

Speed Control Of A Three Phase Induction Motor Using Field Oriented Control

Eng. Mohamed Hussein, Dr. Mahfooz E. Shalaby, Dr. Hamdy A.M. Shatla, Dr. Amr Refky,

Dept. of Electrical Engineering

Al-Azhar University

Cairo, Egypt

Eng_mo.hussein@yahoo.com

Abstract—Induction Motors are rugged, robust and widely used in industrial applications. Field oriented control (FOC) or Vector control is an efficient method to control either the speed or the torque or both of the three phase Induction motor. FOC converts the complex stator currents into two orthogonal components, one of which is responsible for speed control, and the other for the electromagnetic torque control, similar to DC machines. FOC has greater stability and capable of fast torque response and accurate speed control. This research explains the concept of FOC and its simulation rather than experimental implementation using Matlab simulink and MexBios softwares.

Keywords— Three phase Induction Motor; Scalar Control; Simulink program; MexBios software; Field oriented control (FOC); Field oriented control emulation.

I. INTRODUCTION

Induction motors are the preferred choice for industrial applications due to their rugged construction, maintenance free and development in power electronics, which leads to easy control of motor speed.

The three-phase Induction Motors speed is:

$$n = \frac{f_s}{p} (1 - s) \quad (1)$$

where;

f_s : Supply frequency in Hz

p : Number of pole pairs of stator

S : Slip between of rotating field and mechanical speeds

The speed of three phase Induction motor can be controlled by varying the frequency of the applied stator voltage via scalar control or vector control methods.

II. FIELD ORIENTED CONTROL

Field oriented control or offers more precise control of ac motors compared to scalar control. FOC is therefore used in high performance drives where oscillations in air gap flux linkages are intolerable, e.g. robotic actuators, centrifuges etc. In Field oriented control method, the components of the motor stator currents are represented by two orthogonal components

in a rotating reference frame. In a special reference frame, the expression for the electromagnetic torque of the smooth-air-gap machine is similar to the expression for the torque of the separately excited DC machine. The control is normally performed in a reference frame aligned to the rotor flux space vector.

The field oriented algorithm controls the components of the motor stator currents, represented by a vector, in a rotating reference frame d, q aligned with the rotor flux and perpendicular directions. The Clarke transformation converts three-phase currents i_a, i_b and i_c into two-phase orthogonal stator currents: i_α and i_β . These two currents in the fixed coordinate stator phase are transformed to the i_d and i_q currents components in the d, q reference frame using Park transformation.

The stator variables are translated into a flux model which is compared with a required reference values and then is tuned by PI controllers. After a back transformation from field to stator coordinates, the output voltage will be impressed to the machine with Pulse Width Modulation (PWM) technique [2, 3 and 4].

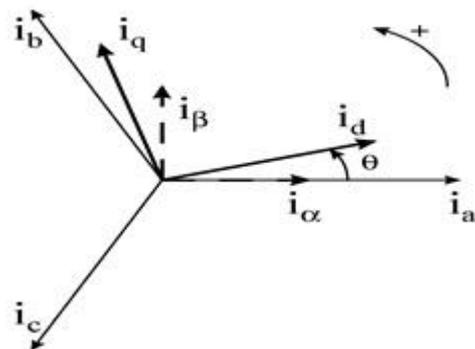


Fig. 1. Stator currents in the d, q rotating reference frame

where;

i_a, i_b, i_c : Three-phase stator currents

i_d : Direct current

i_q : Quadrature current

i_α, i_β : Two-phase orthogonal current

θ : The phase frame rotation angle

A. Mathematical Clarke transform

The mathematical Clarke transformation converts a three-phase system to a two-phase orthogonal system is obtained from:

$$i_{\alpha} = i_a \quad (2)$$

$$i_{\beta} = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \quad (3)$$

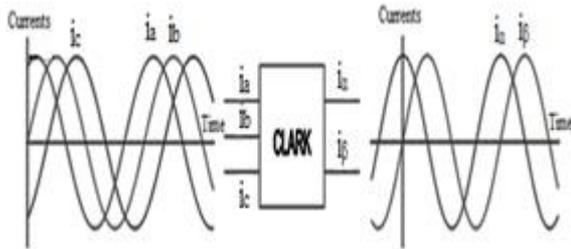


Fig. 2. Direct Clark Transform

B. Mathematical Park transform

The two phase components in the α, β frame representation calculated using Clarke transformation are then fed to a vector rotation block where it is rotated over a flux angle ρ to follow the frame d, q coincides to the rotor flux [2, 3 and 4].

