, which causes a very poor accuracy compared with Figure 6.10. The average errors of all the previous Figures, 6.6-6.14, demonstrate that the rotor speed of motor A can be estimated by using the ANN sensorless method. Substantially, the better design is described in Figure 6.10, which has the best accuracy.

6.2.2 Data A+B Results

As detailed in Table 5.2, data A+B were obtained from two different motors that were identical in their characteristics, see Table 5.1. Therefore, these data, A and B, were combined in order to see if ANN's estimation can provide a good accuracy with two identical WRIMs. Therefore, Figures 6.15, 6.16, and 6.17 show the result of 40 patterns in various widths.



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	2

Figure 6.15 : The prediction errors of data A+B, (A+BII1).

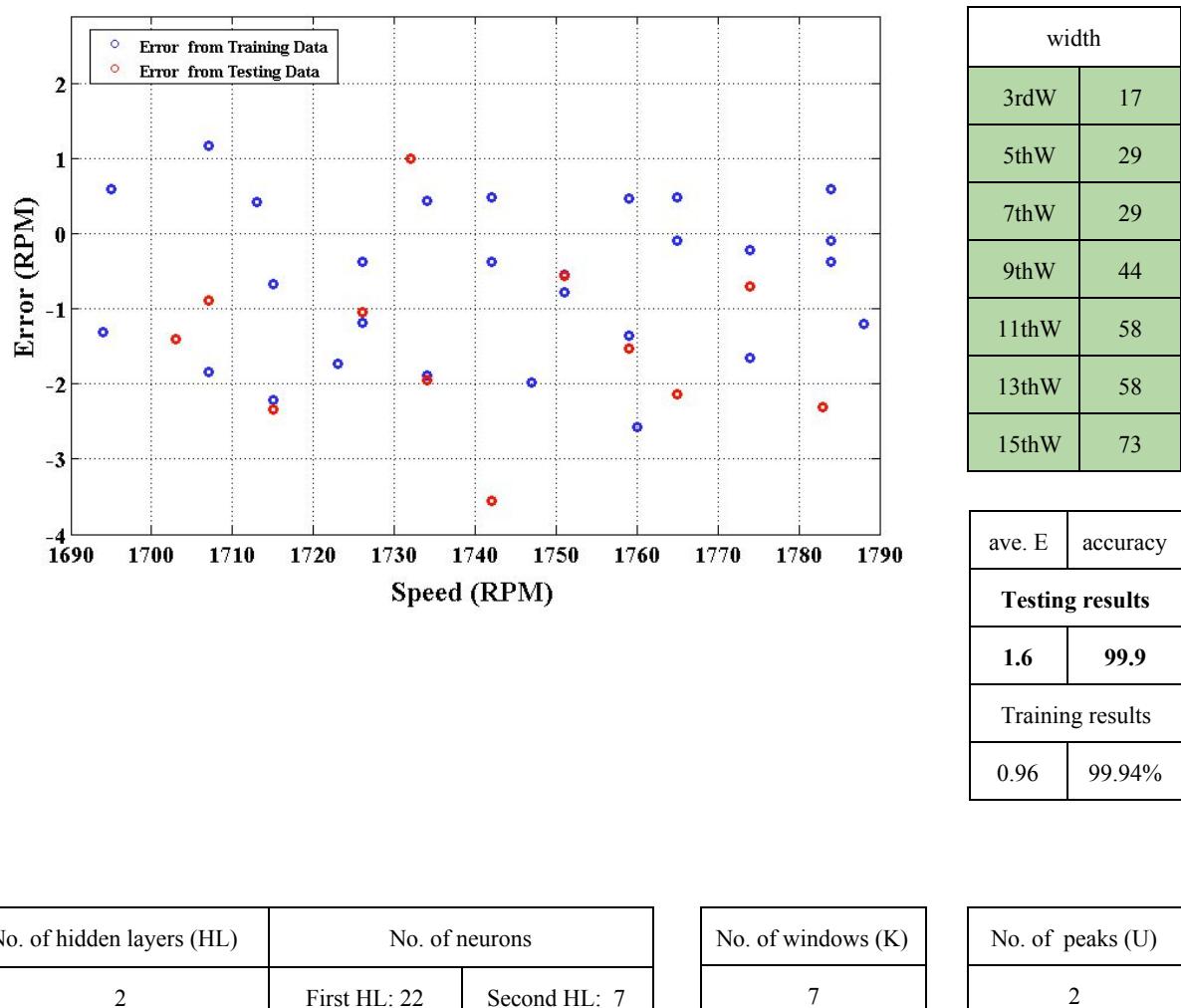
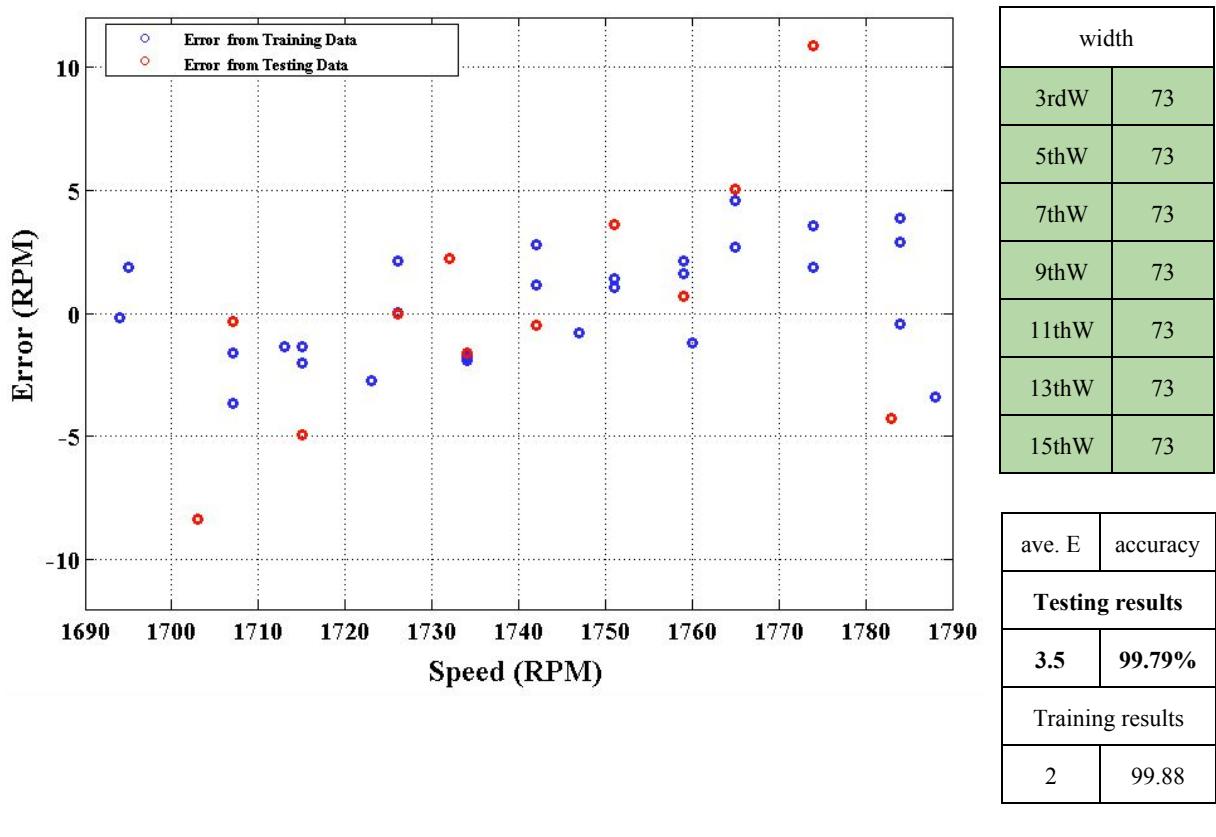


Figure 6.16: The prediction errors of data A+B, (A+BII2).



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	2

Figure 6.17: The prediction errors of data A+B, (A+BII3).

First, A+B graphs have the same design details as Figure 6.10 except the widths might be varied. The precision in Figures 6.15 and 6.17 is lower than in Figure 6.16. However, the results of these graphs depict that ANNs have the successful capability of estimating the rotor speed of identical motors. Indeed, Figure 6.16 shows a very good accuracy close to the accuracy in Figure 6.10 where just data A was used.

6.2.3 Data C Results

Data C that was taken from a SCIM was examined as same as data A. All the factors, K, U, widths, layers, and neurons were utilized to find the best accuracy that could be achieved. Therefore, Figures 6.18 to 6.21 can give an overall perspective of applying the proposed method to a squirrel cage induction motor (SCIM).

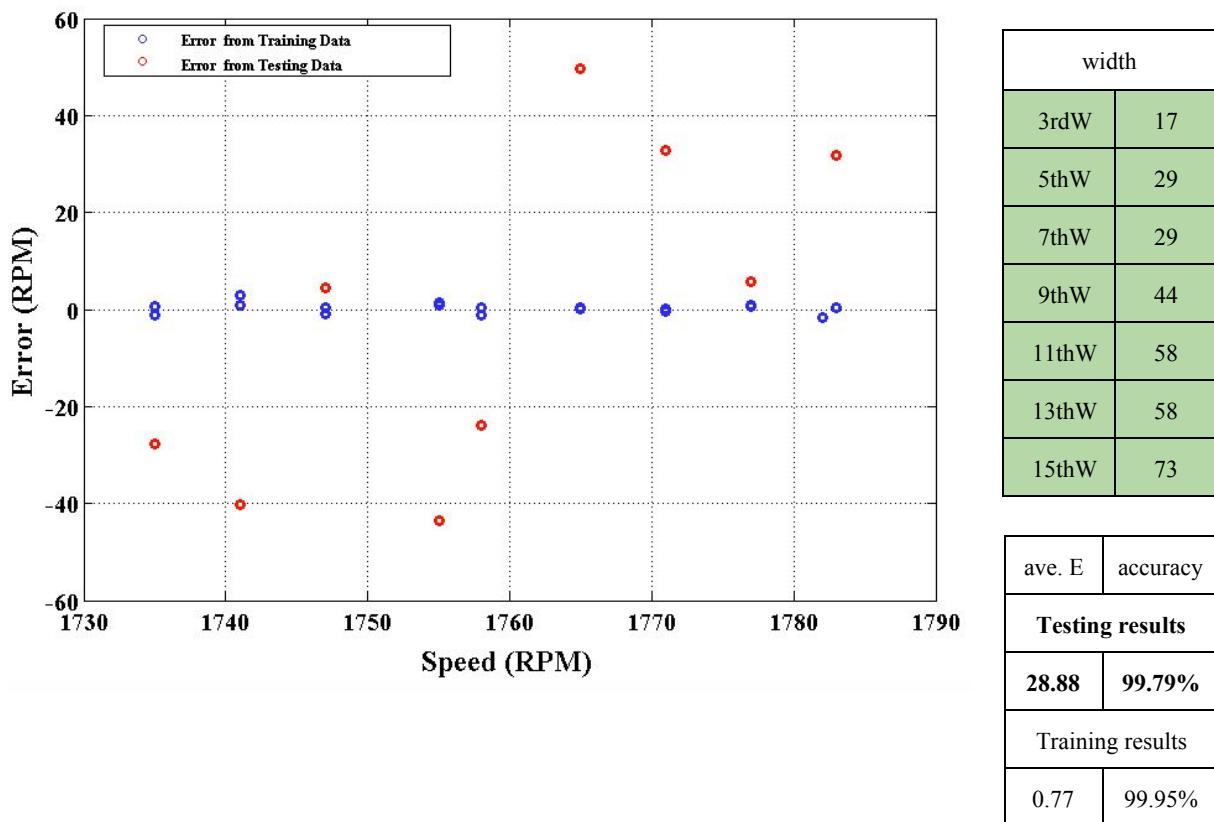


Figure 6.18 : The prediction errors of data C, (CII1).

