Nonlinear Control of Induction Motor

Mutsaied Shirazi, Abdulmajeed Muhammad Kabir
King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

***Abstract* - Induction motors are rugged, reliable and inexpensive thus finding numerous applications in the industry. The control of these machines are however complicated and remain under study. In this work, we make use of the backstepping nonlinear control technique to regulate the speed and magnetic flux of the induction motor with viscous friction. The controllers obtained were implemented in Matlab, thereafter, a practical setup was proposed for real time experimentation of the system.**

Keywords - Induction Motor, Lyapunov function, Backstepping control, Observer, Matlab.

I. INTRODUCTION

An Induction motor is an alternating current asynchronous machine with wide applications in industry and household. The attractiveness of induction motors stems from the absence of mechanical commutation which increases their reliability, ruggedness and life span, reduces maintenance requirements and downtime. The control of these machines however are not straightforward. This is mainly due to the following reasons; first, the induction motor is a highly nonlinear machine, secondly, some of the states such as magnetic flux and rotor currents are unknown or difficult to measure. Thirdly, some parameters such as rotor resistance, inertia and load torque are usually varying [1]. The problem of unavailability of system variables can be solved by the use of estimation techniques. Numerous research effort has been placed in the development and optimization of these methods. In the induction motor, stator currents and rotor speed or position are available for measurement [2]. It is thus paramount to estimate the rotor flux using an observer. Ph. Martin described simple observers for the measurement of flux in induction motors [1]. Among the classical approaches to the control of the induction motor include the variable frequency[], field-oriented control proposed by Blaschke [3] and direct torque control[]. Fekih has applied the field-oriented control technique with partial feedback linearization and a proportional-integral controller for the induction motor. It was also shown that this technique works well in low speeds [2]. Chiasson mentions feedback linearization, input output linearization, backstepping methods, passivity based techniques and variable structure as control methods for induction motors [4]. The input to output linearization technique developed in the past involves finding a nonlinear state feedback and a state transformation such that the transformed system is input to output decoupled or in linear form [2]. After the linearization(decoupling) has been achieved, linear control techniques can then be applied. It was however shown by Marino et al., that the induction motor is not static feedback linearizable [5] thus prompting the development of a new dynamic feedback linearization technique by Chiasson. The concept of dynamic feedback linearization is based on the addition of integrators to the inputs resulting into a higher order system which can be feedback linearizable [6].

In this work, the control of Induction motor based on field-orientation is treated by applying the backstepping control technique. The choice of this controller stems from dislike of the complexity in applying feedback linearization and the undesirable chattering involved in sliding mode control. The nonlinear model of the induction motor is treated as a case with viscous friction and load torque.

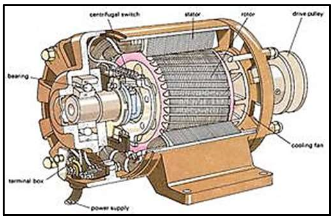

Fig.1: Induction Motor[7]

Sensorless control of induction motor was studied in [8] by reducing the adverse effect caused by the variation of the rotor time constant. A novel robust observer, Luenberger observer and sliding mode approach were utilized in the work. In real time implementation, the current and the voltage values of the motor were transferred to the controller program by means of the analog-to-digital converter on the motion control card. Intelligent control techniques with real time implementation methods have also been studied previously. The intersection between intelligent systems and computer approaches propose ease of real time implementation of controllers. In [9], adaptive speed control of induction motor was described and implemented using a special digital signal processor (DSP). The controller-structure design was based on vector control scheme that transforms the three phase motor currents into flux and torque generating current components. The current control loops were realized with PI controllers while the outer loop was based on a class of model reference adaptive controller (MRAC) with a first order system as the reference model. An insight into real time implementation of the controller was gained from [10] wherein a novel Fuzzy vector control for induction motor was proposed. The technique was proposed to replace the block of Field oriented control with a new block control with the intention to maintain decoupling and overcome the problem of robustness with respect to the parametric variations. The choice of the Fuzzy vector control was shown to be advantageous in a robust sense. The fuzzy vector control scheme was successfully implemented in real-time using a digital signal processor board DS 1104.

