ENHANCING THE SPEED CONTROL OF AN INDUCTION MOTOR USING GENETIC ALGORITHMS

Adzua Peter Terseer¹ and Engr. Dr. Goshwe Nentawe Yilwatda²

¹,²Department of Electrical/Electronics Engineering, University of Agriculture Makurdi, Nigeria

Abstract—Induction motors serve a very essential function in the industrial sector. In fields and drives especially, the induction motor becomes the most important tool. If the speed of an induction motor is not properly controlled, it becomes quite handy to achieve desired tasks with specific applications. Several conventional techniques to control the speed of an induction motor exist. However, these conventional techniques don’t do so well under a wide range of conditions due to their sensitivity to parameter variations. This paper has therefore demonstrated a strategy to improve the speed control capabilities of induction motors by using speed controllers based on artificial intelligence techniques like Genetic Algorithms. This objective is actualized using the versatile utilities of MATLAB/SIMULINK. The speed, torque and stator currents are observed for induction motor simulated with conventional and GA based controllers. Results obtained while using Genetic Algorithm based controllers demonstrate superior control capabilities.

Keywords—Genetic Algorithm, Induction Motor, Objective Function, Speed Control, Stator Current, Torque

I. INTRODUCTION

An induction motor is an alternating current (ac) powered asynchronous motor whose rotor winding is powered by electromagnetic induction from the magnetic field of the stator winding.

In industrialization, induction motors play a vital role. The fact that manufacturers constantly sought to minimize cost and avoid low efficiency has made variable motor speed technology become so important in order to improve efficiency, cut down the cost of production and also reduce noise. Without proper motor speed control, it is also practically impossible to achieve the desired task for a particular application. Motor speed is therefore, necessary in order to enhance maximum torque and efficiency [2,4].

Previously, fixed gain PI and PID controllers were chiefly used for the speed control of induction motors. However, the fact that such conventional control methods don’t do well under wide range of operational conditions due to their sensitivity to parameter variations [5,8,15], Artificial Intelligence based controllers like Fuzzy logic, Adaptive Neuro Fuzzy Inference System (ANFIS) and Genetic Algorithms (GA) have been deployed.

Genetic algorithms (GA) employ the principle of natural selection and genetics which is a stochastic global search method that mimics the process of natural evolution. GA locates high performance areas in complex domains without experiencing the difficulties associated with dimensionality or false optima as may occur with conventional techniques [3,6].

The main aim of this paper is to enhance the speed control of an induction motor using genetic algorithm techniques. Objectively, it aims to show the dynamic performance of an induction motor in terms of torque, rotor speed and stator currents when Genetic Algorithm Controller is used then compare the dynamic performances with the conventional controller and show improvements.

II. SPEED CONTROL TECHNIQUES

The various methods used in controlling the speed of induction motors include; control of supply frequency, slip power recovery, control of stator voltage, control of rotor resistance, pole changing and control of eddy currents [1].
From the list above, power recovery and rotor resistance control apply only to wound rotor motors since they are basically controlled within the rotor circuit. Pole changing applies only to squirrel cage motors whereas control of supply frequency and stator voltage are both applicable to squirrel cage and wound rotor motors.

To analyse and model induction motors, it is important to use their per phase equivalent circuit shown in Figure 1.

III. DYNAMIC INDUCTION MOTOR MODELLING

The torque and voltage equations that describe an induction motor’s dynamic behaviour are time varying and quite complex. Therefore, time varying inductances can be eliminated by reducing a poly phase winding with a d-q two phase winding thereby forming their magnetic axes in quadrature as shown in Figure 2 [10,13,14].

3.1 Inductance Matrix of Induction Motor

As evident in Figure 3, stator inductance $L_s$, and rotor inductance $L_r$, are respectively defined as

\[ L_s = L_{ls} + L_m \]  
\[ L_r = L_{lr} + L_m \]

Where

$L_{ls} = \frac{X_s}{\omega}$ is Stator leakage inductance

$L_m$ is magnetizing inductance and

$L_{lr} = \frac{X_r}{\omega}$ is rotor leakage inductance

However, the rotating electromotive force in Figure 3 are not represented by inductances but by voltage sources. Therefore, the associated flux voltage and current equations are shown as follows;

\[ \psi_{qs} = L_{qs} i_{qs} + L_{m} i_{qr} \]  
\[ \psi_{ds} = L_{m} i_{dr} + L_{qs} i_{ds} \]  
\[ \psi_{qr} = L_{r} i_{qr} + L_{m} i_{qs} \]
\[\Psi_{ds} = L_r i_{dr} + L_m i_{ds} \] (6)