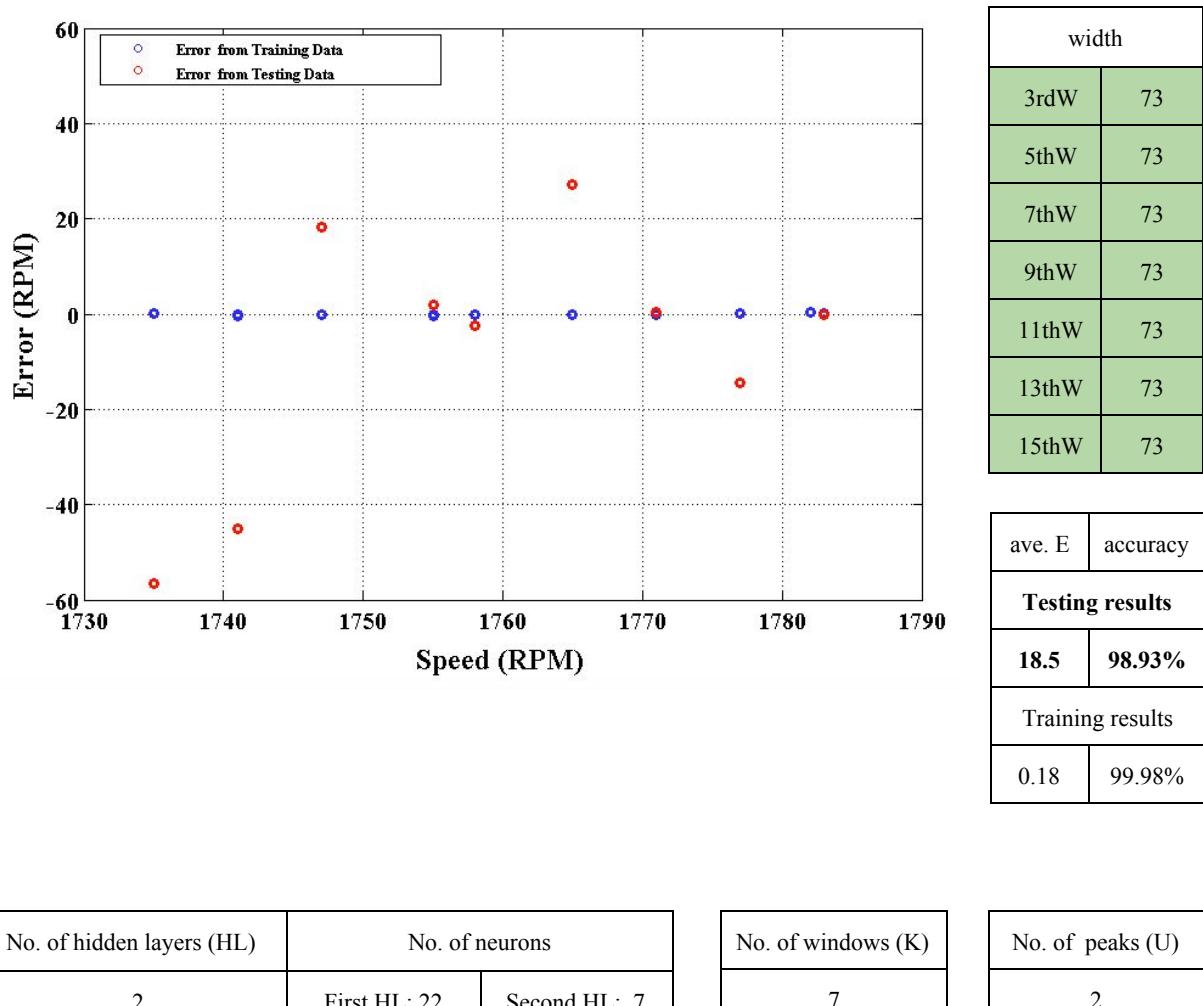
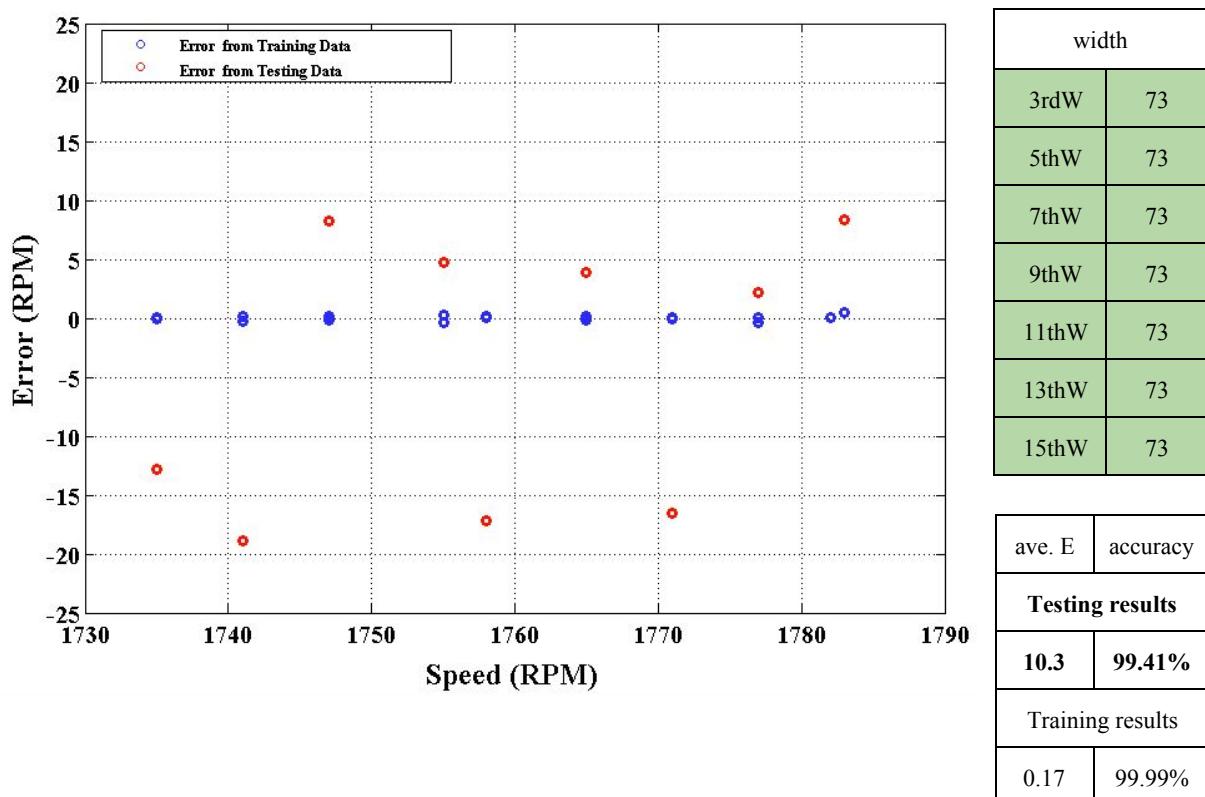


Figure 6.19 : The prediction errors of data C, (CII2).

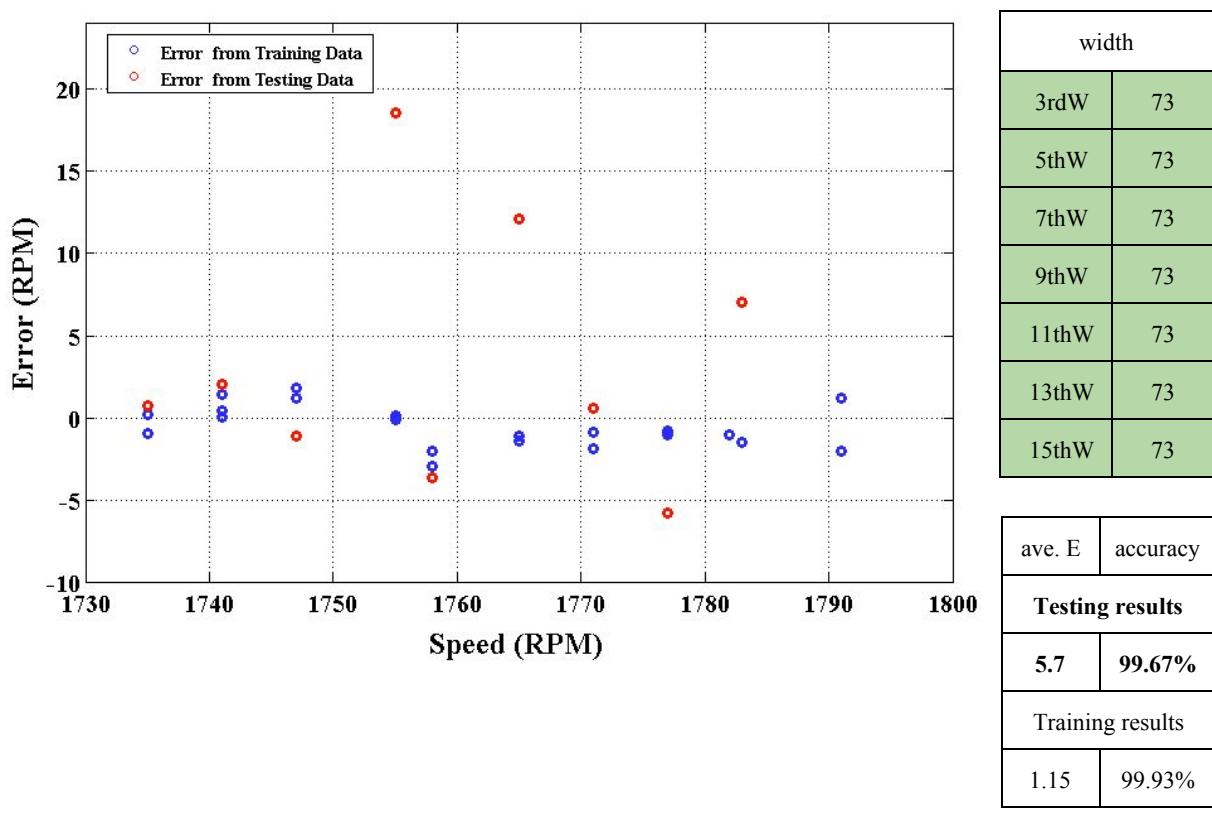
Unfortunately, Figures 6.18 and 6.19 show a very poor testing accuracy that could not be accepted although Figure 6.19 had better average error than Figure 6.18 by 10 RPM. The impact of varying the widths did not add enough accuracy for Figures 6.18 and 6.19. On the other hand, the training results of these graphs have very small errors, which are less than one RPM. Thus, the ANN design in Figure 6.19 was developed several times with different peaks and neurons. It was found that five peaks were the optimal number of U. In addition, the widths were the same

as Figure 6.19, and all the windows were used. However, the testing results of these graphs was not convenient for the implementation. The average error in Figure 6.21 was the most improved result that was achieved for data C. By looking at the red dots in Figure 6.21, five out of nine patterns are very close to their training dots, which means that these patterns have a very good accuracy. But the other patterns that are over ± 5 RPM are causing the overall poor accuracy.