I. INDUCTION MOTOR MODEL

The model of the induction motor has been derived and utilized by numerous authors. The model below (1) was adapted from [4];

$$\begin{bmatrix} \frac{d\theta}{dt} \\ \frac{d\omega}{dt} \\ \frac{d\psi_{Ra}}{dt} \\ \frac{d\psi_{Rb}}{dt} \\ \frac{di_{Sa}}{dt} \\ \frac{di_{Sb}}{dt} \end{bmatrix} = \begin{bmatrix} \omega \\ \mu(\psi_{Ra} i_{Sb} - \psi_{Rb} i_{Sa}) - \frac{T_l}{J} - \frac{B\omega}{J} \\ -\eta\psi_{Ra} - n_p\omega\psi_{Rb} + \eta M i_{Sa} \\ -\eta\psi_{Rb} + n_p\omega\psi_{Ra} + \eta M i_{Sb} \\ \eta\beta\psi_{Ra} + \beta n_p\omega\psi_{Rb} - \gamma i_{Sa} + \frac{u_{Sa}}{\sigma L_s} \\ \eta\beta\psi_{Rb} - \beta n_p\omega\psi_{Ra} - \gamma i_{Sb} + \frac{u_{Sb}}{\sigma L_s} \end{bmatrix} \quad (1)$$

where; $\eta = \frac{R_r}{L_r}, \beta = \frac{M}{\sigma L_r L_s}, \mu = \frac{n_p M}{J L_r}$, $\sigma = 1 - \left(\frac{M^2}{L_r L_s}\right)$,

$$\gamma = \frac{M^2 R_r}{\sigma L_r^2 L_s} + \frac{R_s}{\sigma L_s}$$

In [4], the motor dynamics obtained were transformed to direct-quadrature (d-q) frame. The d-q transformation is used to represent the three-phase motor with an equivalent two-phase model [11]. The model of the induction motor with $\alpha - \beta$ representation according to Marino et al. [12] as presented in [4] is a set of six nonlinear equations given in (1): In this work, we incorporate the term $B\omega/J$ representing viscous friction to the model.

The d-q model is obtained by applying the following transformations (2) [13]:

$$\begin{bmatrix} \omega \\ \rho \\ i_d \\ i_q \\ \psi_d^2 \\ v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \omega \\ \tan^{-1}(\frac{\psi_{Rb}}{\psi_{Ra}}) \\ \cos(\rho)\, i_{sa} + \sin(\rho) i_{sb} \\ -\sin(\rho)\, i_{sa} + \cos(\rho) i_{sb} \\ \psi_{Ra}^2 + \psi_{Rb}^2 \\ +\cos(\rho)\, u_{sa} + \sin(\rho)\, u_{sb} \\ -\sin(\rho)\, u_{sa} + \cos(\rho)\, u_{sb} \end{bmatrix}$$

$$\begin{bmatrix} i_d = \dfrac{(\psi_{Ra} i_{sa} + \psi_{Rb} i_{sb})}{\sqrt{\psi_{Ra}^2 + \psi_{Rb}^2}} \\ i_q = \dfrac{(\psi_{Ra} i_{sb} - \psi_{Rb} i_{sa})}{\sqrt{\psi_{Ra}^2 + \psi_{Rb}^2}} \\ \psi_d = \sqrt{\psi_{Ra}^2 + \psi_{Rb}^2} \\ \psi_q = 0 \end{bmatrix} \quad (2)$$

Finally, we obtain the model of the induction motor in d-q frame as (3):

$$\begin{bmatrix} \frac{d\omega}{dt} \\ \frac{d\psi_d}{dt} \\ \frac{di_d}{dt} \\ \frac{di_q}{dt} \\ \frac{d\rho}{dt} \end{bmatrix} = \begin{bmatrix} \mu\psi_d i_q - \frac{T_l}{J} - \frac{B\omega}{J} \\ -\eta\psi_d + \eta M\ {}_d \\ -\gamma i_d + \eta\beta\ \ {}_d + n_p\omega i_q + \frac{\eta M\ {}_q^2}{\psi_d} + \frac{u_d}{\sigma L_s} \\ -\gamma i_q - \beta n_p\omega\psi_d - n_p\omega i_d - \frac{\eta M i_q i_d}{\psi_d} + \frac{u_q}{\sigma L_s} \\ n_p\omega + \frac{\eta M\ {}_q}{\psi_d} \end{bmatrix} \quad (3)$$

II. BACKSTEPPING CONTROL

In this section, the backstepping algorithm is presented in part A while its implementation for the induction mo tor is developed in part B.