The voltage equations therefore are

\[V_{qs} = R_s i_{qs} + \frac{d\Psi_{qs}}{dt} + \omega \Psi_{ds} \] (7)
\[V_{ds} = R_s i_{ds} + \frac{d\Psi_{ds}}{dt} - \omega \Psi_{qs} \] (8)
\[V_{qr} = R_r i_{qr} + \frac{d\Psi_{qr}}{dt} + (\omega - \omega_r)\Psi_{ds} \] (9)
\[V_{dr} = R_r i_{dr} + \frac{d\Psi_{dr}}{dt} - (\omega - \omega_r)\Psi_{qs} \] (10)

Where

\(\Psi_{qs}, \Psi_{ds}\) are the stator flux components in q and d axes respectively
\(\Psi_{qr}, \Psi_{dr}\) are the rotor flux components in q and d axes respectively
\(V_{qs}, V_{ds}\) are the stator voltage components in q and d axes respectively
\(V_{qr}, V_{dr}\) are the rotor voltage components in q and d axes respectively

Therefore, by substituting equations (6) and (7) into equation (10), \(V_{qs}\) becomes

\[V_{qs} = R_s i_{qs} + \frac{d(L_s i_{qs} + L_m i_{qr})}{dt} + \omega (L_m i_{dr} + L_s i_{ds}) \]

Similarly

\[V_{ds} = R_s i_{ds} + L_s \frac{d i_{qs}}{dt} + L_m \frac{d i_{qr}}{dt} + \omega L_m i_{dr} + \omega L_s i_{ds} \] (11)
\[V_{qr} = R_r i_{qr} + L_r \frac{d i_{qs}}{dt} + L_m \frac{d i_{qr}}{dt} + (\omega - \omega_r) L_m i_{ds} + (\omega - \omega_r) L_r i_{dr} \] (12)
\[V_{dr} = R_r i_{dr} + L_r \frac{d i_{qs}}{dt} + L_m \frac{d i_{qr}}{dt} - (\omega - \omega_r) L_m i_{qs} - (\omega - \omega_r) L_r i_{qr} \] (13)

However, \(V_{dr}\) and \(V_{qr}\) are shorted. Therefore, they both equal to zero. This implies that in matrix form, electrical transients of currents and voltages can be given as

\[
\begin{pmatrix}
V_{qs} \\
V_{ds} \\
0
\end{pmatrix} =
\begin{pmatrix}
R_s + L_s P & \omega L_s & L_m P \\
-\omega L_s & R_s + L_s P & -\omega L_m & L_m P \\
0 & (\omega - \omega_r) L_m & R_r L_r P & (\omega - \omega_r) L_r \\
-(\omega - \omega_r) L_m & L_m P & -(\omega - \omega_r) L_r & R_r L_r P
\end{pmatrix}
\begin{pmatrix}
i_{qs} \\
i_{ds} \\
i_{qr}
\end{pmatrix}
\] (15)

Where the stationary reference frame operator \(\frac{d}{dt}\) is represented by \(P\).

However, for synchronous frame, \(\omega = \omega_e\)

\[
\begin{pmatrix}
V_{qs} \\
V_{ds} \\
0
\end{pmatrix} =
\begin{pmatrix}
R_s + L_s P & \omega_e L_s & L_m P & \omega_e L_m \\
-\omega_e L_s & R_s + L_s P & -\omega_e L_m & L_m P \\
0 & (\omega_e - \omega_r) L_m & R_r L_r P & (\omega_e - \omega_r) L_r \\
-(\omega_e - \omega_r) L_m & L_m P & -(\omega_e - \omega_r) L_r & R_r L_r P
\end{pmatrix}
\begin{pmatrix}
i_{qs} \\
i_{ds} \\
i_{qr} \\
i_{dr}
\end{pmatrix}
\] (16)

As proposed by Robert H. Park, the equation below is a representation of how abc frame can be transformed into the d-q frame

\[
\begin{pmatrix}
V_d \\
V_q \\
V_0
\end{pmatrix} =
\begin{pmatrix}
\cos \theta & \sin(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\
\sin \theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\
0.5 & 0.5 & 0.5
\end{pmatrix}
\begin{pmatrix}
V_a \\
V_b \\
V_c
\end{pmatrix}
\] (17)

The inverse transformation is given as

\[
\begin{pmatrix}
V_a \\
V_b \\
V_c
\end{pmatrix} =
\begin{pmatrix}
\cos \theta & \sin \theta & 1 \\
\cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) & 1 \\
\cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) & 1
\end{pmatrix}
\begin{pmatrix}
V_d \\
V_q \\
V_0
\end{pmatrix}
\] (18)
3.2. Torque Equation of an Induction Motor

In vector space, torque equation is given as:

\[ T_e = \frac{3}{2} + P \times \frac{1}{\omega_b} (\Psi_{ds} i_{qs} - \Psi_{qs} i_{ds}) \]

\[ \Rightarrow T_e = \frac{3}{4} P \frac{l_m}{l_r} (\Psi_s I_s) \]  \hspace{1cm} (19)

Even though this torque equation is gotten from a stationary reference frame, it applies to other reference frames.