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 22	Second HL: 7	7	5

Figure 6.20 : The prediction errors of data C, (CII3).



No. of hidden layers (HL)	No. of neurons		No. of windows (K)	No. of peaks (U)
2	First HL: 11	Second HL: 11	7	5

Figure 6.21 : The prediction errors of data C, (CII4).

6.2.4 Data D Results

Data D was taken from motor A, and it was the same data as A but with more patterns and a different T_m , which was 500 m second. The purpose was to decrease the time delay that had occurred in data A, B+A, and C because they were monitored at $T_m=1$ second. Data D has 67 patterns, which is more than A+B.

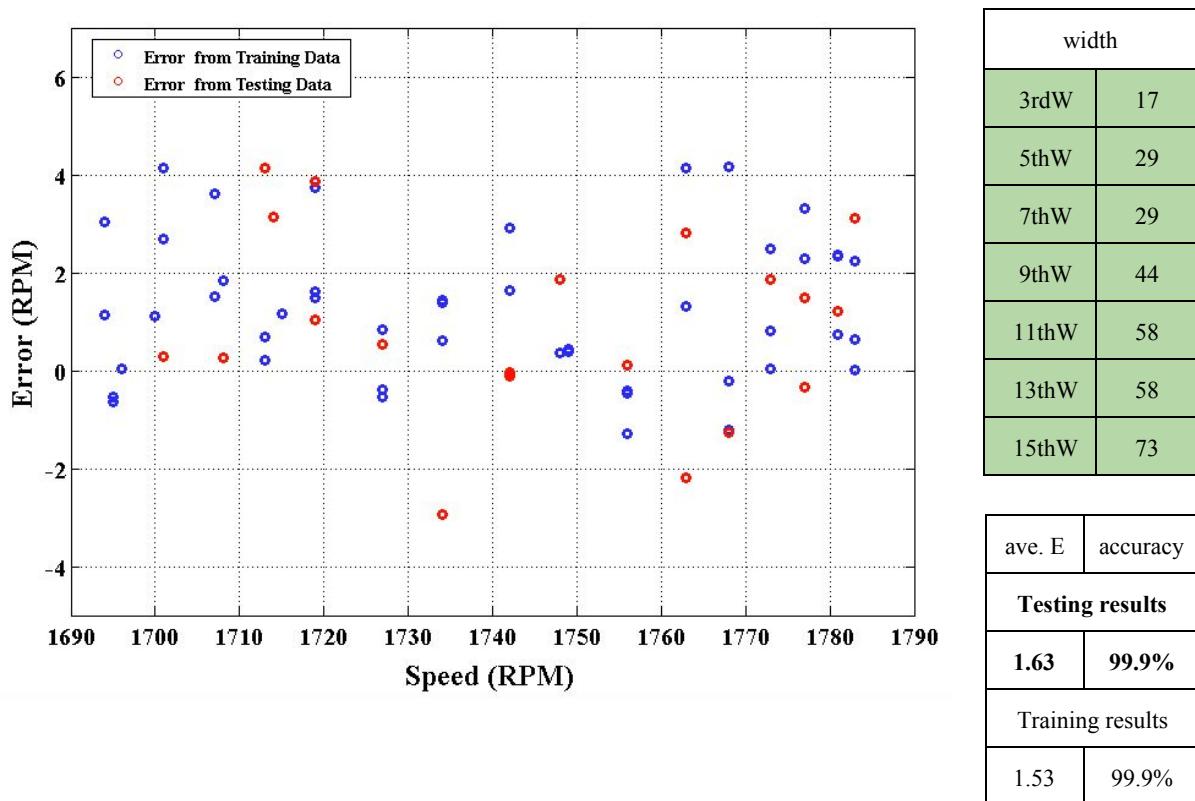


Figure 6.22: The prediction errors of data D, (DII1).

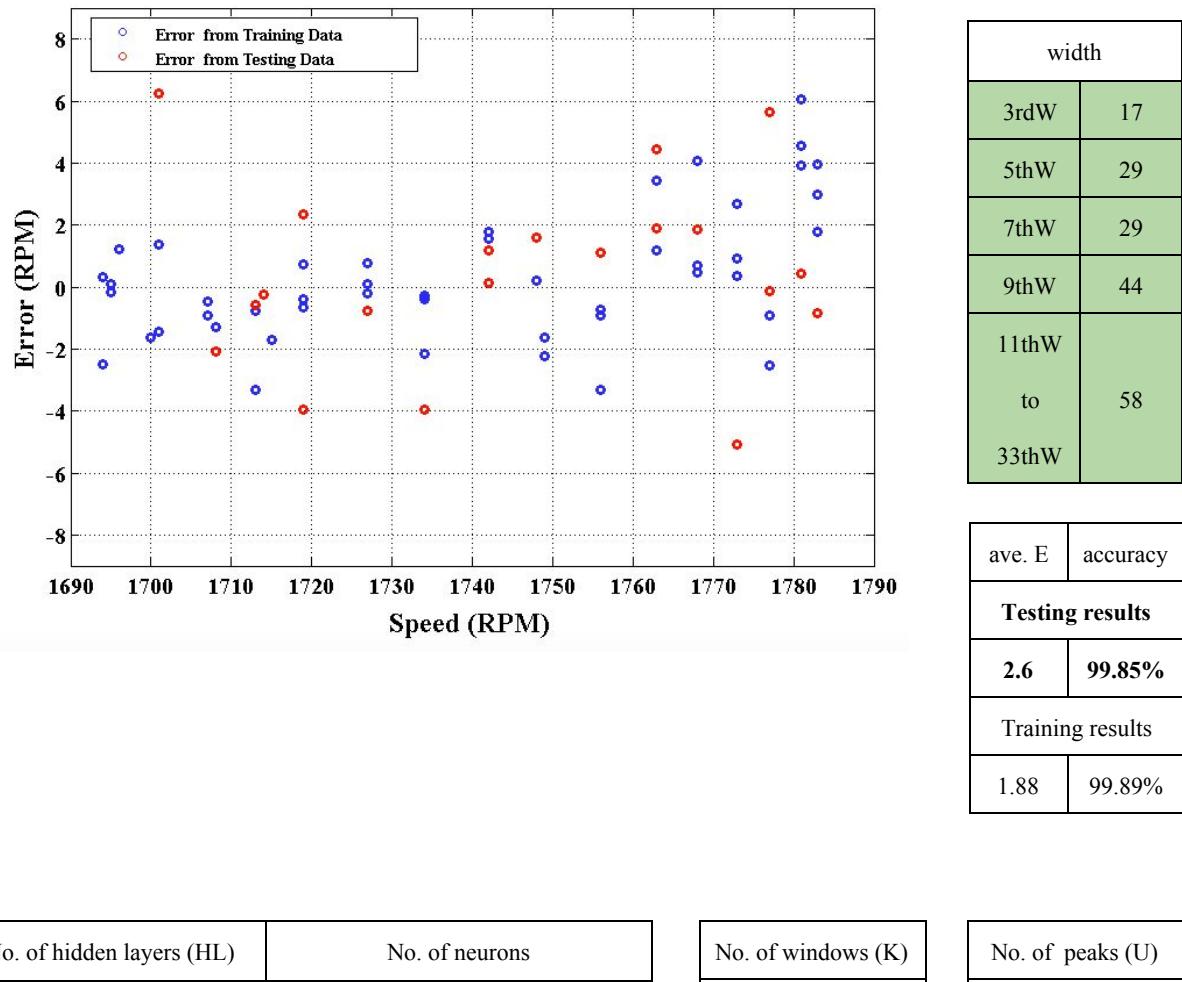


Figure 6.23: The prediction errors of data D, (DII1).

It is obvious to notice that the result of data D confirms the success of ANN speed estimation of WRIMs even when the acquisition time is half a second. Figure 6.22 illustrates a very good testing accuracy, where the ANN design is the same as Figure 6.10. Because the Nyquist frequency of data D was 2000 Hz, the windows can be extended to the 33rd harmonic without any problem. Thus, all the windows from 3rdW to 33rdW were applied to data D. It

should be mentioned that the width of the windows from 11th to 33rd is the same, which was 58 Hz. As a matter of fact, Figure 6.23 depicts a good accuracy, where the new windows decrease the testing accuracy of Figure 6.22 by less than 1 RPM. Finally, to sum up the overall results of all the data, the proposed method can precisely estimate the speed of motor A and B but not C.