The rotation over a flux angle ρ is achieved according to the formulae:

$$i_d = i_{\alpha} \cos(\theta) + i_{\beta} \sin(\theta) \quad (4)$$

$$i_q = -i_{\alpha} \sin(\theta) + i_{\beta} \cos(\theta) \quad (5)$$

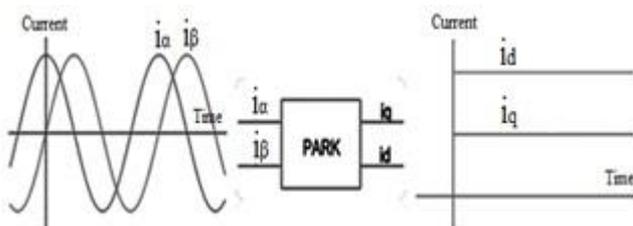


Fig.3. Direct Park Transform

C. Mathematical Inverse Park and Clarke transforms

Transformation uses the Inverse Clarke transformation, Figure 4. The vectors in the d, q frame are transformed from d, q frame to the two phases α, β frame representation calculated with rotation over a flux angle ρ according to the formulae:

$$i_{\alpha} = i_d \cos(\theta) - i_q \sin(\theta) \quad (6)$$

$$i_{\beta} = i_d \sin(\theta) + i_q \cos(\theta) \quad (7)$$

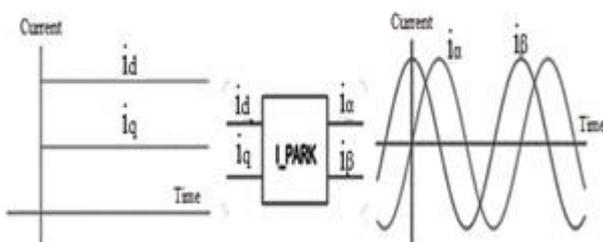


Fig. 4. Inverse Park Transform

The modification from a two-phase orthogonal α, β frame to a three-phase system is done using the equations:

$$i_a = i_{\alpha} \quad (8)$$

$$i_b = -\frac{1}{2} i_{\alpha} + \frac{\sqrt{3}}{2} i_{\beta} \quad (9)$$

$$i_c = -\frac{1}{2} i_{\alpha} - \frac{\sqrt{3}}{2} i_{\beta} \quad (10)$$

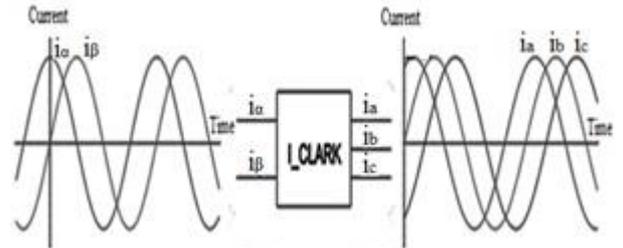


Fig. 5. Inverse Clarke Transform

III. SIMULINK PROGRAM

Simulink is a software add-on to MATLAB which is a mathematical tool that enables to simulate model, and analyze systems whose outputs change over time. Such systems are often referred to as dynamic systems.

S-functions (system-functions) provide a powerful mechanism for extending the capabilities of Simulink to allow addition new blocks to Simulink models [5].

A. Simulation of Vector Control of Induction Motor using Simulink program

Vector Control of three phase Induction Motor is simulated by the block diagram and S-Function as shown in Figures. 6 and 7 with following components:

- 1- Three phase Induction Motor
- 2- Clarke and Park transforms
- 3- i_d PI controller
- 4- i_q PI controller
- 5- Inverse Park and Clarke transforms
- 6- Plus width modulation (PWM)
- 7- Three phase Inverter
- 8- Speed PI controller

The simulation of the Motor starts at No load and when the Motor reaches the steady state speed, there will be two case studies of simulation:

- 1- Rotating the motor in the reverse direction by inverting the sign of quadrature current (i_q) with no load.
- 2- Increasing the load torque (T_m).

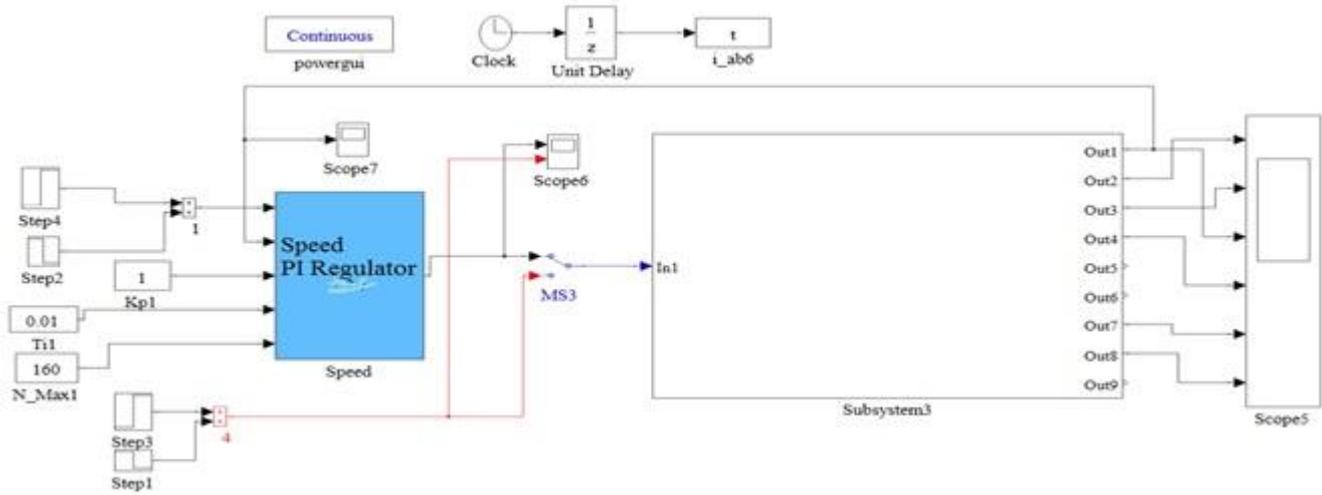


Fig. 6. Field oriented control for three phase Induction Motor schematic diagram in Simulink