A. Backstepping Algorithm

The backstepping controller is regarded as a very useful tool when some states are controlled through other states. This technique uses a state as a virtual controller to another state given that the system is in triangular feedback form. It also overcomes the problem of finding a control Lyapunov function as a design tool. In backstepping control design, nonlinear systems or subsystems of the form (4):

$$\dot{x} = f(x) + g(x)\xi \quad (4)$$
$$\dot{\xi} = u$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ is the control input, $\xi \in \mathbb{R}$, $f: \mathcal{D} \to \mathbb{R}^n$, $g: \mathcal{D} \to \mathbb{R}^n$. Assume that there exists a state feedback $\lambda(x)$, $\lambda(0) = 0$ such that the origin of the system:

$$\dot{x} = f(x) + g(x)\lambda(x)$$

is as asymptotically stable. Also, if a Lyapunov function $V(x)$ is known such that:

$$\frac{\partial V(x)}{\partial x}[f(x) + g(x)\lambda(x)] \le -W(x), \qquad \forall\, x \in \mathcal{D}$$

then, the controller:

$$u = \frac{\partial\lambda}{\partial x}[f(x) + g(x)\xi] - \frac{\partial V}{\partial x} g(x) - k(\xi - \lambda(x)) \quad (5)$$

makes the origin of the system asymptotically stable [14]. Furthermore, the time derivative of the Lyapunov function:

$$\tilde{V} = V(x) + \frac{1}{2}\left(\xi - \lambda(x)\right)^2$$

along the trajectories of the system is given by:

$$\dot{\tilde{V}} \le -W(x) - k\left(\xi - \lambda(x)\right)^2 \quad (6)$$

B. Implementation of the Backstepping Controller

In the design of the backstepping controller, we start by making a simplification transformation for the states and constants:

$$x_1 = \omega, \quad x_2 = \psi_d, \quad x_3 = \rho, \quad x_4 = i_d, \quad x_5 = i_q$$

$$k_1 = \mu, k_2 = \frac{B}{J}, k_3 = \eta, k_4 = \eta M, k_5 = n_p, k_6 = \gamma, k_7 = \eta\beta, k_8 = \frac{1}{\sigma L_s}, k_9 = n_p\beta, u_d = u_1, u_q = u_2$$

resulting to the state space model:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix} = \begin{bmatrix} k_1x_2x_5 - k_2x_1 - \frac{T_L}{J} \\ -k_3x_2 + k_4x_4 \\ k_5x_1 + \frac{k_4x_4}{x_2} \\ -k_6x_4 + k_7x_2 + k_5x_1x_4 + \frac{k_4x_4^2}{x_2} + k_8u_1 \\ -k_6x_5 - k_9x_1x_2 - k_5x_1x_4 - \frac{k_4x_4x_5}{x_2} + k_8u_2 \end{bmatrix} \quad (7)$$

In the derivation of the backstepping controller however, we consider the case of zero torque load $T_L = 0$ and define the error e_1 *and* e_2 as the deviation of the respective states from their required values:

$$\begin{aligned} e_1 &= x_{1r} - x_1 \\ e_2 &= x_{2r} - x_2 \end{aligned} \quad (8)$$

x_{1r} is the desired speed reference and x_{2r} is the desired flux reference.