Table 6.2
The peaks and differences for data A

Pattern	3rdW						5thW						7thW						9thW						11thW						13thW						speed	
	Peaks			Diff.			Peaks			Diff.			Peaks			Diff.			Peaks			Diff.			Peaks			Diff.										
	P11	P12	D11	D12	P21	P22	D21	D22	P31	P32	D31	D32	P41	P42	D41	D42	P51	P52	D51	D52	P61	P62	D61	D62	P71	P72	D71	D72	d									
training	1	174	168	6	12	297	276	3	24	417	410	3	10	528	502	12	38	654	601	6	59	729	778	51	2	851	855	49	45	1784								
	2	163	179	17	1	299	297	1	3	417	416	3	4	503	506	37	34	605	629	55	31	750	729	30	51	840	866	60	34	1784								
	3	179	165	1	15	297	285	3	15	417	416	3	4	497	513	43	27	652	641	8	19	726	723	54	57	867	838	33	62	1784								
	4	176	167	4	13	295	296	5	4	415	407	5	13	525	527	15	13	635	629	25	31	760	755	20	25	891	838	9	62	1774								
	5	162	177	18	3	295	294	5	6	415	412	5	8	531	511	9	29	650	606	10	54	777	776	3	4	889	894	11	6	1774								
	6	174	173	6	7	293	291	7	9	413	409	7	11	515	502	25	38	605	646	55	14	726	732	54	48	846	894	54	6	1765								
	7	177	167	3	13	293	283	7	17	413	394	7	26	495	529	45	11	623	646	37	14	740	726	40	54	882	889	18	11	1765								
	8	176	170	4	10	292	299	8	1	412	411	8	9	503	536	37	4	648	659	12	1	723	728	57	52	831	883	69	17	1759								
	9	162	174	18	6	292	280	8	20	412	413	8	7	516	508	24	32	646	647	14	13	775	766	5	14	853	832	47	68	1759								
	10	169	163	11	17	290	299	10	1	410	417	10	3	521	534	19	6	641	634	19	26	729	745	51	35	889	827	11	73	1751								
testing	11	167	174	13	6	290	289	10	11	410	398	10	22	531	505	9	35	640	618	20	42	739	776	41	4	874	830	26	70	1751								
	12	166	164	14	16	289	288	11	12	409	408	11	12	512	529	28	11	625	630	35	30	774	730	6	50	881	869	19	31	1742								
	13	169	172	11	8	289	288	11	12	408	409	12	11	513	521	27	18	645	619	15	41	775	750	5	30	841	853	59	47	1742								
	14	171	174	9	6	287	286	13	14	407	406	13	14	508	522	32	18	636	604	24	56	743	727	37	53	869	870	31	30	1734								
	15	175	172	5	8	287	273	13	27	407	416	13	4	507	515	33	25	652	642	8	18	730	764	50	16	891	856	9	44	1734								
	16	178	174	2	6	285	286	15	14	405	409	15	11	529	512	11	28	619	624	41	36	775	765	5	15	866	860	34	40	1726								
	17	176	178	4	2	285	286	15	14	405	393	15	27	524	525	16	15	629	640	31	20	761	747	19	33	896	840	4	60	1726								
	18	165	179	15	1	283	282	17	18	403	402	17	18	527	510	13	30	624	618	36	42	766	737	14	43	895	885	5	15	1715								
	19	178	177	2	3	283	273	17	27	403	393	17	27	536	514	4	26	604	623	56	37	744	770	36	10	853	869	47	31	1715								
	20	168	162	12	18	281	272	19	28	401	402	19	18	496	504	44	36	616	621	44	39	741	755	39	25	876	895	24	5	1707								
	21	167	176	13	4	281	282	19	18	401	402	19	18	519	539	21	1	636	624	24	36	769	725	11	55	897	856	3	44	1707								
testing	22	171	169	9	11	295	291	5	9	415	414	5	6	514	525	26	15	606	640	54	20	734	739	46	41	891	828	9	72	1774								
	23	179	171	1	9	293	292	7	8	413	418	7	2	536	503	4	37	654	634	6	26	747	769	33	11	856	894	44	6	1765								
	24	175	173	5	7	292	290	8	10	412	411	8	9	512	535	28	5	628	644	32	16	767	725	13	55	830	887	70	13	1759								
	25	170	163	10	17	290	297	10	3	410	399	10	21	536	512	4	28	640	616	20	44	753	739	27	41	856	894	44	6	1751								
	26	169	179	11	1	288	289	12	11	408	409	12	11	498	496	42	44	618	653	42	7	738	740	42	40	895	897	5	3	1742								
	27	166	179	14	1	287	285	13	15	407	410	13	10	497	512	43	28	633	628	27	32	764	742	16	38	861	891	39	9	1734								
	28	176	165	4	15	285	286	15	14	405	406	15	14	536	499	4	41	627	617	33	43	732	761	48	19	829	828	71	72	1726								
	29	168	176	12	4	283	279	17	21	403	419	17	1	522	514	18	26	628	634	32	26	737	735	43	45	849	884	51	16	1715								
	30	166	176	14	4	281	282	19	18	401	402	19	18	502	504	38	36	636	612	24	48	736	756	44	24	840	871	60	29	1707								

Table 6.3

The peaks and differences for data A+B

Pattern	3rdW				5thW				7thW				9thW				11thW				13thW				speed d					
	Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.							
	P11	P12	D11	D12	P21	P22	D21	D22	P31	P32	D31	D32	P41	P42	D41	D42	P51	P52	D51	D52	P61	P62	D61	D62	P71	P72	D71	D72		
training	1	174	168	6	12	297	276	3	24	417	410	3	10	528	502	12	38	654	601	6	59	729	778	51	2	851	855	49	45	1784
	2	163	179	17	1	299	297	1	3	417	416	3	4	503	37	34	605	629	55	31	750	729	30	51	840	866	60	34	1784	
	3	179	165	1	15	297	285	3	15	417	416	3	4	497	513	43	27	652	641	8	19	726	723	54	57	867	838	33	62	1784
	4	171	169	9	11	295	291	5	9	415	414	5	6	514	525	26	15	606	640	54	20	734	739	46	41	891	828	9	72	1774
	5	162	177	18	3	295	294	5	6	415	412	5	8	531	511	9	29	650	606	10	54	777	776	3	4	889	894	11	6	1774
	6	179	171	1	9	293	292	7	8	413	418	7	2	536	503	4	37	654	634	6	26	747	769	33	11	856	894	44	6	1765
	7	177	167	3	13	293	283	7	17	413	394	7	26	495	529	45	11	623	646	37	14	740	726	40	54	882	889	18	11	1765
	8	175	173	5	7	292	290	8	10	412	411	8	9	512	535	28	5	628	644	32	16	767	725	13	55	830	887	70	13	1759
	9	162	174	18	6	292	280	8	20	412	413	8	7	516	508	24	32	646	647	14	13	775	766	5	14	853	832	47	68	1759
	10	170	163	10	17	290	297	10	3	410	399	10	21	536	512	4	28	640	616	20	44	753	739	27	41	856	894	44	6	1751
testing	11	167	174	13	6	290	289	10	11	410	398	10	22	531	505	9	35	640	618	20	42	739	776	41	4	874	830	26	70	1751
	12	169	179	11	1	288	289	12	11	408	409	12	11	498	496	42	44	618	653	42	7	738	740	42	40	895	897	5	3	1742
	13	169	172	11	8	289	288	11	12	408	409	12	11	513	522	27	18	645	619	15	41	775	750	5	30	841	853	59	47	1742
	14	166	179	14	1	287	285	13	15	407	410	13	10	497	512	43	28	633	628	27	32	764	742	16	38	861	891	39	9	1734
	15	175	172	5	8	287	273	13	27	407	416	13	4	507	515	33	25	652	642	8	18	730	764	50	16	891	856	9	44	1734
	16	176	165	4	15	285	286	15	14	405	406	15	14	536	499	4	41	627	617	33	43	732	761	48	19	829	828	71	72	1726
	17	176	178	4	2	285	286	15	14	405	393	15	27	524	525	16	15	629	640	31	20	761	747	19	33	896	840	4	60	1726
	18	168	176	12	4	283	279	17	21	403	419	17	1	522	514	18	26	628	634	32	26	737	735	43	45	849	884	51	16	1715
	19	178	177	2	3	283	273	17	27	403	393	17	27	536	514	4	26	604	623	56	37	744	770	36	10	853	869	47	31	1715
	20	166	176	14	4	281	282	19	18	401	402	19	18	502	504	38	36	636	612	24	48	736	756	44	24	840	871	60	29	1707
testing	21	167	176	13	4	281	282	19	18	401	402	19	18	519	539	21	1	636	624	24	36	769	725	11	55	897	856	3	44	1707
	22	174	163	6	17	298	272	2	28	417	403	3	17	535	499	5	41	628	651	32	9	757	761	23	19	881	880	19	20	1788
	23	171	162	9	18	293	282	7	18	413	395	7	25	495	535	45	5	641	657	19	3	738	751	42	29	837	882	63	18	1760
	24	179	169	1	11	299	281	1	19	409	392	11	28	512	523	28	17	608	609	52	51	727	728	53	52	841	843	59	57	1747
	25	179	176	1	4	285	284	15	16	405	404	15	16	497	513	43	27	617	624	43	36	754	736	26	44	874	838	26	62	1723
	26	165	173	15	7	282	272	18	28	402	404	18	16	532	525	8	15	613	653	47	7	738	741	42	39	853	855	47	45	1713
	27	167	178	13	2	279	298	21	2	399	408	21	12	534	506	6	34	657	651	3	9	771	773	9	7	893	898	7	2	1695
	28	172	175	8	5	278	281	22	19	398	403	22	17	496	495	44	45	616	633	44	27	764	754	16	26	886	894	14	6	1694
	29	176	167	4	13	295	296	5	4	415	407	5	13	525	527	15	13	635	629	25	31	760	755	20	25	891	838	9	62	1774
	30	174	173	6	7	293	291	7	9	413	409	7	11	515	502	25	38	605	646	5	14	726	732	54	48	846	894	54	6	1765
	31	176	170	4	10	292	299	8	1	412	411	8	9	503	536	37	4	648	659	12	1	723	728	57	52	831	883	69	17	1759
	32	169	163	11	17	290	299	10	1	410	417	10	3	521	534	19	6	641	634	19	26	729	745	51	35	889	827	11	73	1751
	33	166	164	14	16	289	288	11	12	409	408	11	12	512	529	28	11	625	630	35	30	774	730	6	50	881	869	19	31	1742
testing	34	171	174	9	6	287	286	13	14	407	406	13	14	508	522	32	18	636	604	24	56	743	727	37	53	869	870	31	30	1734
	35	178	174	2	6	285	286	15	14	405	409	15	11	529	512	11	28	619	624	41	36	775	765	5	15	866	860	34	40	1726
	36	165	179	15	1	283	282	17	18	403	402	17	18	527	510	13	30	624	618	36	42	766	737	14	43	895	885	5	15	1715
	37	168	162	12	18	281	272	19	28	401	402	19	18	496	504	44	36	616	621	44	39	741	755	39	25	876	895	24	5	1707
	38	163	166	17	14	299	276	1	24	417	419	3	1	517	537	23	3	625	651	35	9	721	731	59	49	862	826	38	74	1783
	39	173	174	7	6	286	277	14	23	406	407	14	13	513	537	27	3	649	648	11	12	775	734	5	46	860	894	40	6	1732
	40	173	174	7	6	281	280	19	20	401	400	19	20	537	520	3	20	618	645	42	15	721	729	59	51	886	895	14	5	1703