Therefore, can also be written as

\[ T_e = \frac{3}{4} P \frac{l_m}{l_r} (I_q \varphi_r) \]  \hspace{1cm} (20)

3.3. Current Calculations of an Induction Motor

According to figure 3, the stator and rotor current equations in q and d axis can be written as

\[ i_{qs} = \frac{(\varphi_{qs} - \varphi_{mq})}{X_{ls}} \]  \hspace{1cm} (21)

\[ i_{ds} = \frac{(\varphi_{ds} - \varphi_{md})}{X_{ls}} \]  \hspace{1cm} (22)

\[ i_{qr} = \frac{(\varphi_{qr} - \varphi_{mq})}{X_{ls}} \]  \hspace{1cm} (23)

\[ i_{dr} = \frac{(\varphi_{dr} - \varphi_{md})}{X_{ls}} \]  \hspace{1cm} (24)

Considering the current flow and having \( i = \frac{\varphi}{X} \), when \( \varphi = \varphi_{mq} \) and \( \varphi = \varphi_{md} \),

\[ \varphi_{mq} = X_{ml} \frac{\varphi_{qs}}{X_{ls}} + \frac{\varphi_{qr}}{X_{lr}} \]  \hspace{1cm} (25)

\[ \varphi_{md} = X_{ml} \frac{\varphi_{ds}}{X_{ls}} + \frac{\varphi_{dr}}{X_{lr}} \]  \hspace{1cm} (26)

\[ X_{ml} = \frac{1}{\frac{1}{X_{ml}} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}}} \]  \hspace{1cm} (27)

With regards to equations (10), (11), (12), (13), the flux linkage equations can be written as

\[ \frac{1}{\omega_b} d\varphi_{qs} = V_{qs} - \frac{\omega_e}{\omega_b} \varphi_{ds} - R_s i_{qs} \]  \hspace{1cm} (28)

\[ \frac{1}{\omega_b} d\varphi_{ds} = V_{ds} - \frac{\omega_e}{\omega_b} \varphi_{qs} + R_s i_{ds} \]  \hspace{1cm} (29)

\[ \frac{1}{\omega_b} d\varphi_{qr} = V_{qr} - \frac{\omega_e}{\omega_b} \varphi_{dr} + R_e (\varphi_{mq} - \varphi_{qr}) \]  \hspace{1cm} (30)

And

\[ \frac{1}{\omega_b} d\varphi_{dr} = V_{dr} - \frac{\omega_e}{\omega_b} \varphi_{qr} + R_e (\varphi_{md} - \varphi_{dr}) \]  \hspace{1cm} (31)

3.4. Rotor Speed of an Induction Motor

The rotor speed \( \omega_r \) and the motor torque are related in equation (35) below

\[ T_e = T_{load} + \int \omega_m \omega_m dt = T_{load} + \frac{2}{P} \int \omega_r \omega_r dt \]  \hspace{1cm} (32)

Which makes

\[ \omega_r = \int \frac{P}{2j} (T_e - T_l) \]  \hspace{1cm} (33)

Where:

\( P \) is the number of poles and

\( J \) is the moment of inertia (kg/m2)

IV. CONVENTIONAL INDUCTION MOTOR SIMULATION WITH MATLAB/SIMULINK
With respect to equations 1-33 above, the simulation of the induction motor with the conventional controller in MATLAB/SIMULINK is depicted in figure 3 below.

Figure 3. Conventional Induction Motor Simulation in MATLAB/SIMULINK

V. GENETIC ALGORITHMS

Genetic algorithm is an optimization technique in computing that finds exact or nearly exact solutions to search problems. It is a global search heuristic. This is a class of evolutionary algorithm technique inspired by evolutional biology like crossover, mutation, inheritance and selection. GA is implemented in simulations where a population of individual’s solution to an optimization problem evolves towards better solutions. Typically, binary strings of Os and 1s represent solutions [3].