Taking the Lyapunov function of the form:

$$\begin{aligned} V_1 &= \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \\ \dot{V}_1 &= e_1\dot{e}_1 + e_2\dot{e}_2 \end{aligned} \quad (9)$$

The derivatives of the error terms are:

$$\begin{aligned} \dot{e}_1 &= \dot{x}_{1r} - \dot{x}_1 \\ \dot{e}_1 &= \dot{x}_{1r} - k_1x_2x_5 + k_2x_1 \end{aligned} \quad (10)$$

similarly;

$$\begin{aligned} \dot{e}_2 &= \dot{x}_{2r} - \dot{x}_2 \\ \dot{e}_2 &= \dot{x}_{2r} + k_3x_2 - k_4x_4 \end{aligned} \quad (11)$$

then the derivative of the Lyapunov becomes:

$$\dot{V}_1 = e_1(\dot{x}_{1r} - k_1x_2x_5 + k_2x_1) + e_2(\dot{x}_{2r} + k_3x_2 - k_4x_4) \quad (12)$$

by the backstepping control technique, we take x_2x_5 and x_4 as the virtual controllers.

$$x_2x_{5_r} = \frac{1}{k_1}(\dot{x}_{1r} - k_2x_1 + g_1e_1)$$

and

$$x_{4_r} = \frac{1}{k_4}(k_3x_2 + \dot{x}_{2r} + g_2e_2) \quad (13)$$

The above choice of virtual control inputs renders the derivative of the Lyapunov function negative definite.

$$\dot{V}_1 = -g_1e_1^2 - g_2e_2^2, \qquad g_1, g_2 > 0 \quad (14)$$

Thus the reference controllers $x_2x_{5_r}$ and x_{4_r} defined above would make the states converge to some reference value and to achieve the tracking of the state references, we further define the errors:

$$\begin{aligned} e_3 &= x_2x_{5_r} - x_2x_5 \\ e_4 &= x_{4_r} - x_4 \end{aligned}$$

The new Lyapunov function is chosen as:

$$V_2 = V_1 + V_\varrho \quad (15)$$

where

$$V_\varrho = \frac{1}{2}e_3^2 + \frac{1}{2}e_4^2$$

The derivative of the Lyapunov function V_2 is:

$$\dot{V}_2 = -g_1e_1^2 - g_2e_2^2 + e_3\dot{e}_3 + e_4\dot{e}_4 \quad (16)$$

with

$$\dot{e}_3 = (x_2\dot{x}_{5_r}) - x_2\dot{x}_5 - \dot{x}_2x_5 \text{ and } \dot{e}_4 = (\dot{x}_{4_r}) - \dot{x}_4 \quad (17)$$

Thus:

$$\begin{aligned} \dot{V}_2 = &-g_1e_1^2 - g_2e_2^2 + e_3\big((x_2\dot{x}_{5_r}) - x_2\dot{x}_5 - \dot{x}_2x_5\big) \\ &+ e_4((\dot{x}_{4_r}) - \dot{x}_4) \end{aligned} \quad (18)$$

upon substitution of (17) into (16);

$$\begin{aligned} \dot{V}_2 = &-g_1e_1^2 - g_2e_2^2 + e_3\Big(\big((x_2\dot{x}_{5_r}) - x_5(-k_3x_2 + k_4x_4)\big) \\ &-x_2\left(-k_6x_5 - k_4x_1x_2 - k_5x_1x_4 - \frac{k_4x_4x_5}{x_2}\right) - x_2k_8u_1\Big) \\ &+e_4\left((\dot{x}_{4_r}) + k_6x_4 - k_7x_2 - k_5x_1x_4 - \frac{k_4x_4^2}{x_2} + k_8u_2\right) \end{aligned} \quad (19)$$

The control variables u_1 and u_2 appearing in the above equation are set to make the Lyapunov function negative definite. The appropriate choice of control inputs is thus:

$$\begin{aligned} u_1 = \left(\frac{1}{x_2k_8}\right)\Big[&-x_2\Big(-k_6x_5 - k_4x_1x_2 - k_5x_1x_4 - \frac{k_4x_4x_5}{x_2}\Big) \\ &- x_5(-k_3x_2 + k_4x_4) + (x_2\dot{x}_{5_r}) + g_3e_3\Big] \end{aligned}$$

and

$$u_2 = \left(\frac{1}{k_8}\right)\left[\frac{k_4x_4^2}{x_2} + k_5x_1x_4 + k_7x_2 - k_6x_4 - (\dot{x}_{4_r}) - g_4e_4\right] \quad (20)$$

Substituting the controller values gives

$$\dot{V}_2 = -g_1e_1^2 - g_2e_2^2 - g_3e_3^2 - g_4e_4^2, \qquad g_{i=1\ldots4} > 0$$

Finally by implementing the backstepping controllers above, u_1 and u_2, input reference tracking of the induction motor can be achieved.