Table 6.4

The peaks and differences for data C

Pattern	3rdW				5thW				7thW				9thW				11thW				13thW				15thW				speed				
	Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.						
	P11	P12	D11	D12	P21	P22	D21	D22	P31	P32	D31	D32	P41	P42	D41	D42	P51	P52	D51	D52	P61	P62	D61	D62	P71	P72	D71	D72					
training	1	178	166	2	14	299	282	1	18	413	402	7	18	530	503	10	37	625	619	35	41	741	763	39	17	866	827	34	73	1791			
	2	176	165	4	15	293	281	7	19	398	395	22	25	495	524	45	16	627	647	33	13	736	727	44	53	866	865	34	35	1791			
	3	178	166	2	14	298	272	2	28	401	398	19	22	526	534	14	6	641	643	19	17	739	775	41	5	865	836	35	64	1791			
	4	164	172	16	8	299	291	1	9	403	399	17	21	498	507	42	33	656	643	4	17	724	772	56	8	871	851	29	49	1782			
	5	177	165	3	15	270	272	30	28	410	405	10	15	537	539	3	1	646	639	14	21	771	779	9	1	862	845	38	55	1783			
	6	168	175	12	5	272	293	28	7	395	416	25	4	515	535	25	5	654	651	6	9	733	776	47	4	874	899	26	1	1777			
	7	176	164	4	16	284	281	16	19	403	399	17	21	531	503	9	37	634	610	26	50	779	732	1	48	874	890	26	10	1777			
	8	169	174	11	6	271	298	29	2	414	393	6	27	536	508	4	32	606	602	54	58	768	756	12	24	846	879	54	21	1771			
	9	166	164	14	16	284	291	16	9	411	402	9	18	498	530	42	10	650	637	10	23	740	729	40	51	878	846	22	54	1771			
	10	166	173	14	7	282	292	18	8	411	395	9	25	513	514	27	26	641	634	19	26	743	722	37	58	863	867	37	33	1765			
	11	163	165	17	15	270	295	30	5	390	414	30	6	497	502	43	38	620	658	40	2	731	755	49	25	881	852	19	48	1765			
	12	176	169	4	11	299	272	1	28	419	392	1	28	521	528	19	12	648	655	12	5	756	762	24	18	853	885	47	15	1758			
	13	167	176	13	4	297	278	3	22	400	393	20	27	539	520	1	20	608	650	52	10	757	764	23	16	838	884	62	16	1758			
	14	170	176	10	4	299	279	1	21	419	394	1	26	498	497	42	43	649	642	11	18	732	725	48	55	843	838	57	62	1755			
	15	166	167	14	13	298	294	2	6	393	416	27	4	526	518	14	22	633	621	27	39	730	777	50	3	887	845	13	55	1755			
	16	175	168	5	12	298	296	2	4	393	412	27	8	504	526	36	14	622	644	38	16	748	756	32	24	878	879	22	21	1747			
	17	165	179	15	1	299	274	1	26	419	397	1	23	510	539	30	1	651	650	9	10	771	725	9	55	894	846	6	54	1747			
	18	166	162	14	18	282	295	18	5	390	398	30	22	528	498	12	42	644	653	16	7	773	763	7	17	877	861	23	39	1741			
	19	162	172	18	8	272	282	28	18	414	413	6	7	511	507	29	33	643	647	17	13	754	746	26	34	895	897	5	3	1741			
	20	175	169	5	11	299	288	1	12	394	400	26	20	538	515	2	25	659	602	1	58	744	736	36	44	890	879	10	21	1735			
	21	166	177	14	3	298	281	2	19	393	408	27	12	503	506	37	34	621	644	39	16	779	740	1	40	839	860	61	40	1735			
	22	165	170	15	10	279	274	21	26	391	398	29	22	521	513	19	27	654	653	6	7	759	755	21	25	849	871	51	29	1783			
	23	164	177	16	3	272	286	28	14	394	392	26	28	517	504	23	36	607	631	53	29	767	754	13	26	874	839	26	61	1777			
	24	168	178	12	2	286	273	14	27	391	395	29	25	497	514	43	26	615	640	45	20	731	736	49	44	848	877	52	23	1771			
	25	175	172	5	8	299	298	1	2	392	411	28	9	539	530	1	10	635	632	25	28	762	740	18	40	859	885	41	15	1765			
	26	174	173	6	7	292	293	8	7	411	408	9	12	504	501	36	39	613	620	47	40	762	759	18	21	858	872	42	28	1758			
	27	162	164	18	16	298	272	2	28	407	399	13	21	524	508	16	32	658	648	2	12	731	775	49	5	855	837	45	63	1755			
	28	179	170	1	10	299	290	1	10	419	408	1	12	510	533	30	7	627	648	3	12	730	737	50	43	846	839	54	61	1747			
	29	179	164	1	16	296	295	4	5	419	418	1	2	522	512	18	28	630	643	30	17	776	771	4	9	835	849	65	51	1741			
	30	179	169	1	11	299	290	1	10	419	416	1	4	528	507	12	33	648	650	12	33	648	650	12	10	779	735	1	45	852	891	48	9

Table 6.5

The peaks and differences for motors A, data D

Pattern	3rdW				5thW				7thW				9thW				11thW				13thW				15thW			
	Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.	
	P11	P12	D11	D12	P21	P22	D21	D22	P31	P32	D31	D32	P41	P42	D41	D42	P51	P52	D51	D52	P61	P62	D61	D62	P71	P72	D71	D72
1	174	168	6	12	297	276	3	24	417	410	3	10	528	502	12	38	654	601	6	59	729	778	51	2	851	855	49	45
2	163	179	17	1	299	297	1	3	417	416	3	4	503	506	37	34	605	629	55	31	750	729	30	51	840	866	60	34
3	179	165	1	15	297	285	3	15	417	416	3	4	497	513	43	27	652	641	8	19	726	723	54	57	867	838	33	62
4	171	169	9	11	295	291	5	9	415	414	5	6	514	525	26	15	606	640	54	20	734	739	46	41	891	828	9	72
5	162	177	18	3	295	294	5	6	415	412	5	8	531	511	9	29	650	606	10	54	777	776	3	4	889	894	11	6
6	179	171	1	9	293	292	7	8	413	418	7	2	536	503	4	37	654	634	6	26	747	769	33	11	856	894	44	6
7	177	167	3	13	293	283	7	17	413	394	7	26	495	529	45	11	623	646	37	14	740	726	40	54	882	889	18	11
8	175	173	5	7	292	290	8	10	412	411	8	9	512	535	28	5	628	644	32	16	767	725	13	55	830	887	70	13
9	162	174	18	6	292	280	8	20	412	413	8	7	516	508	24	32	646	647	14	13	775	766	5	14	853	832	47	68
10	170	163	10	17	290	297	10	3	410	399	10	21	536	512	4	28	640	616	20	44	753	739	27	41	856	894	44	6
11	167	174	13	6	290	289	10	11	410	398	10	22	531	505	9	35	640	618	20	42	739	776	41	4	874	830	26	70
12	169	179	11	1	288	289	12	11	408	409	12	11	498	496	42	44	618	653	42	7	738	740	42	40	895	897	5	3
13	169	172	11	8	289	288	11	12	408	409	12	11	513	522	27	18	645	619	15	41	775	750	5	30	841	853	59	47
14	166	179	14	1	287	285	13	15	407	410	13	10	497	512	43	28	633	628	27	32	764	742	16	38	861	891	39	9
15	175	172	5	8	287	273	13	27	407	416	13	4	507	515	33	25	652	642	8	18	730	764	50	16	891	856	9	44
16	176	165	4	15	285	286	15	14	405	406	15	14	536	499	4	41	627	617	33	43	732	761	48	19	829	828	71	72
17	176	178	4	2	285	286	15	14	405	393	15	27	524	525	16	15	629	640	31	20	761	747	19	33	896	840	4	60
18	168	176	12	4	283	279	17	21	403	419	17	522	514	18	26	628	634	32	26	737	735	43	45	849	884	51	16	
19	178	177	2	3	283	273	17	27	403	393	17	27	536	514	4	26	604	623	56	37	744	770	36	10	853	869	47	31
20	166	176	14	4	281	282	19	18	401	402	19	18	502	504	38	36	636	612	24	48	736	756	44	24	840	871	60	29
21	167	176	13	4	281	282	19	18	401	402	19	18	519	539	21	1	636	624	24	36	769	725	11	55	897	856	3	44
22	174	163	6	17	298	272	2	28	417	403	3	17	535	499	5	41	628	651	32	9	757	761	23	19	881	880	19	20
23	171	162	9	18	293	282	7	18	413	395	7	25	495	535	45	5	641	657	19	3	738	751	42	29	837	882	63	18
24	179	169	1	11	299	281	1	19	409	392	11	28	512	523	28	17	608	609	52	51	727	728	53	52	841	843	59	57
25	179	176	1	4	285	284	15	16	405	404	15	16	497	513	43	27	617	624	43	36	754	736	26	44	874	838	26	62
26	165	173	15	7	282	272	18	28	402	404	18	16	532	525	8	15	613	653	47	7	738	741	42	39	853	855	47	45
27	163	179	17	1	299	297	1	3	417	416	3	4	503	506	37	34	605	629	55	31	750	729	30	51	840	866	60	34
28	179	165	1	15	297	285	3	15	417	416	3	4	497	513	43	27	652	641	8	19	726	723	54	57	867	838	33	62
29	171	169	9	11	295	291	5	9	415	414	5	6	514	525	26	15	606	640	54	20	734	739	46	41	891	828	9	72
30	162	177	18	3	295	294	5	6	415	412	5	8	531	511	9	29	650	606	10	54	777	776	3	4	889	894	11	6
31	179	171	1	9	293	292	7	8	413	418	7	2	536	503	4	37	654	634	6	26	747	769	33	11	856	894	44	6
32	175	173	5	7	292	290	8	10	412	411	8	9	512	535	28	5	628	644	32	16	767	725	13	55	830	887	70	13
33	162	174	18	6	292	280	8	20	412	413	8	7	516	508	24	32	646	647	14	13	775	766	5	14	853	832	47	68
34	170	163	10	17	290	297	10	3	410	399	10	21	536	512	4	28	640	616	20	44	753	739	27	41	856	894	44	6

(continued on following page)

Table 6.5 (continued)