The evolution begins with a population of randomly generated individuals and takes place in generations. It begins with no knowledge of the actual solution and depends wholly on environmental and operator responses to achieve the best solution. The fitness of each individual in each population of each generation is evaluated and multiple individuals are stochastically selected based on their fitness, recombined and randomly mutated to form a new population. The resultant population is once again iterated. The algorithm halts when the maximum number of generations is reached or a desired fitness level attained for the population.

Traditionally, Genetic Algorithm requires
a. A genetic representation of the solution domain.
b. A fitness function to evaluate the solution domain.

Genetic Algorithm is based on Charles Darwin’s principle of the survivor of the fittest. The individuals that can survive, live while others die. The principle is ideal for finding fitness values, f(x). The fitness function is an assigned numerical score that indicates how well a particular solution solves the problem. It represents the adaptability capability of the individual to the environment.

5.1. Implementing genetic algorithms

In order to reduce power loss in the motor, a PWM inverter is introduced in this model. This pulse width modulation inverter is responsible for controlling the power supplied to the motor. The average value of voltage and current supplied to the motor is controlled by turning ON and OFF the switch between supply and load at a fast rate [9,11,16].

To implement Genetic Algorithms in MATLAB, the optimization toolbox is used according to the following steps with the screen shot of figure 4.

1. on the MATLAB command line, the optimization toolbox is invoked by typing `optimtool` and Figure 4 below appears
2. The solver is set to Genetic Algorithm
3. The Objective function is entered as “@GA_obj_ftn”. This summarizes and assigns a figure showing how close a particular chromosome is to achieving the set aims of the problem at hand. In this research work, the objective function determines the best PID controller that produces the fastest rise and settling times. It is thus created according to equation
\[ C_{pid} = \frac{K_p}{S^2} + K_D S + K_I \] (34)
4. According to equation (34), the objective function has three variables (P,I,D). Therefore, the number of variables is set to 3.
5. The lower and upper bound limits are set to 2 and 100 respectively.
6. The population size is set to 120. This refers to the number of chromosomes involved in the iteration.
7. Population type is set to double vector
8. The initial population, scores and range are set to default.
9. Fitness scaling is set to RANK in order to prevent premature convergence (This determines how fit an individual solution is for selection)
10. Stochastic Uniform selection method is preferred in order to give weaker members of the population (according to their fitness) a chance to be chosen and thus reduce the unfair nature of fitness-proportional selection methods. (Selection is the stage of a genetic algorithm at which individual genomes are chosen from a population for future recombination/crossover).
11. Elite count is set to 4; this specifies the number of individuals that are guaranteed to survive to the next generation
12. A crossover fraction of 0.8 is used to show the fraction of the next generation, other than elite children, that are produced by crossover. (Crossover performs sexual genetic recombination).
13. Mutation function is set to constraint dependent. This is the probability with which all bit positions of all chromosomes in a new population undergo a random change after the process of selection.
14. The crossover function is set to “arithmetic” in order to create children that are the weighted arithmetic mean of the two parents
15. The migration fraction and interval are both set to 0.2 and 20 respectively
16. The initial constraint penalty and penalty factor are set to 10 and 100 respectively.
17. The stopping criteria is set to 120 generations
18. The plot functions are set to Best Fitness and Best Individual
19. The algorithm is initiated by clicking the start button.
20. When the algorithm terminates, the backend code responsible for the simulation of the algorithm is generated by clicking on file, then Generate code. The generated code is subsequently saved as “ga_code.m”

21. This code is then embedded in a MATLAB function block and it replaces the conventional controller of Error! Reference source not found. as the GA controller as shown in Figure 5 below.

![Figure 5. Induction motor with GA controller](image)

22. Thereafter, the entire motor is simulated and results observed and compared with the conventional controllers.

VI. RESULTS AND DISCUSSIONS

![Figure 6. Induction Motor Speed](image)

![Figure 7. Induction Motor Torque](image)
Figure 8. Induction Motor Stator Current

From Figure 6, it is apparent that even though the results are taken in the same time line, the speed increases while the speed rise time drastically reduces as the genetic algorithm based controller is introduced. While the rise time is quite slow with the conventional controller due to the observed obvious ripples to speed response, a massive improvement is observed when the Genetic Algorithm based controller is employed. The settling time just like the rise time reduces as the GA based controller is introduced.

With regards to computational time, it is also apparent that the conventional controller takes a longer time to reach its peak as compared to the GA based controller.

More also, with GA based controller, there are negligible ripples in speed response as compared to the conventional controller.

Furthermore, as it is clear to see in Figure 7, converging to zero in GA based controller is happening intensively in less duration of time. Unlike the GA based controller where the torque oscillations are reduced and decay more rapidly, worrying oscillations occur in the torque response of the conventional controller before it finally settles down.