III. SIMULATION, RESULTS AND EXPERIMENTAL SETUP

A. Matlab Simulation

The backstepping controller was simulated in Matlab using Simulink. Fig.2 below shows the Simulink block diagram.

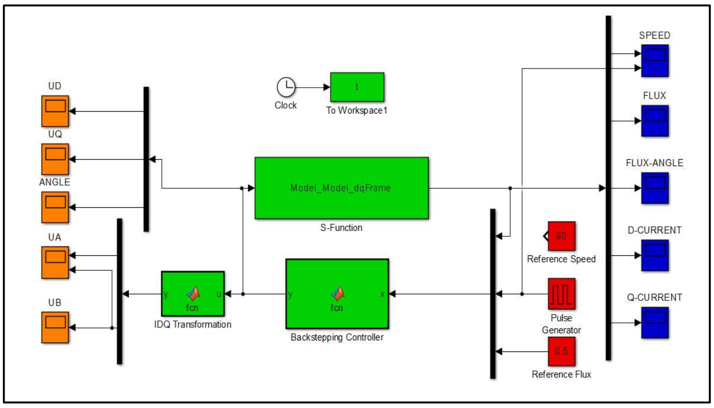

Fig.2: Simulink implementation of the backstepping controller.

The motor parameters used for the simulations are given in Table 1. It is essential to choose appropriate configuration parameters in the simulation module. The system was simulated with a fixed step ode4 (Runge-Kutta) with a step size of 10^{-3}. The controller gains were chosen as $g_1, g_2, g_3, g_4 = 5$. A constant flux reference of 0.5 Wb was chosen.

Table 1: Parameters for the induction machine

Motor Parameters	
Stator Resistance	1.34 ohms
Rotor Resistance	1.18 ohms
Mutual Inductance	0.17 H
Rotor Inductance	0.18 H
Stator Inductance	0.18 H
No. of Pole Pairs	2
Motor Load Inertia	0.0153 kgm^2

B. Results

Fig.3 and 4 shows the successful tracking of the flux magnitude and rotor speed with their respective references. After taking the inverse $d-q$ transform of the control variables (u_d and u_q), the output waveforms of the controller are presented in Fig.5. It was observed that the controller gains influence the system in two ways; firstly, higher controller gains force the system to reach the reference value faster, secondly; they increase the magnitude of the controller outputs. The choice of controller gains is constrained by the available power electronics and resources in real time. Optimization algorithms such as ‘particle swarm’ and ‘genetic algorithm’ with desired performance criteria can be used to obtain the optimal gains for the controller. Fig.6 is the plot of the stator voltages for the case of the tracking of a constant speed reference.

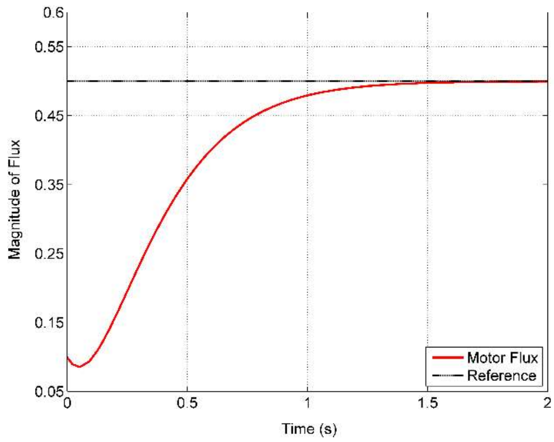

Fig.3: Flux magnitude tracking reference value.