Pattern	3rdW				5thW				7thW				9thW				11thW				13thW				15thW					
	Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.		Peaks		Diff.			
	P11	P12	D11	D12	P21	P22	D21	D22	P31	P32	D31	D32	P41	P42	D41	D42	P51	P52	D51	D52	P61	P62	D61	D62	P71	P72	D71	D72	d	
training	35	167	174	13	6	290	289	10	11	410	398	10	22	531	505	9	35	640	618	20	42	739	776	41	4	874	830	26	70	1751
	36	169	179	11	1	288	289	12	11	408	409	12	11	498	496	42	44	618	653	42	7	738	740	42	40	895	897	5	3	1742
	37	169	172	11	8	289	288	11	12	408	409	12	11	513	522	27	18	645	619	15	41	775	750	5	30	841	853	59	47	1742
	38	166	179	14	1	287	285	13	15	407	410	13	10	497	512	43	28	633	628	27	32	764	742	16	38	861	891	39	9	1734
	39	175	172	5	8	287	273	13	27	407	416	13	4	507	515	33	25	652	642	8	18	730	764	50	16	891	856	9	44	1734
	40	176	165	4	15	285	286	15	14	405	406	15	14	536	499	4	41	627	617	33	43	732	761	48	19	829	828	71	72	1726
	41	176	178	4	2	285	286	15	14	405	393	15	27	524	525	16	15	629	640	31	20	761	747	19	33	896	840	4	60	1726
testing	42	168	176	12	4	283	279	17	21	403	419	17	1	522	514	18	26	628	634	32	26	737	735	43	45	849	884	51	16	1715
	43	178	177	2	3	283	273	17	27	403	393	17	27	536	514	4	26	604	623	56	37	744	770	36	10	853	869	47	31	1715
	44	179	176	1	4	285	284	15	16	405	404	15	16	497	513	43	27	617	624	43	36	754	736	26	44	874	838	26	62	1723
	45	165	173	15	7	282	272	18	28	402	404	18	16	532	525	8	15	613	653	47	7	738	741	42	39	853	855	47	45	1713
	46	167	178	13	2	279	298	21	2	399	408	21	12	534	506	6	34	657	651	3	9	771	773	9	7	893	898	7	2	1695
	47	172	175	8	5	278	281	22	19	398	403	22	17	496	495	44	45	616	633	44	27	764	754	16	26	886	894	14	6	1694
	48	176	167	4	13	295	296	5	4	415	407	5	13	525	527	15	13	635	629	25	31	760	755	20	25	891	838	9	62	1774
testing	49	174	173	6	7	293	291	7	9	413	409	7	11	515	502	25	38	605	646	54	14	726	732	54	48	846	894	54	6	1765
	50	169	163	11	17	290	299	10	1	410	417	10	3	521	534	19	6	641	634	19	26	729	745	51	35	889	827	11	73	1751
	51	171	174	9	6	287	286	13	14	407	406	13	14	508	522	32	18	636	604	24	56	743	727	37	53	869	870	31	30	1734
	52	178	174	2	6	285	286	15	14	405	409	15	11	529	512	11	28	619	624	41	36	775	765	5	15	866	860	34	40	1726
	53	165	179	15	1	283	282	17	18	403	402	17	18	527	510	13	30	624	618	36	42	766	737	14	43	895	885	5	15	1715
	54	168	162	12	18	281	272	19	28	401	402	19	18	496	504	44	36	616	621	44	39	741	755	39	25	876	895	24	5	1707
	55	163	166	17	14	299	276	1	24	417	419	3	1	517	537	23	3	625	651	35	9	721	731	59	49	862	826	38	74	1783
testing	56	173	174	7	6	286	277	14	23	406	407	14	13	513	537	27	3	649	648	11	12	775	734	5	46	860	894	40	6	1732
	57	174	173	6	7	293	291	7	9	413	409	7	11	515	502	25	38	605	646	55	14	726	732	54	48	846	894	54	6	1765
	58	176	170	4	10	292	299	8	1	412	411	8	9	503	536	37	4	648	659	12	1	723	728	57	52	831	883	69	17	1759
	59	169	163	11	17	290	299	10	1	410	417	10	3	521	534	19	6	641	634	19	26	729	745	51	35	889	827	11	73	1751
	60	166	164	14	16	289	288	11	12	409	408	11	12	512	529	28	11	625	630	35	30	774	730	6	50	881	869	19	31	1742
	61	171	174	9	6	287	286	13	14	407	406	13	14	508	522	32	18	636	604	24	56	743	727	37	53	869	870	31	30	1734
	62	178	174	2	6	285	286	15	14	405	409	15	11	529	512	11	28	619	624	41	36	775	765	5	15	866	860	34	40	1726
testing	63	165	179	15	1	283	282	17	18	403	402	17	18	527	510	13	30	624	618	36	42	766	737	14	43	895	885	5	15	1715
	64	168	162	12	18	281	272	19	28	401	402	19	18	496	504	44	36	616	621	44	39	741	755	39	25	876	895	24	5	1707
	65	163	166	17	14	299	276	1	24	417	419	3	1	517	537	23	3	625	651	35	9	721	731	59	49	862	826	38	74	1783
	66	173	174	7	6	286	277	14	23	406	407	14	13	513	537	27	3	649	648	11	12	775	734	5	46	860	894	40	6	1732
	67	173	174	7	6	281	280	19	20	401	400	19	20	537	520	3	20	618	645	42	15	721	729	59	51	886	895	14	5	1703

CHAPTER 7

CONCLUSION

7.1 Discussion

First, it has to be stated that the ANN method was proposed to be fed by a certain peak from each window, which were from 3rdW to 15thW. That was the reason to choose all the windows for the ANN final design of each data, as illustrated in Figures 6.7 to 6.20. Thus, the tracking algorithm was applying on all the windows because the location of the saliency harmonics is usually unknown, and it might be in any window between 3rdW to 15thW.

The argument that says a proportional function can be work better than the proposed method. It could be right only if the location of the saliency harmonics and the parameter of Equation 3.9 were known. In other words, the location of the saliency harmonics might not be at the first peak or even the second peak that was extracted from a window. By looking back at Table 6.1 of pattern 24 in Table 6.3, the first peak of the 5thW was 299 Hz. Then, the speed value could be calculated by using Equation 3.9, where k=1, and ($f_{ecc} + 60$) because the harmonic was extracted from 5thW. Also, the number of rotor slots (R) was equal to 12.

$$N_r = \frac{60}{1 * 12} (299 + 60) = 1795 RPM$$

From the above calculation, which would be utilized in the proportional function, the actual speed was 1747 RPM while the calculated speed was 1795. The difference between the

actual and calculated speed is 48 RPM, which is a very large error. This result might happen again with another pattern if the 5thW is considered as the main location of the saliency harmonics of motor A. By taking another example, see pattern 12 in Table 6.3, the first peak of 5thW was 288 Hz, so by using Equation 3.9 again, the calculated speed is 1740 RPM while the actual speed is 1742 RPM. The difference, which equals 2 RPM, is small. However, the proposed method could be a better solution instead of using the proportional function because the locations of the saliency harmonics and the parameters of the system are not required to be known. On the other hand, the question that might probably be raised is, "Does the estimated speed by ANN change with age?" The answer is no, because the relation between the saliency harmonics and the rotor speed does not rely on time, as proven in Equations 3.4 to 3.9.

It was proved that ANNs succeeded to estimate the rotor speed of the WRIMs but it failed with the SCIM. All the results of Figures 6.5 to 6.17 demonstrate that the rotor speed of WRIMs can be estimated successfully by utilizing two hidden layers and the backpropagation algorithm. Single hidden layer ANNs could predict the speed of motor A and B accurately by placing the windows at 5th and 7th harmonics. Otherwise, single hidden layer ANNs would fail if all the windows are applied. Therefore, when all windows were placed (3rdW-15thW), two hidden layers had to be set instead of one hidden layer in order to increase the power of ANNs prediction. That does not mean that three hidden layers or more are going to provide more performance because "empirical studies often found that deep networks generally performed no better, and often worse, than neural networks with one or two hidden layers" [43].

The sensorless speed estimation of the WRIMs could not be accomplished if the tracking algorithm that was proposed in Chapter 5 had not succeeded. Thus, the desired outputs of data A,

A+B, and D were estimated precisely with two peaks from each window, where the testing average error less than 2 RPM; the ANN results and details are seen in Figures 6.10, 6.16, and 6.22. The reasons behind this success is that motors A and B where data A, B, and D were taken from have the saliency harmonics inside the windows from 3rd to 15th.

On the other hand, the tracking algorithm unfortunately could not provide the saliency harmonics of the SCIM. The investigation of the testing patterns of motor C, which represent SCIMs, shows a very poor accuracy where their testing average errors exceed 5 RPM. This due to that the saliency harmonics of motor C do not appear as peaks. The other possibility is that the saliency harmonics might not exist in the windows from 3rd to 15th in case the number of rotor slots (R) of motor C is bigger than 32 because of the positive correlation between R and the saliency harmonics. Therefore, by using Equation 3.9 in the calculations below and assuming Nr is the synchronous speed, 1800 RPM, the saliency harmonics of motor C would be higher than the Nyquist frequency, 1000 Hz, which means they cannot be extracted.

$$N_r = \frac{60}{KR} (f_{ecc} \pm f_e)$$

$$1800 = \frac{60}{1 * 32} (f_{ecc} \pm 60)$$

$$f_{ecc} = 1020, \text{ or } 900 \text{ Hz}$$

This problem might be solved by increasing the samples of the signals which in turn will raise the Nyquist frequency. Otherwise, the tracking algorithm and ANN design possibly needs some developments where the saliency harmonics of SCIMs can be recognized. Then, the

sensorless speed estimation of motor C hopefully will be better than what was achieved in this paper. There are other reasons of the untraceable saliency harmonics, some of which might be using inefficient tools and having high levels of noise. Furthermore, the delay time of the data A, B, and C were one second, which is very long in a control system. Thus, data D were examined with T_m equals 500 m second, and it was found that the testing average error was still very good accuracy, 1.6 RPM. Therefore, the delay time could not be a serious issue that is related to the proposed method.

7.2 Summary

There is a tremendous amount of effort being focused on finding an alternative technique to replace the rotational transducer in ACIMs. The reason is that these motors consume massive amounts of electric power, so the sensorless speed method will increase ACIMs' reliability, which in turn saves so much energy and money. These researches [1-43] represent and investigate a variety of sensorless methods, which were reviewed in this paper. The conventional non-intelligent methods need to measure the voltage and current simultaneously and all parameters of the motor must be known in order to measure the rotor speed. The artificial intelligence method does not require the system equations and parameters. On the other hand, ANNs need a collection of data because they learn from previous experiences as shown in this paper.

The results of the lab experiment confirmed that ANNs can be successfully applied to measure the rotor speed of WRIMs instead of the rotor transducer. That is based on extracting a set of stator current frequencies that hopefully contain speed-dependent components, saliency

harmonics. ANNs were proved to accurately predict the rotor speed of WRIMs, while the ANNs estimation was difficult for SCIMs. Therefore, the system of selecting ANN inputs needs some development, so the saliency harmonics of SCIMs can be detected easily.

7.3 Conclusion

The aim of this paper was to develop a sensorless speed estimation of IMs by utilizing the artificial neural network. In order to evaluate this method, three ACIMs were used to collect data from using the lab equipment. These motors were two identical WRIMs, 0.25 H.P, and SCIM, 1/3 H.P. The collected data were analyzed to extract a set of peaks from the stator current spectrum, which was based on the tracking algorithm that was built especially for this method.

These peaks were trained by using the backpropagation algorithm, where the desired output was the rotor speed that was measured simultaneously with the stator current. The examined data were collected into four groups: data A, A+B, C, and D. Data A and D were taken from the same WRIM while data B was taken from the other identical WRIM. Data C was obtained from the SCIM. All these data had the same acquisition time, which was one second except data D, which was half a second. The best testing average errors that were achieved are 1 RPM for data A, 1.5 RPM for data A+B, 5.7 RPM for data C, and 1.6 RPM for data D. Thus, the experimental results show that the proposed method is an appropriate approach to estimate the rotor speed of WRIMs, while this method might need some developments to be workable with SCIMs. Eventually, reducing the overall system costs and improving the reliability of the drive system can be achieved by the proposed method.

7.4 Future Work

As the proposed method needs to be in real-time application, there has to be a master device to receive the stator current signals, run a FFT algorithm, and implement an artificial neural network. Therefore, FPGA board is the appropriate device for the parallel processing. The easiest way is to use a computer to do the training and testing processes as done in this paper. Then, after achieving a precise result, the eventual weights and the required stages, the FFT and tracking algorithm, can be implemented in a structure as illustrated in the flowchart of the proposed method, see Figure 5.6.