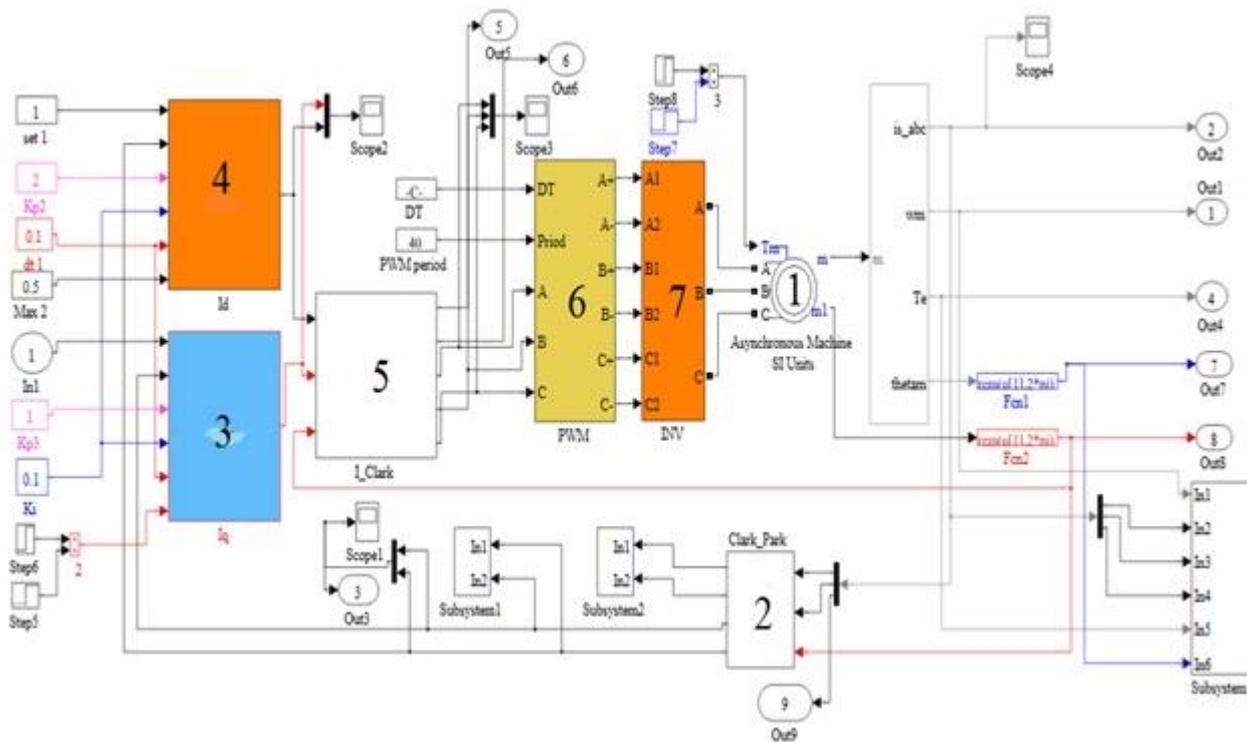


Fig. 7. Matlab Subsystem schematic diagram

B. Rotating the motor in the reverse direction

The motor starts with no load, the three phase currents (i_a, i_b, i_c) will start with large values (I_{start}) and they will have some distortions until the motor reaches it's no load speed. When the motor reaches its rated speed and its rated three phase currents, the motor continues running with no load for 0.5 seconds until we inverse the sign of quadrature current (I_q).

So, the quadrature current (I_q) reaches its peak value with negative sign then falls down to be close zero value because the motor runs with no load and the Direct current continue with its positive value as shown in Figure 8.

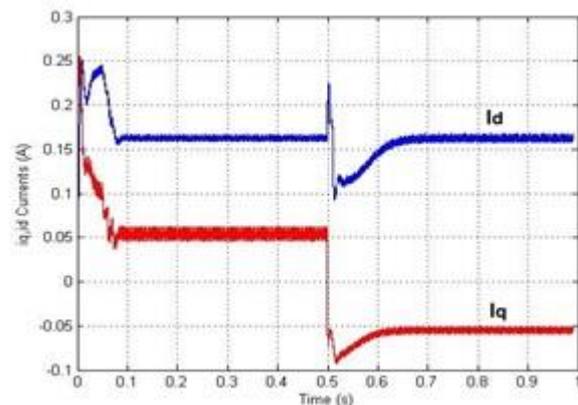


Fig. 8. Two phase currents VS Time

When the motor starts, the electromagnetic torque increases to reach the peak value, then falls down to be close zero value because the motor runs with no load. After Inversing the sign of quadrature current (I_q), the torque inverses its sign to reach the peak value and also falls down to be close zero value but with negative sign as shown in Figure 9.