With regards to Figure 8, the induction motor with conventional controller requires very high startup current and once it has started, it exhibits a higher degree of stator current ripples (current imbalance). Whereas, the GA based controller has smoothened the ripples in the motor torque and stator currents. The GA based controller has exhibited very low (acceptable) startup current which took lesser time to settle as compared to its conventional counterpart.

VII. CONCLUSION

Computer based simulations have been successfully carried out for induction motor speed control using an induction motor with a conventional and Genetic Algorithm based controller. At the end of the simulation therefore, it is found that the genetic algorithm based controller has yielded better results. It has yielded the highest speed in the shortest time with no ripples to speed response, its torque convergence to zero is the quickest and at startup, it has exhibited very low stator current that balanced up in the shortest time.

REFERENCES

[1] K. Anitha, G. Santosh, L. Suneel, K. Spandana, and E. Raviteja, Simulation and Speed Control of 3-Phase Induction Motor Drives. International Journal of Engineering Science and Computing, 2016, ISSN 2321 336
[2] L. M. Barazane and R. Ouigouini, ANFIS Speed Controller for Vector Control of an Induction Motor. EFEEA’10 International Symposium on Environment Friendly Energies in Electrical ApplicationsGhardaïa, Algeria, 2010.
[3] E. G. Carrano, R. H. C. Takahashi, W. M. Caminhas and O. M. Neto, A Genetic Algorithm for Multiobjective Training of ANFIS Fuzzy Networks. Congr. Evol. Comput. pp. 3259-3265, 2008.
[4] R. Divya, S. Swati, and B. Vijay, Fuzzy Speed Controller Design of Three Phase Induction Motor. International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, 2012, Volume 2, Issue 5.
[5] V. D. Gauri and S. S. Sankeshwari, Speed Control of Induction Motors using Hybrid Pi plus Fuzzy Controller. International Journal of Advances in Engineering & Technology, 2013, ISSN: 22311963.
[6] K. M. Indurupalli, K. S. R. Anjaneyulu and B. V. Prashanth, Design of Robust Speed Controller by Optimization Techniques for DTC IM Drive. International Journal of Computer Networks and Communications Security, 2013, VOL. 1, NO. 1.

[7] B. Iulian, M. Virgil, P. Sorin, and R. Calin, Indirect Vector Control of an Induction Motor with Fuzzy-Logic based Speed Controller. 3rd International Symposium on Electrical Engineering and Energy Converters Suceava, 2009.

[8] L. M. Madhav, and S. P. Muley. Induction Motor Speed Control using PID Controller. International Journal of Technology and Engineering Science Vol 1(2), pp151-155

[9] M. Meenakshi and K. Vinay, Asynchronous Machine Modelling using Simulink fed by PWM Inverter. International Journal of Advances in Engineering & Technology, 2011, ISSN: 2231-1963

[10] H. M. Mohamadand G. K. Pouria, A New Simulation of Induction Motor. Australasian Universities Power Engineering Conference, (AUPEC), pp. 1-6, 2008.

[11] A. Nabae, S. Ogasawara and H. Akagi, A novel control scheme for current-controlled PWM inverters. IEEE Trans. on Industry Applic., vol.22, no. 4. Pp. 697-701, 1986.

[12] D. B. Shashank and P. D. Gajanan, ANFIS Control Scheme for the Speed Control of the Induction Motor. Int. Journal of Engineering Research and Applications ISSN: 2248-9622, Vol. 4, Issue 3 (Version 1), pp.35-39 2014.

[13] D. B. Shashank and P. D. Gajanan, Design of Simulink Model with ANFIS Controller for Controlling Parameters of 3Phase Induction Motor Drives. International Journal of Advanced Research in Computer Science and Software Engineering, 2014, Volume 4, Issue 4.

[14] S. A. Sifat and M. B. Rashid, Direct Quadrature (D-Q) Modeling of 3-Phase InductionMotor Using MatLab / Simulink. Canadian Journal on Electrical and Electronics Engineering, 2012, Vol. 3, No. 5

[15] S. Swati and B. Amol, Discrete PI and PID Controller Based Three Phase Induction Motor Drive: A Review. International Journal of Electrical, Electronics ISSN No. (Online): 2277-2626 and Computer Engineering 2(2): pp 97-100, 2013.

[16] H.W. Van der Broeck, H. Skudelny, and G. Stanke, Analysis and realization of a pulse width modulator based on voltage space vectors. IEEE Trans. on Industry Applications. vol. 24, no. 1. Pp. 142-150, 1988.