REFERENCES

- [1] Alsofyani, Ibrahim M., and Nik Rumzi Nik Idris. "A review on sensorless techniques for sustainable reliability and efficient variable frequency drives of induction motors." *Renewable and Sustainable Energy Reviews* 24 (2013): 111-121.
- [2] Zhao, Tong, and Donald S. Zinger. "Induction motor speed detection by applying Goertzel's algorithm to current harmonics." In *Applied Power Electronics Conference and Exposition, 2000. APEC 2000. Fifteenth Annual IEEE*, vol. 1, pp. 140-143. IEEE, 2000.
- [3] Zatocil, H. "Physical understanding of multiple saliences in induction motors and their impact on sensorless control." In *Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008. International Symposium on*, pp. 1503-1508. IEEE, 2008.
- [4] V. Damin Door. "Speed Detection for Induction Motors Using Motor Current Signature Analysis." *International Journal of Engineering Research & Technology, IJERT*, Nov. 2013, pp. 3599-3601.
- [5] Rață, Gabriela, Mihai Rață, Iulian Graur, and Dan L. Milici. " Induction Motor Speed Estimator Using Rotor Slot Harmonics".*" Advances in Electrical and Computer Engineering* 9, no. 1 (2009): 70-73.
- [6] Blasco Giménez, Ramón. "High performance sensorless vector control of induction motor drives." PhD diss., University of Nottingham, 1995.
- [7] Chen, Zhiguo, Xianming Deng, Kun Huang, Wenhuan Zhen, and Lei Wang. "Sensorless control of wound rotor synchronous machines based on high-frequency signal injection into the stator windings." *Journal of Power Electronics* 13, no. 4 (2013): 669-678.
- [8] Gao, Zhi, Larry Turner, Roy S. Colby, and Benoit Leprette. "A frequency demodulation approach to induction motor speed detection." *Industry Applications, IEEE Transactions on* 47, no. 4 (2011): 1632-1642.
- [9] Walivadekar, V. N., V. Sarat Babu, and C. Aparna. "Sensorless electronic tachometer for squirrel cage induction motor." In *Power Electronics, Drives and Energy Systems for Industrial Growth, 1996., Proceedings of the 1996 International Conference on*, vol. 1, pp. 67-73. IEEE, 1996.
- [10] Giri, Fouad, ed. *AC electric motors control: Advanced design techniques and applications*. John Wiley & Sons, 2013.
- [11] Kubota, Hisao, and Kouki Matsuse. "Speed sensorless field-oriented control of induction motor with rotor resistance adaptation." *Industry Applications, IEEE Transactions on* 30, no. 5 (1994): 1219-1224.

- [12] Abu-Rub, Haitham, Atif Iqbal, and Jaroslaw Guzinski, eds. High performance control of AC drives with MATLAB/Simulink models. John Wiley & Sons, 2012.
- [13] Gao, Wei, and Zhirong Guo. "Speed Sensorless Control of PMSM using Model Reference Adaptive System and RBFN." *Journal of Networks* 8, no. 1 (2013): 213-220.
- [14] Rambetius, Alexander, Sven Luthardt, and Bernhard Piepenbreier. "Speed estimation and compensation for harmonics in sensorless wound rotor synchronous machines." In *Sensorless Control for Electrical Drives (SLED)*, 2014 IEEE 5th International Symposium on, pp. 1-8. IEEE, 2014.
- [15] Abu-Rub, Haitham, Mariusz Malinowski, and Kamal Al-Haddad. Power electronics for renewable energy systems, transportation and industrial applications. John Wiley & Sons, 2014.
- [16] Karanayil, Baburaj, Muhammed Fazlur Rahman, and Colin Grantham. "Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive." *Industrial Electronics, IEEE Transactions on* 54, no. 1 (2007): 167-176.
- [17] Gadoue, Shady M., Damian Giaouris, and John W. Finch. "Low speed operation improvement of MRAS sensorless vector control induction motor drive using neural network flux observers." In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pp. 1212-1217. IEEE, 2006.
- [18] Gadoue, Shady M., Damian Giaouris, and John W. Finch. "Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers." *Industrial Electronics, IEEE Transactions on* 56, no. 8 (2009): 3029-3039.
- [19] Nandi, Subhasis, Shehab Ahmed, and Hamid Toliyat. "Detection of rotor slot and other eccentricity related harmonics in a three phase induction motor with different rotor cages." *Energy Conversion, IEEE Transactions on* 16, no. 3 (2001): 253-260.
- [20] Kubota, Hisao, and Kouki Matsuse. "Speed sensorless field oriented control of induction machines using flux observer." In *IECON-PROCEEDINGS-*, pp. 1611-1611. Institute of Electrical and Electronics Engineers INC (IEEE), 1994.
- [21] Bodson, Marc, and John Chiasson. "A comparison of sensorless speed estimation methods for induction motor control." In *Proceedings of the American control conference*, vol. 4, pp. 3076-3081. 2002.
- [22] Schauder, Colin. "Adaptive speed identification for vector control of induction motors without rotational transducers." *Industry applications, IEEE Transactions on* 28, no. 5 (1992): 1054-1061.
- [23] Fakhfakh, Tahar, Walter Bartelmus, Fakher Chaari, Radoslaw Zimroz, and Mohamed Haddar. "Condition Monitoring of Machinery in Non-Stationary Operations." In *Proceedings of the Second International Conference Condition Monitoring of Machinery in Non-Stationary Operations CMMNO'2012*. 2012.
- [24] Kumar, PNH Phanindra, D. M. Deshpande, and Manisha Dubey. "Model Reference Adaptive System (MRAS) Based Speed Sensorless Vector Control of Induction Motor Drive."
- [25] Zhen, Lu, and Longya Xu. "Sensorless field orientation control of induction machines based on a mutual MRAS scheme." *Industrial Electronics, IEEE Transactions on* 45, no. 5 (1998): 824-831.

- [26] Petrovic, Goran, Tomislav Kilic, and Bozo Terzic. "Sensorless speed detection of squirrel-cage induction machines using stator neutral point voltage harmonics." *Mechanical systems and signal processing* 23, no. 3 (2009): 931-939.
- [27] Degner, Michael W., and Robert D. Lorenz. "Using multiple saliencies for the estimation of flux, position, and velocity in AC machines." *Industry Applications, IEEE Transactions on* 34, no. 5 (1998): 1097-1104.
- [28] Degner, Michael W., and Robert D. Lorenz. "Position estimation in induction machines utilizing rotor bar slot harmonics and carrier-frequency signal injection." *Industry Applications, IEEE Transactions on* 36, no. 3 (2000): 736-742.
- [29] Cusido, Jordi, Luis Romeral, Juan Ortega, Javier Rosero, and Antonio Garcia Espinosa. "Fault detection in induction machines using power spectral density in wavelet decomposition." *Industrial Electronics, IEEE Transactions on* 55, no. 2 (2008): 633-643.
- [30] Rambetius, Alexander, and Bernhard Piepenbreier. "Comparison of carrier signal based approaches for sensorless wound rotor synchronous machines." In *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2014 International Symposium on*, pp. 1152-1159. IEEE, 2014.
- [31] Djurović, S., and S. Djukanović. "A sensorless speed estimation method for wound rotor induction machine." In *Electric Machines & Drives Conference (IEMDC), 2011 IEEE International*, pp. 878-883. IEEE, 2011.
- [32] Drif, M'hamed, and AJ Marques Cardoso. "Airgap-eccentricity fault diagnosis, in three-phase induction motors, by the complex apparent power signature analysis." *IEEE transactions on industrial electronics* 55, no. 3 (2008): 1404-1410.
- [33] Antonino-Daviu, Jose, Vicente Climente-Alarcon, I. Tsoumas, George Georgoulas, and R. B. Perez. "Multi-harmonic tracking for diagnosis of rotor asymmetries in wound rotor induction motors." In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pp. 5555-5560. IEEE, 2013.
- [34] Jansen, Patrick L., and Robert D. Lorenz. "Transducerless position and velocity estimation in induction and salient AC machines." *Industry Applications, IEEE Transactions on* 31, no. 2 (1995): 240-247.
- [35] Williamson, S., and S. Djurović. "Origins of stator current spectra in DFIGs with winding faults and excitation asymmetries." In *Electric Machines and Drives Conference, 2009. IEMDC'09. IEEE International*, pp. 563-570. IEEE, 2009.
- [36] Tshiloz, K., D. S. Vilchis-Rodriguez, and S. Djurović. "Investigation of harmonic emissions in wound rotor induction machines." (2014): 2-3.
- [37] Fuchs, Ewald, and Mohammad AS Masoum. *Power Quality in Power Systems and Electrical Machines*. Academic Press, 2015.
- [38] Keysan, Ozan, and Bülent H. Ertan. "Determination of rotor slot number of an induction motor using an external search coil." *Facta universitatis-series: Electronics and Energetics* 22, no. 2 (2009): 227-234.

[39] Rezig, Ali, Mohamed Rachid Mekideche, and Abdesselem Djerdir. "Effect of rotor eccentricity faults on noise generation in permanent magnet synchronous motors." *Progress In Electromagnetics Research C* 15 (2010): 117-132.

[40] Djurović, S., and S. Williamson. "Influence of supply harmonic voltages on DFIG stator current and power spectrum." In *Electrical Machines (ICEM), 2010 XIX International Conference on*, pp. 1-6. IEEE, 2010.

[41] Franco, Leonardo, and José M. Jerez, eds. *Constructive neural networks*. Vol. 258. Springer, 2009.

[42] Kriesel, David. "A brief introduction to neural networks." Retrieved August 15 (2007): 2011.

[43] Bengio, Yoshua, and Yann LeCun. "Scaling learning algorithms towards AI." *Large-scale kernel machines* 34, no. 5 (2007).

APPENDIX
THE MATLAB CODE OF THE PROPOSED METHOD

```
%%%%% All the code and data were taken and written by: Abdulelah Alkhoraif (eng.v400@gmail.com) , Oct. 2015 %%%%
%%% After downloading the file (MS_thesis_Abdulelah_2015_data_and_code), make sure to move all %%%%
%%% the collecting data: (WRIMallDATA30patternsMotorA.mat),(WRIMallDATA40patternsMotorAandB.mat), %%%%
%%% (SCIMallDATA30patternsMotorC.mat), and ('DATAmotorA68patterns500msecond.mat') to (MATLAB) file in %%%%
%%% your computer. Code for Two Hidden Layers %%%%
%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%% %%%%%%
```

```
%% 1- import just one of the following data %%
```

```
AAW=importdata('WRIMallDATA30patternsMotorA.mat'); % This is data A, wound rotor induction motor, 30 patterns, 1 second.
%AAW=importdata('WRIMallDATA40patternsMotorAandB.mat'); %This is data A and B, wound rotor induction motors, 40 patterns, 1 second
%AAW=importdata('SCIMallDATA30patternsMotorC.mat'); % squirrel cage induction motor, 30 patterns, 1 second.
%AAW=importdata('DATAmotorA68patterns500msecond.mat');
```

```
format long
[m,L] = size(AAW);
AAZ= AAW(:,1:L-1);
AAQ=AAW(:,L);
```

```
[m,L] = size(AAZ);
T = 1; %
Fa= 60; % the fundamental frequency

%%%%% set up the windows%%%%%
%%%%% set up the windows%%%%%
%%%%% set up the windows%%%%%
```

```
Fbr= 17; % this is for 3rdW
Fcr= 29; % this is for 5thW
Fdr= 29; % this is for 7thW
Fer= 44; % this is for 9thW
Ffr= 58; % this is for 11thW
Fgr= 58; % this is for 13thW
Fhr= 73; % this is for 15thW
```

```
% you can change these windows as you like, remember that optimal widths are :
% Fbr= 17; Fcr= 29; Fdr= 29; Fer= 44; Ffr= 58; Fgr= 58; Fhr= 73; see Table 5.4.
```

```
%%How many peaks for each harmonics you want as inputs for ANN?
pi=2;
```

```
%%%%% set up the ANN%%%%%
%%%%% set up the ANN%%%%%
%%%%% set up the ANN%%%%%
```

```
hidden_neurons = 22;
se_hidden_neurons= 7;
epochs = 4000;
```

```
%%%%%%%%%%%%%
%%%%%%%%% START%%%%%%%%%
%%%%%%%%%%%%%
```