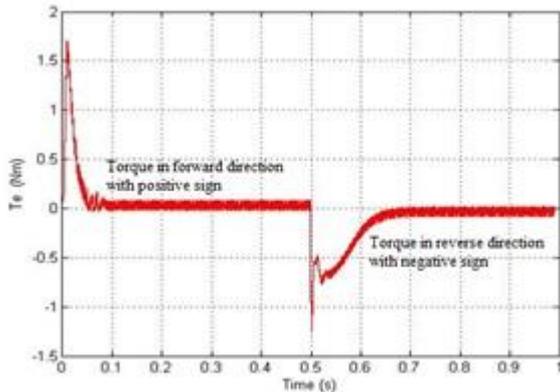


Fig. 9. Motor electromagnetic torque VS Time

When we inverse the sign of quadrature current (I_q) the motor will rotate in the reverse direction because the motor torque will inverse it's direction and the stator currents (i_a, i_b, i_c) will be reflected from (R S T) to (S T R) as shown in Figure 10.

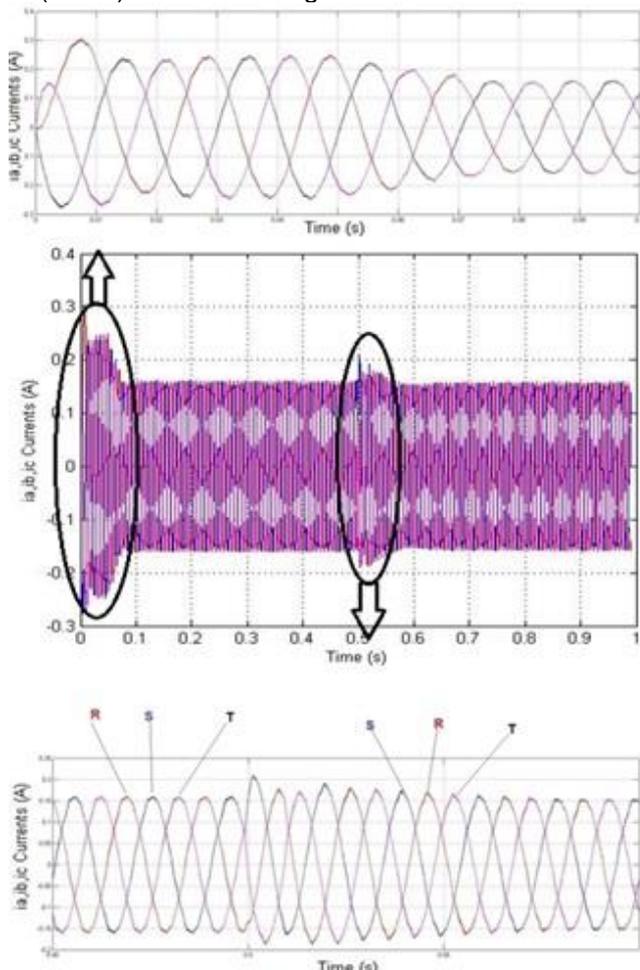


Fig. 10. Three phase (i_a, i_b, i_c) starting currents and after reflected from (R S T) to (S T R)

The motor speed will decrease to zero value, then the motor will reverse its rotation direction and will continue running at the reverse direction to reach the rated speed as shown in Figure 11.

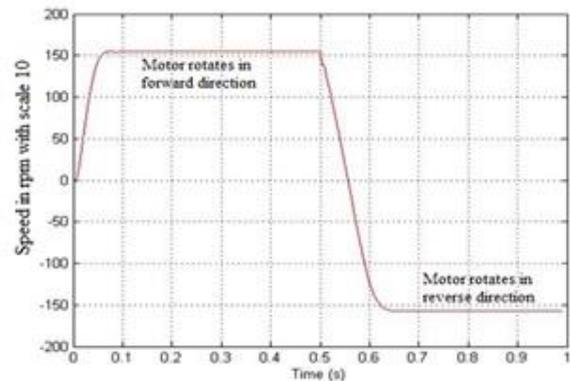


Fig. 11. Speed VS Time

When the motor starts with no load, the value of Theta increases slowly. After the motor reaches its rated speed, the value of Theta continues increasing with fast regular values. After Inversing the sign of quadrature current (I_q), the value of Theta starts decreasing and after the motor reaches the rated speed in the reverse direction, the value of Theta will decrease regularly. The simulation uses a function block to make the value of Theta increases to reach (2π) then falls down to zero and so on. At the reverse direction, the value of Theta decreases to reach zero value then rises up to (2π) and so on as shown in Figure 12.

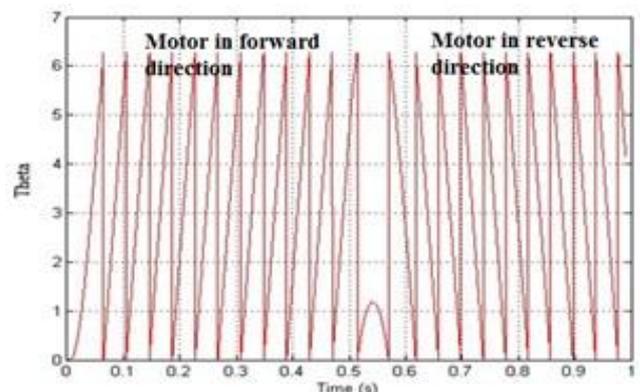


Fig. 12. Theta VS Time

C. Increasing the load torque using Current control

In this simulation the motor starts at No load and continue running with no load for (0.5 seconds) and after that we will suddenly increase the load torque (T_m), So, the three phase currents (i_a, i_b, i_c) will increase to cover this load, The (I_q PI controller) will increase the value of quadrature current (I_q) but the value of direct current (I_d) will not be affected because the direct current responsible only for the magnetization and the quadrature current (I_q) responsible for the torque as shown in Figure 13.