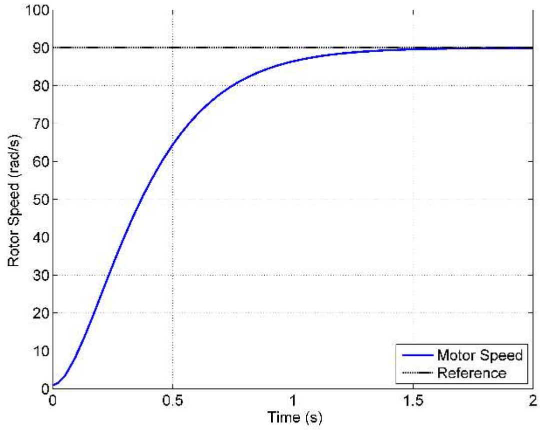

Fig.4: Rotor Speed tracking the reference value

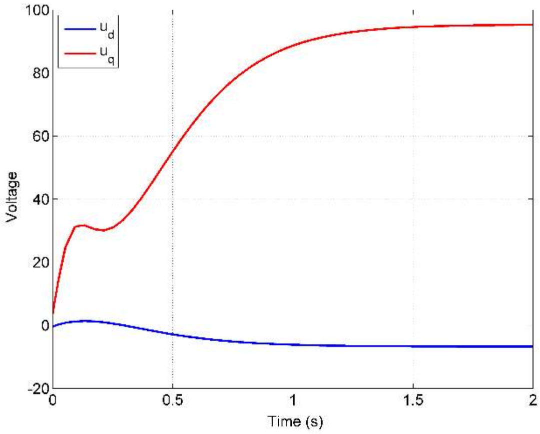

Fig.5: Controller Outputs

The case of a varying speed reference is presented in Fig.7.

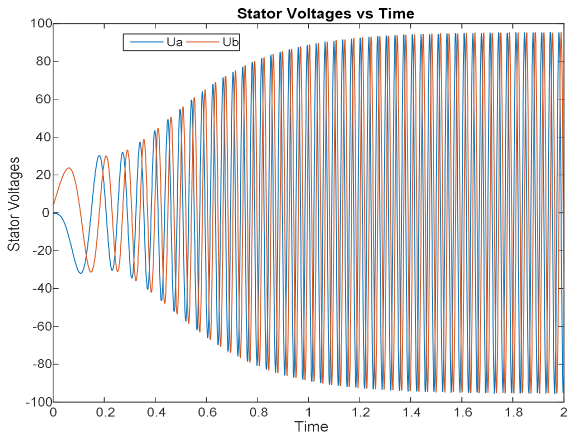

Fig.6: Plot of the stator voltages against time

The controller was able to track a periodic square wave of period 4sec. and pulse width of 50% as reference. Fig.8. shows the variation of the stator voltages with time