%% The following steps are for the FFT and the tracking algorithm %%

```
% make loop from 1 to m
for i=1:m;
```

```
Volt=AAZ(i,:);
L=length(Volt);
N=L;
Fs=N/T;
e=1/Fs;
t=[0:e:T-e];
nfft=L; % some time it is chosen to be 1024;
Yk=fft(Volt,nfft);
Yk=Yk(1:nfft/2);
f=[0:1:nfft/2-1]*Fs/nfft;
absYk = abs(Yk);
magYk1 = abs(Yk);
%plot(f, magYk1); % stem(f, absYk);
% tracking Algorithm
[Ma,Ia] = max(magYk1);
Fa=Ia-1;
Fb=3 * Fa;
Fc=5 * Fa;
Fd=7 * Fa;
Fe=9 * Fa;
Ff=11 * Fa;
Fg=13 * Fa;
Fh=15 * Fa;
X= magYk1';
%=====
Fbl= Fb - Fbr;
Fbl = floor(Fbl);
Fbh = Fb+1;
Xb = X(Fbl:Fbh);
[Bb,Ib]=sort(Xb);
Ib=flipud(Ib);
Ib = (Fbl-2)+Ib;
Ib=Ib(Ib~=Fb);
Ib=Ib';
%=====
Fcl = Fc - Fer;
Fcl = floor(Fcl);
Fch = Fc+1;
Xc = X(Fcl:Fch);
[Bc,Ic]=sort(Xc);
Ic=flipud(Ic);
Ic = (Fcl-2)+Ic;
Ic=Ic(Ic~=Fc);
Ic=Ic';
%=====
Fdl = Fd-Fdr;
Fdh= Fd+1;
Xd = X(Fdl: Fdh);
[Bd,Id]=sort(Xd);
Id=flipud(Id);
Id = (Fdl-2)+Id;
Id=Id(Id~=Fd);
Id=Id';
%=====
Fel = Fe-Fer;
Feh= Fe+1;
```

```

Xe = X(Fel: Feh);
[Be,Ie]=sort(Xe);
Ie=flipud(Ie);
Ie = (Fel-2)+Ie;
Ie=Ie(Ie~=Fe);
Ie=Ie';
%=====
Ffl = Ff-Ffr;
Ffh= Ff+1;
Xf = X(Ffl: Ffh);
[Bf,If]=sort(Xf);
If=flipud(If);
If = (Ffl-2)+If;
If=If(If~=Ff);
If=If';
%=====
Fgl = Fg-Fgr;
Fgh= Fg+1;
Xg = X(Fgl: Fgh);
[Bg,Ig]=sort(Xg);
Ig=flipud(Ig);
Ig = (Fgl-2)+Ig;
Ig=Ig(Ig~=Fg);
Ig=Ig';
%=====
Fhl = Fh-Fhr;
Fhh= Fh+1;
Xh = X(Fhl: Fhh);
[Bh,Ih]=sort(Xh);
Ih=flipud(Ih);
Ih = (Fhl-2)+Ih;
Ih=Ih(Ih~=Fh);
Ih=Ih';
%=====

```

```

Ib = Ib(1:pi);
Ib = Fb - Ib;

```

```

Ic = Ic(1:pi);
Ic = Fc - Ic;

```

```

Id = Id(1:pi);
Id = Fd - Id;

```

```

Ie = Ie(1:pi);
Ie = Fe - Ie;

```

```

If = If(1:pi);
If = Ff - If;

```

```

Ig = Ig(1:pi);
Ig = Fg - Ig;

```

```

Ih = Ih(1:pi);
Ih = Fh - Ih;

```

```
%=====
```

```

AT=[Ib Ic Id Ie If Ig Ih];
AAT(i,:)=AT;
end

%=====
Data_all=[AAT AAQ];
%Data_all=flipud(Data_all);
%=====

[m,L] = size(Data_all);

Mtest=floor(0.3*m);
Mtrain=m - Mtest;

AAx=zeros(Mtrain,L); %train data
AAy=zeros(Mtest,L); %test data

%=====

p=1;
for i=1: Mtest;
Di(i)=p+3;
p=Di(i);
end

%=====

j=1;
z=1;
for i=1: m;
    if any(Di==i)
        AAy(z,:)= Data_all(i,:);
        z=z+1;
    else
        AAx(j,:)=Data_all(i,:);
        j=j+1;
    end
end

%%%%%%%
%%% ANN Code %%%
%%%%%%%
% Now we can implement ANN

data_in=AAx(:,1:L-1);
dtat_out= AAx(:,L);
data_in=data_in';
dtat_out=dtat_out';
Input_A=data_in;
Output_A=dtat_out;

test_data_in= AAy(:,1:L-1);
test_dtat_out= AAy(:,L);

```

```

min_Input_A=0; % assuming the minimum inputt
max_Input_A=Fhr+15; % assuming the maximum input change it to 50 for squirrel cage data
new_min_input= 0;
new_max_input= 1;
slop_input=(new_max_input-new_min_input)/(max_Input_A-min_Input_A);
c_input= new_min_input-(slop_input*min_Input_A);

% now we are doing normalization for the output data
min_output_A=0; % assuming the minimum output
max_output_A=1900; % assuming the maximum output
new_min_output= -1;
new_max_output= 1;
slop_output=(new_max_output-new_min_output)/(max_output_A-min_output_A);
c_ouput= new_min_output-(slop_output* min_output_A);

%===== thirded stage
% now we find the new input and output after normalization
new_Input_A= slop_input.*Input_A+c_input;
new_output_A=slop_output.*Output_A+c_ouput;

%===== fourth stage
% train the data
train_inp=new_Input_A';
train_out=new_output_A';

patterns = size(train_inp,1);
%add a bias as an input
bias = ones(patterns,1);
train_inp = [train_inp bias];
inputs = size(train_inp,2); % this is the number of input

Test=test_data_in;
K=size(Test);
K=K(1,1);
Test_new=slop_input.* Test +c_input;
bias = ones(K,1);
Test_new=[Test_new bias];

%add button for early stopping
hstop = uicontrol('Style','PushButton','String','Stop', 'Position', [5 5 70 20],'callback','earlystop = 1;');
earlystop = 0;

%add button for resetting weights
hreset = uicontrol('Style','PushButton','String','Reset Wts', 'Position', get(hstop,'position')+[75 0 0 0],'callback','reset = 1;');
reset = 0;

%add slider to adjust the learning rate
hlr = uicontrol('Style','slider','value',.1,'Min',.01,'Max',1,'SliderStep',[0.01 0.1],'Position', get(hreset,'position')+ [75 0 100 0]);

```

```
% ----- set weights -----
%set initial random weights
N_y=1; % output neurons
weight_input_hidden = (randn(hidden_neurons,inputs) - 0.5)/10;
weight_se_input_hidden = (randn(se_hidden_neurons,hidden_neurons) - 0.5)/10;
weight_hidden_output = (randn(N_y,se_hidden_neurons) - 0.5)/10;

for iter = 1:epochs

    %get the learning rate from the slider
    alr = 1;    % learning rate for hidden layers
    blr = 0.02; % learning rate for output layer

    %loop through the patterns, selecting randomly
    for j = 1:patterns

        %set the current pattern
        this_pat = train_inp(j,:);
        act = train_out(j,1);

        Fi_hval = (tanh(this_pat*weight_input_hidden'));
        hval = (tanh(Fi_hval* weight_se_input_hidden'));
        pred = hval*weight_hidden_output';
        error = pred - act;

        % adjust weight hidden - output
        delta_HO = error.*(1);
        delta_se_IO = ((delta_HO*weight_hidden_output).*(1-(hval.^2)));
        delta_IH= ((delta_se_IO* weight_se_input_hidden).*(1-(Fi_hval.^2)));

        weight_hidden_output = weight_hidden_output - blr.*(delta_HO* hval);
        weight_se_input_hidden = weight_se_input_hidden - (alr.* (delta_se_IO'* Fi_hval));
        % adjust the weights input - hidden

        weight_input_hidden = weight_input_hidden - (alr.* (delta_IH'*this_pat));

    end
    % -- another epoch finished

    %plot overall network error at end of each epoch
    pred = (tanh((tanh(train_inp*weight_input_hidden))* weight_se_input_hidden))*weight_hidden_output';
    error = pred - train_out;
    err(iter) = (sum(error.^2))^0.5;

    figure(1);
    plot(err)

    %reset weights if requested
    if reset
        weight_input_hidden = (randn(hidden_neurons,inputs) - 0.5)/10;
        weight_se_input_hidden = (randn(se_hidden_neurons,hidden_neurons) - 0.5)/10;
        weight_hidden_output = (randn(N_y,se_hidden_neurons) - 0.5)/10;
```

```

fprintf('weights reaset after %d epochs\n',iter);
reset = 0;
end

%stop if requested
if earlystop
    fprintf('stopped at epoch: %d\n',iter);
    break
end

%stop if error is small
if err(iter) < 0.002
    fprintf('converged at epoch: %d\n',iter);
    break
end

%add a condition that measure the error of test data
    ANN_output_pred = (pred - c_ouput)/slop_output;
    Diff=dtat_out' -ANN_output_pred;
    errr(iter) = (sum(Diff.^2))^.5;

    ANN_output_Test = (tanh((tanh(Test_new*weight_input_hidden))* weight_se_input_hidden'))*weight_hidden_output';
    ANN_output_Test = (ANN_output_Test - c_ouput)/slop_output;
    test_diff= test_dstat_out - ANN_output_Test;
    errt(iter) = (sum(test_diff.^2))^.5;

%stop if vadilation error is small
vs=0;
if errt(iter) < 7 % 8.5 for 3 RPM Diff, or 5.6 for 2 RPM Diff
    if errr(iter) < 7
        vs=1;
    end
end

if vs==1
    fprintf('converged at epoch: %d\n',iter);
break
end

%let vi=0;
vi=0;

if (mean(abs(test_diff))) < 1.9
    if (mean(abs(Diff))) < 1.5
        vi=1;
    end
end

if vi==1
    fprintf('stopped at epoch: %d\n',iter);
    break
end

end

pred = (tanh((tanh(train_inp*weight_input_hidden))* weight_se_input_hidden'))*weight_hidden_output';
ANN_output_pred = (pred - c_ouput)/slop_output;
Diff=dtat_out' -ANN_output_pred;

```

```
ANN_output_Test = (tanh((tanh(Test_new*weight_input_hidden'))*  
weight_se_input_hidden'))*weight_hidden_output';  
ANN_output_Test = (ANN_output_Test - c_ouput)/slop_output;  
test_diff= test_dstat_out - ANN_output_Test;  
  
AAARr=[dstat_out' ANN_output_pred];  
AAATr=[test_dstat_out ANN_output_Test];  
  
P1= scatter(dstat_out, Diff,'b','LineWidth',1.2);  
hold on  
P2=scatter(test_dstat_out, test_diff,'r','LineWidth',1.2);  
  
legend([P1,P2],' Error from Training Data',' Error from Testing Data');
```