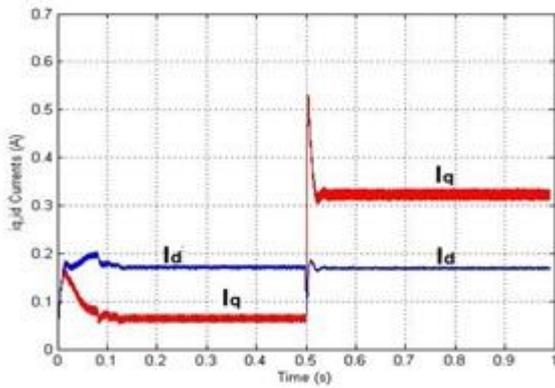


Fig. 13. Two phase currents/Time when increasing the torque

We note that when the motor starts, the mechanical torque increases to reach the peak value and falls down to be close zero value because the motor running with no load. After the load increases, the motor output torque will increase to cover this load as shown in Figure 14.

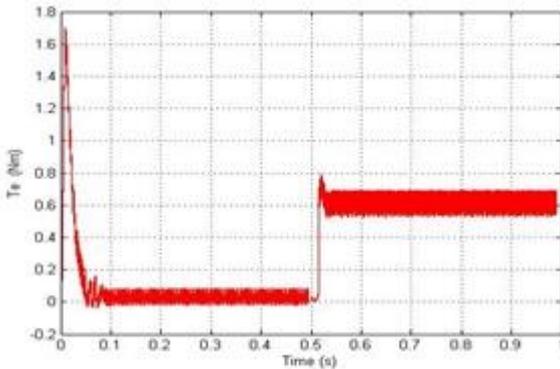


Fig. 14. Torque VS Time at increasing the load

When we use Current control only the motor speed will decrease according to the increase of the load but the controller will control the output speed to make the decrease of the speed better p.u. values as shown in Figure 15.

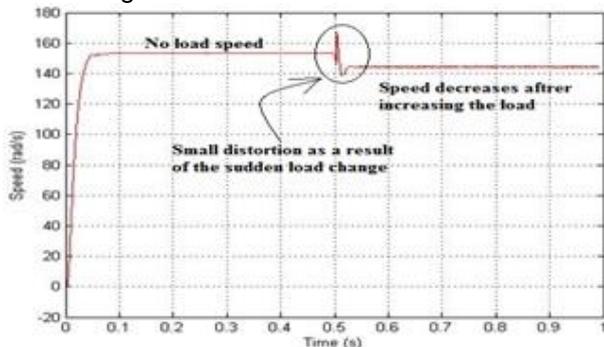


Fig. 15. Speed VS Time at increasing the load torque using Current control only

D. Increasing the load torque using Speed and Current control

In this simulation the motor starts at No load and after the motor reaches its rated speed the simulation will increase the load torque suddenly but will increase speed PI_controller to control the speed.

The speed PI_controller will increase speed PI_controller to control the speed. The speed PI_controller will cover the decrease of the speed to fix the speed at the set point of the speed PI controller we note that when we compare between Figures 15 and 16.

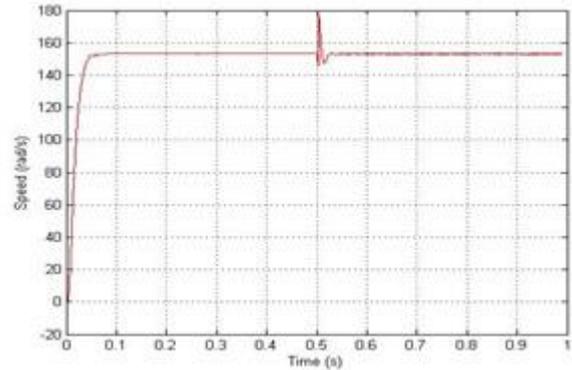


Fig. 16. Speed VS Time at increasing the load torque using Speed and Current control

IV. MEXBIOS SOFTWARE

MexBios software is a mathematical tool used to simulate vector control of three phase Induction motor. MexBios software has a very important feature, that it can simulate the program before loading it to Microcontroller, so the results can be expected by simulation before running the real Motor. MexBios program will simulate Vector Control of Induction Motor and the same program will be loaded to the Microcontroller [6].

A. Simulation of Vector Control of Induction Motor using MexBios program

The Simulation uses MexBios software to simulate Sensor Less Vector Control of two phase Induction Motor by simulating all components for the following schematic diagram as shown in Figure 17:

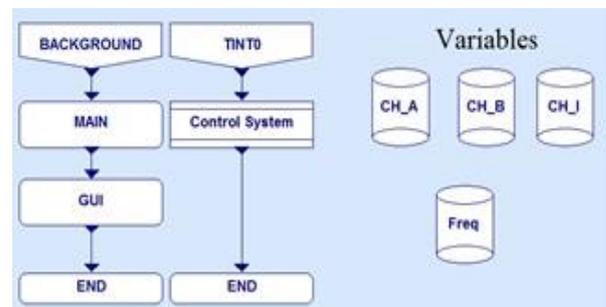


Fig. 17. Schematic diagram of control blocks

Schematic diagram for control blocks contains:

- 1- GUI (Graphical User Interface) which contains:
 - a) RADIO IN selector to select between Feedback from Motor Model or from real Motor
 - b) Reset Start / Stop Switch to reset all feedback signals
 - c) Two blocks to set Iq & Id reference values
 - d) Scopes and displays for Feedback signals Current Loop Formula

B. Simulation of positive value of quadrature current

When the positive value of quadrature current, the motor Speed starts from zero and increases fast (only one second) to reach the peak value (0.97 PU), these value mean that the motor runs at No load Speed. After the Motor reaches No load Speed, it will continue as shown in Figure 20.