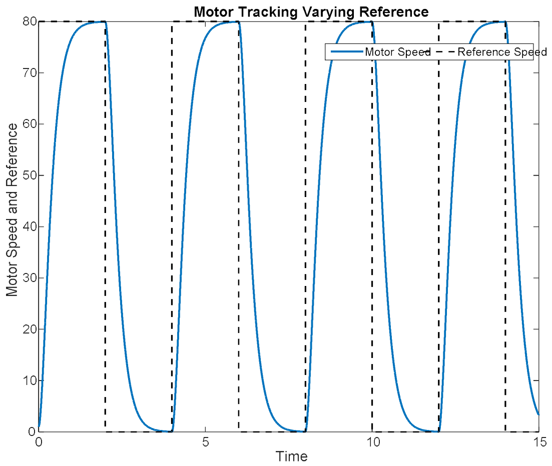

Fig.7: Motor speed tracking a varying reference

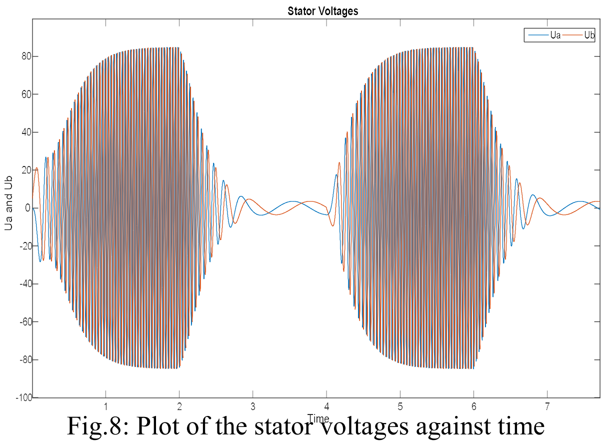

Fig.8: Plot of the stator voltages against time

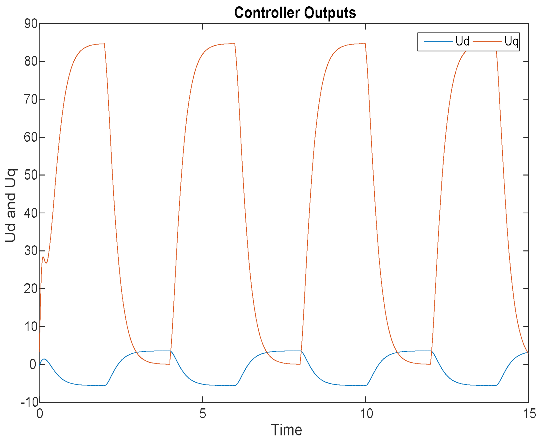

Fig.9: Controller Outputs

VI. SETUP FOR REAL-TIME IMPLEMENTATION

In this section, the experimental setup for the real time implementation of the proposed control system is proposed. The proposed setup consists of a software implementation part using Matlab and the required hardware setup. In the software implementation part, the different algorithms and transformations needed for real time system operation is discussed while the hardware components and devices required for implementation is detailed thereafter.

A. Hardware Setup

DSpace-DS1104 control board, connector and LED panel can be used to implement the control algorithm in real time. Six digital output channels are used to transmit a gate switching sequence to the power inverter. The power inverter is fed with a rail to rail supply of 400V DC. DSpace I/Os follows the TTL (0 – 5V) voltage levels in contract with the inverter gates which require +15V to turn on. An intermediary level shift circuit is necessary to map the TTL levels to 0 – 15V DC. A simple comparator circuit can perform this task using an operational amplifier. The output of the inverter is the two phase Sine wave which is fed to the motor for set point speed tracking.

The proposed nonlinear feedback controller needs the full state vector as input. Among five states of the system, only the two phase currents i_a, i_b and the speed ω can be measured using current sensors and shaft encoder. For the rest i.e. flux vectors (ψ_a, ψ_b), although there are sensors available for the measurement which can be installed inside the air gap of the stator, but due the complexity of installation and their effect on the overall reliability of the machine, they are not used. In general, we do not have physical measurement of the flux rotating inside the machine so we have to estimate it using the observer. Observer design for flux estimation is described in the next section.

Current sensing can be achieved using current transformers (CTs). At the output of the CTs, shunt resistors are used in series wherein voltage drops across the resistor is directly proportional to the current passing through it. This voltage drop is fed to the DSpace analog input channel for the phase current calculation. A shaft encoder generates digital pulses on its output due to shaft rotation. Depending on the encoder specifications, the rotor speed can be calculated easily. Details of the speed and the phase current calculation algorithms are not covered here.

Electrical machines are known to generate high frequency conductive noise due to their inductive nature and this may cause instrumentation errors and other problems in embedded circuitry – in this case the control board DS1104. It is necessary to build an optical or magnetic isolation barrier between the control circuit (Dspace Control board) and the power circuit (Inverter). Standard optocouplers can be used for isolation. Block diagram of the complete hardware setup is presented in Fig. 10.

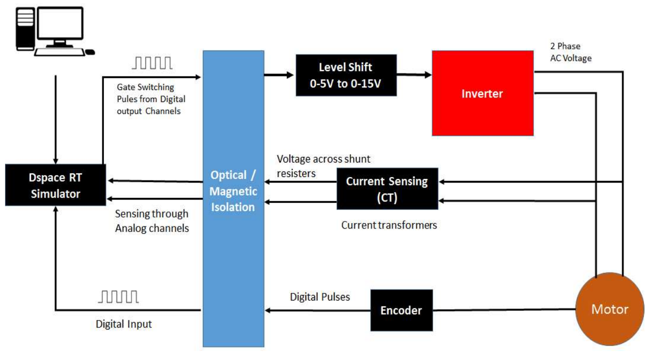

Fig.10: Block Diagram of hardware setup

B. Software Algorithms

After receiving the analog and digital signal from the current transformers and shaft encoder respectively. Rotor speed and phase currents are calculated (details of this are out of scope of this project). Then rotor flux is estimated using these measured quantities in d-q frame. The other states are also transformed to the d-q reference frame before using them for the controller. The nonlinear controller generates the required stator voltages in d-q frame (u_d, u_q) depending on the error between the current speed and the reference speed. The inverse d-q transformation is used to obtain the true stator voltages required for the machine. The real time reconstruction of these sine A backstepping controller was successfully designed to track steady speed reference, a varying speed reference and motor flux magnitude with a nonlinear motor model with viscous friction. The developed controller was simulated in Matlab and a technique for real-time implementation was proposed.