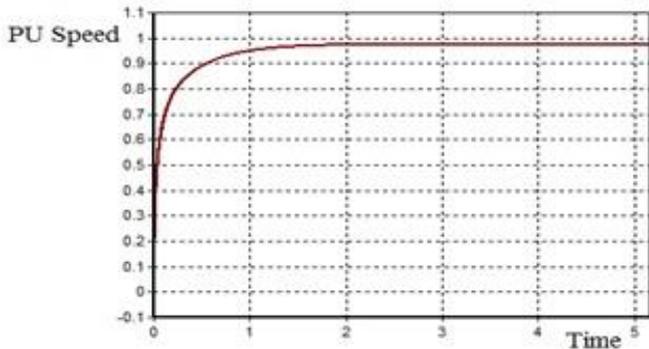


Fig.20. Speed Time response

When the motor starts, the starting I_q , I_d currents reach their peak values and starting to decrease to reach their No load values as shown in Figure 21 a and After the Motor reaches No load Speed, direct and quadrature currents will continue at their No load values as shown in Figure 21 b.

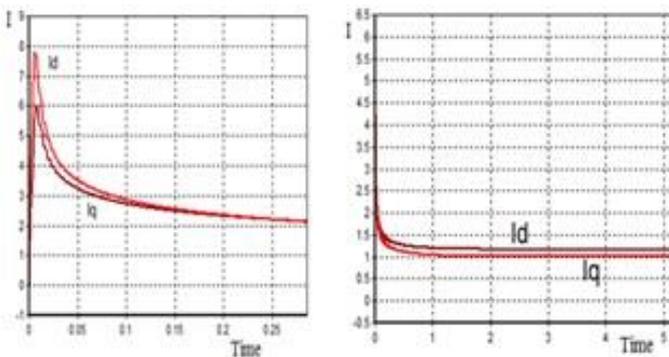


Fig. 21.a) I_q , I_d starting values b). I_q , I_d No load values

Since the Motor starts with Sensorless control, the value of Theta will start with a lot of disturbances for only few milliseconds until the Motor feedback signals return to the control circuits as shown in Figure 22 a and Fig 22 b, the same also happens with real work as in Figures 29 and 30.

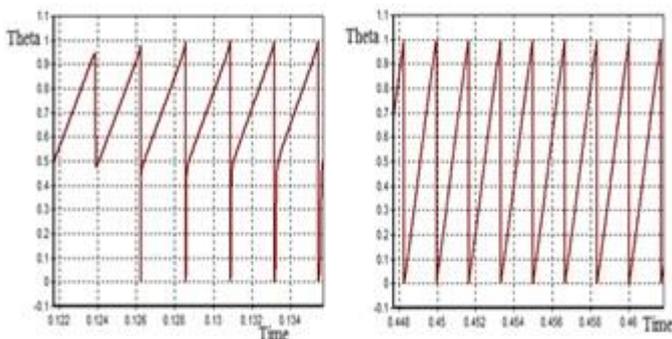


Fig. 22 a. Theta starting value b. Theta after few milliseconds

C. Simulation of negative value of quadrature current

For the negative value of quadrature current, the motor Speed starts from zero and increases fast (only one second) to reach the peak negative value (-0.97 PU), these value mean that the motor runs at No load Speed in the reverse direction. After the Motor reaches No load Speed, it will continue as shown in Figure 23.

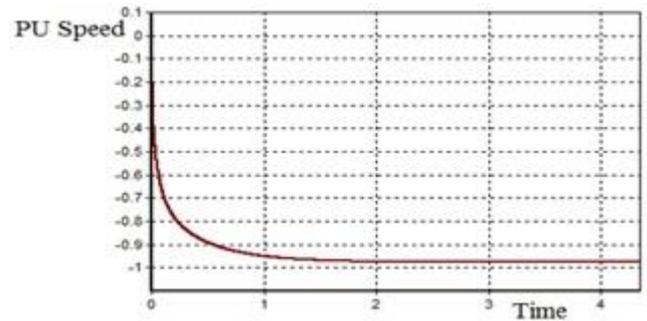


Fig. 23: Speed curve in PU with negative quadrature current

When the motor starts, the starting I_q , I_d currents reach their negative peak values and starting to increase to reach their negative No load values as shown in Figure 24 a. After the Motor reaches No load Speed, I_q , I_d currents will continue at their negative No load values as shown in Figure 24 b.

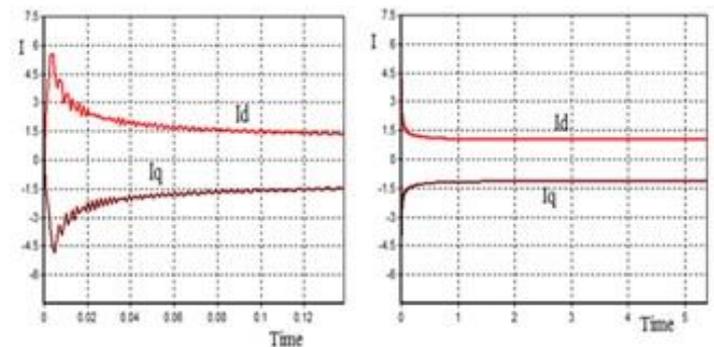


Fig. 24 a: Two phase current starting values with negative quadrature current b): Two phase current No load values with negative quadrature current

The value of Theta will start from zero and jump to one. Theta starts to decrease to reach to zero and jump again to one and so on. Because the Motor starts with Sensor less control, the value of Theta will start with a lot of disturbances for only few milliseconds until the Motor feedback signals return to the control circuits as shown in Figure 25 a and Figure 25 b.

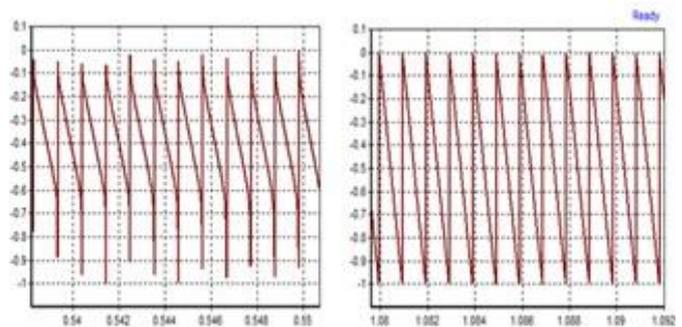


Fig. 25 a: Theta starting values with negative quadrature b): Theta after few milliseconds with negative quadrature

V. REAL WORK FOR VECTOR CONTROL OF INDUCTION MOTOR

A. Steps of the Experimental setup

The Experimental setup will use the following tools

- 1- Three phase Induction motor
- 2- Two DC Power supplies
- 3- Digital photo tachometer DT6234B
- 4- Motor Control Board which contains (MCB-02)
 - a- mZdsp 2812 Kit
 - b- Emulator (XDS 100V2)
- 5- Incremental shaft encoder BDK 05A 1024
- 6- MexBios software

B. Incremental shaft encoder BDK 05A 1024

A shaft encoder, also called a rotary encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code.

The Experimental setup will use Incremental shaft encoder (optical encoder) BDK 05A 1024 PPR (Figure 26).



Fig.26: Incremental shaft encoder BDK 05A 1024 PPR

The position information is delivered using a series of pulses – usually in phase quadrature, so that direction of travel can be determined. These are usually referred to as A/B pulses as shown in Figure 27. A separate pulse train, typically referred to as the N reference, provides one pulse per revolution to act as a reference mark [7].

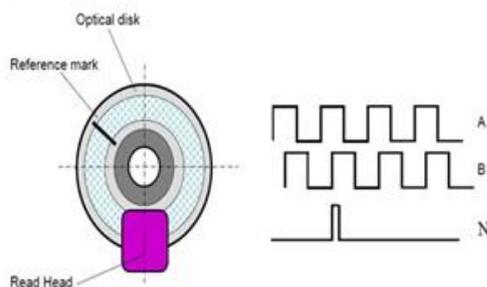


Fig. 27: Schematic of an incremental optical sensor with a reference pulse

VI. STARTING THE WORK

A. The connection

The arrangement of the experimental work is shown in Figure 28 however, the connection diagram is shown in Figure 29:



Fig. 28: Connection between MCB-2 Kit and Laptop using mZdsp 2812 Kit and XDS 100V2 emulator

• r

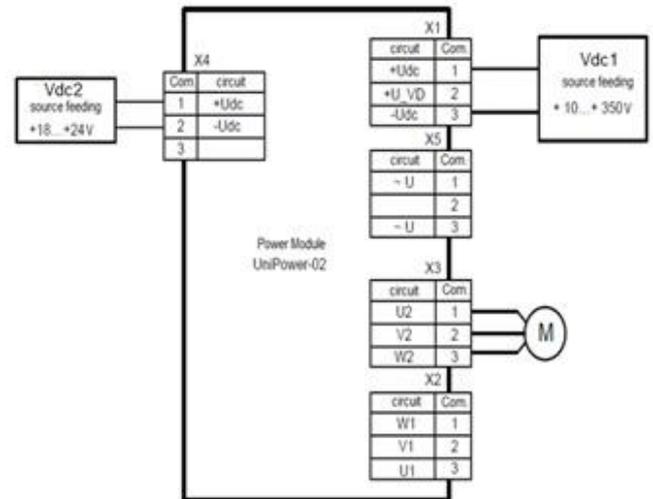


Fig. 29: Connection diagram

After the completion of installation of all the experimental work tools and pressing ON to the power supplies, start the MexBios program in Laptop and open GUI page to change Iq ref. value [8].

B. Experimental work will start with the appropriate setting as:

- 1- q ref. = 1
- 2- Vdc1 = 60 V
- 3- Vdc2 = 24 V
- 4- 0.25 Kw Motor with Delta connection

After finishing the program setting and pressing load to Ram the motor will start to rotate.