waves for the machine hardware is obtainable using an inverter as highlighted previously. This sine wave is sunk at the digital output port of the Dspace in the form of switching signals of the inverter switches. Block diagram of the software portion is presented in Fig. 11.

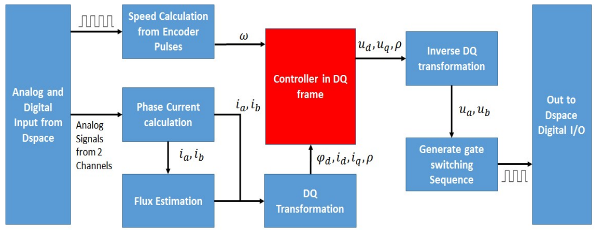

Fig.11: Block diagram of software implementation

VII. Conclusions

A backstepping controller was successfully designed to track steady speed reference, a varying speed reference and motor flux magnitude with a nonlinear motor model with viscous friction. The developed controller was simulated in Matlab and a technique for real-time implementation was proposed.

References

[1] P. Martin and P. Rouchon, “Two simple flux observers for induction motors,” *Int. J. Adapt. Control Signal Process.*, vol. 14, no. 2–3, pp. 171–175, 2000.

[2] A. Fekih and F. N. Chowdhury, “On nonlinear control of induction motors: comparison of two approaches,” *Am. Control Conf. 2004*, no. 1, pp. 1135–1140, 2004.

[3] W. Leonhard, *Control of Electrical Drives*. New York: Springer, 1985.

[4] J. Chiasson, “Nonlinear Controller for an Induction Motor,” *Control Eng. Pract.*, vol. 4, no. 7, pp. 977–990, 1996.

[5] P. Marino, R., Peresada, S., Tomei., “Adaptive input-output linearizing control of induction motors,” *IEEE Trans. Automat. Contr.*, vol. 38, no. 2, pp. 208–211, 1993.

[6] J. Chiasson, “Dynamic Feedback Linearization of the Induction Motor,” *IEEE Trans. Automat. Contr.*, vol. 38, no. 10, pp. 1588–1594, 1993.

[7] Cable-Online, “Squirrel Cage Induction Motor.” [Online]. Available: http://www.cable-online.cn/img/tx/squirrel-cage-rotor-induction-motor/electric_motor.jpg.

[8] M. G. Aydeniz and I. Senol, “A novel approach to sensorless control of induction motors,” in *International Conference on Power Electronics and Intelligent Control for Energy Conservation*, 2009, pp. 179–183.

[9] P. E, Leksono, “Adaptive speed control of induction motor with DSP implementation,” in *Industrial Electronics Society*, 2004, pp. 1423 – 1428.

[10] S. Rafa and N. E. ; A. L. ; M. M. ; A. H. ; L. Barazane, “Real-time implementation in DSPACE of intelligent vector control of induction motor,” in *3rd International Conference on Systems and Control*, 2013, pp. 451–456.

[11] J. Chiasson, “A New Approach to Dynamics Feedback Linearization Control of an Induction Motor,” *IEEE Trans. Automat. Contr.*, vol. 43, no. 3, pp. 161–167, 1998.

[12] R. Marino, S. Peresada, and P. Valigi, “Adaptive input-output linearizing control of induction motors,” *IEEE Trans. Automat. Contr.*, vol. 38, no. 2, pp. 208–221, 1993.

[13] H. Tan and J. Chang, “Adaptive backstepping control of induction motor with uncertainties,” *Am. Control Conf.*, vol. 1, no. June, pp. 1–5, 1999.

[14] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Pearson, 2001.