- After pressing Start Refreshing button and saving the Scope output Curves in buffer mode, we will get the following curves in Figure 30 have the following:

- 1- $I_q \text{ ref} = 1$
- 2- I_{qs} = the quadrature current feedback from the motor has a positive value.
- 3- I_{ds} = the direct current feedback from the motor continue with its positive value.
- 4- Theta starts from zero and jump to one and starts to decrease to reach zero and jump again to one and so on, it means the motor rotate in the reverse direction.
- 5- The motor rotates with Speed = 175.8 RPM Clock wise direction

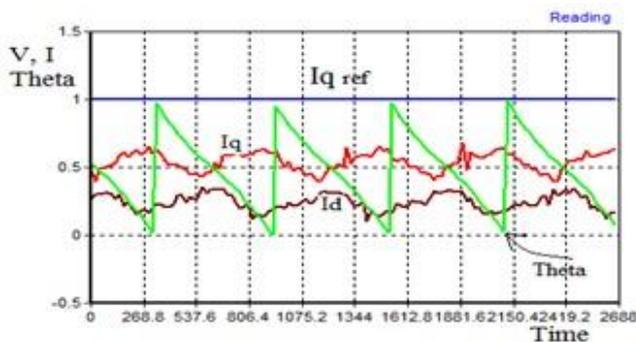


Fig. 30: Output Curves at $I_q \text{ ref} = 1$

C. Experimental work will start with the appropriate setting as:

- 1- $q \text{ ref} = -1$
- 2- $V_{dc1} = 60 \text{ V}$
- 3- $V_{dc2} = 24$
- 4- $V_{0.25 \text{ Kw}}$ Motor with Delta connection

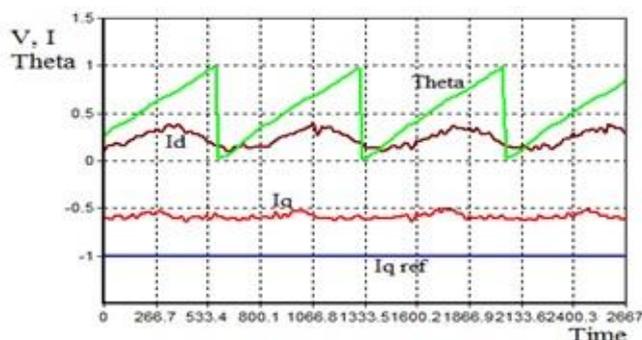


FIG. 31: OUTPUT CURVES AT $I_q \text{ REF} = -1$

- Output Curves in Figure 31 have the following:

- 1- $I_q \text{ ref} = -1$
- 2- I_{qs} = the quadrature current feedback from the motor has a negative value.
- 3- I_{ds} = the direct current feedback from the motor continue with its positive value.

- 4- Theta starts from zero and increase to reach one and jump to zero and starts to increase again to reach one and so on, it means the motor rotate in the positive direction.
- 5- The motor rotates with Speed = 175.8 RPM.

VII. CONCLUSIONS

The field oriented control of three phase Induction motor is more over simulated using MexBios software for closed loop speed control. This simulation is effective because the MexBios software is specialist significantly in control at many types of motors and after finishing the simulation the researchers can load the same program to the microcontroller RAM. The proposed system is verified experimentally using a three phase Induction motor and a PC. A low cost I/O interface card is used to transfer data between control circuit and the PC. The power circuit and MCB-02 Motor Control Board Kit, Incremental shaft encoder, mZdsp 2812 Kit are discussed. Motor speed and motor currents are sensed using interface card by the Laptop and using a digital photo tachometer.

The experimental work also shows that the speed control using field oriented control controls all the motor characteristics and makes the motor control easy. The motor characteristics are easily affected by adjusting the $I_q \text{ ref}$ value.

VIII. REFERENCES

- [1] Ion Boldea and Syed A . Nasar, "The Induction Machine Handbook", CRC Press 2002 Second Edition, ISBN: 978-0-8493-0004-2
- [2] Mahfooz Shalaby, "ELECTRIC DRIVES", Dar Elfikr Elarabi 2012 Third Edition, ISBN: 977-10-2659-3
- [3] Mircea Popescu, "INDUCTION MOTOR MODELLING FOR VECTOR CONTROL PURPOSES", Helsinki University of Technology Espoo 2000.
- [4] "Clarke & Park Transforms on the TMS320C2xx", Texas Instruments Incorporated 1997, Literature Number: BPRA048.
- [5] Ghislain Remy, Jean Deprez, "Workshop Matlab / Simulink in Drives and Power electronics", Palestine Polytechnic University, 18-21 May 2009.
- [6] "MexBIOS™ Development Studio Getting started – new user's manual Description of integrated development environment", NPF «Mechatronica-Pro» LTD, <http://mechatronica-pro.com>.
- [7] "Incremental encoders", TR - ELECTRONIC GmbH, Global Quality Management, TR-VCE-VI-GB-0002-05.DOC.
- [8] "mZdsp 2812 – Developer's module for processor TMS320LF2812 with pre-installed OE MexBIOS", Technical manual , Rev. 1.33" ", NPF «Mechatronica-Pro» LTD, <http://mechatronica-pro.com>.