Available online at www.sciencedirect.com

SCIENCE DIRECT®

Engineering Applications of Artificial Intelligence 18 (2005) 57–63

Engineering Applications of ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Neural network modeling of torque estimation and *d–q* transformation for induction machine

Kaijam M. Woodley, Hui (Helen) Li, Simon Y. Foo*

Department of Electrical & Computer Engineering, FAMU-FSU College of Engineering, 2525 Pottsdamer Street, Tallahassee, FL 32310, USA

Received 27 May 2004

Abstract

This paper presents a neural network approach in modeling of torque estimation and Parks *d–q* transformation for an open-loop induction machine. The nonlinear approximation capability of neural networks makes it possible to map the Parks *d–q* transformation and torque estimation in an induction motor, which would otherwise require extensive complex calculations. The neural network simulation results will be compared to those of directly DSP calculated transformation and estimation. The results show improved performance with the neural network approach. We conclude that machine systems transformations and estimations can take advantage of the neural network technology for improved performance and cost reduction in the long run.

Keywords: Neural network; Feed forward back propagation; Weights; Bias; Torque estimation; Parks transformation

1. Introduction

Motor applications play an integral role in today's society. With this a key component of all motors is the overall efficiency of the system. Controlling motor speed with varying parameters has allowed motors to have a success in many different real-world applications where a varied motor speed is needed for the stability of a certain type of system. In applications that range from electrical vehicles to wind turbines it becomes of utmost importance to get optimal efficiency and stability of the motor systems. With improvements in the efficiency of the motor system, an overall cost reduction can be accomplished along with improved durability of the system. This paper will discuss the use of artificial neural networks for axes transformation for a three-phase induction machine. The use of an artificial neural network can be applied based on a simple understanding of how the operating regions of the torque speed for a variable speed drive system with a variable frequency and a variable voltage operate (Simões and Bose, 1995). Neural networks have shown better results when estimating or controlling nonlinear systems. From the torque speed curve, it is known that there is a relationship between the voltage and frequency that produces a torque and speed operating point that causes the machine to operate at its optimum efficiency (Simões and Bose, 1995; Shi et al., 2001; Theocharis and Petridis, 1994; Elmas and Ustun, 2003; Yi et al., 2003). In the proposed paper the artificial neural network will be implemented into an open-loop voltage hertz control for an induction machine. The complete system would have three artificial neural network blocks for its implementation. The first artificial neural network block will be used as a command voltage generator. This command voltage generator is basically an axes transformation or Park's transformation (two-phase rotating reference frame to three-phase transformation). Two inputs entered into the artificial neural network will be the q-axis voltage for the rotating frame (V_{qs}) and θ, which are based on the rotor electrical speed. The three outputs generated will be the three-phase voltages. The second

*Corresponding author. Tel.: +1 850 410 6474; fax: +1 850 410 6479.

E-mail address: foo@eng.fsu.edu (S.Y. Foo).

0952-1976/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2004.08.022

neural network block will be a three-phase to rotating d–q transformation block, used to generate control for the torque estimation block. The artificial neural network in the third block is used to predict the required peak rotor flux linkage (ψ_r), cos(θ), sin(θ), and the electromagnetic torque (T_e) required for control of the induction machine. In the third artificial neural network block will be used to mimic the operation of a torque transducer for the torque measurement of the electric motor.

The networks will be trained from data patterns generated from calculations using the general mathematical equations based on the equivalent circuit of the induction machine. The general mathematical equations are Parks three- to two-phase transformation, inverse Parks transformation and torque estimation equations. Feed forward back propagation training algorithm is used in training the neural network for the simulation parameters. The model will be validated by simulation using Matlab/Simulink and Neural Network toolbox.

The artificial neural network implementation of the transformation blocks provides a much faster execution time than traditional methods for control processing of the system. In real-world application this speed would provide better efficiency with loads that have a constant changing rotor speed. With such an improvement in the efficiency the overall system can have a drastic reduce in energy cost of a lengthy period of time. An artificial neural networks block also gets rid of the computational requirements for the transformations blocks.

2. Neural network

A neural network is a machine that is designed to model the way in which the brain performs a particular task (Haykin, 1999). Neural network machine models are mathematical based equations that allow derivable meaning from complicated or imprecise data, which can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques (Haykin, 1999). The general neural network structure used throughout the paper is feed forward back propagation algorithm consisting of three layers, an input layer, hidden layer, and output layer. The input and output layer consists of neurons that are determined by the input components and the output values. The hidden layer values will have a number of neurons that vary between 10 and 25 depending on the complexity of generating the output values (this will be discussed later on). The network from presimulated values through a learning process acquires knowledge. The weights are then used to store the acquired knowledge.

For the torque estimation, Parks transformation and inverse Parks transformation, each input layer uses a direct linear transition. The input activation function is given as (Haykin, 1999; Bose, 2002; Simões and Bose, 1995; Shi et al., 2001)

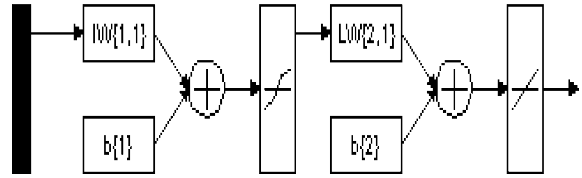

Fig. 1. Multilayer feed forward neural network.

$$v_j(n) = \sum_{i=1}^{n} w_{ij} \times x_i + \theta_i,$$

where w_{ij} is the weight matrix, x_i is the input vector, and θ_I is the bias function.

The hidden layer uses a hyperbolic tangent sigmoid transfer function as the transfer function (Haykin, 1999). These hidden layers allow the network to learn complex calculations by extracting progressively more meaningful features from the input patterns, through each successful iteration (Shi et al., 2001; Theocharis and Petridis, 1994). While the output layer uses a linear function. The setup of the neural network is shown in Fig. 1. With the use of hyperbolic tangent the neural network will have the capability to deal with nonlinear type calculations, like multiplications, divisions, and square roots (Simões and Bose, 1995). All such calculations are need for the torque estimation block and Parks transformation. The hyperbolic tangent is defined as (Haykin, 1999)

$$\varphi(v_j(n)) = \tanh(b \times v_j(n)),$$

where a and b are constants.

The weights and biases of the networks are of great importance because they store the information of the neural network. The training procedure used for the neural network as stated earlier is the feed forward back propagation supervised network.

3. Neural network training

With training data provided from a Matlab/Simulink open-loop induction machine the network can be trained. For initial training of the network the weights are first randomly initialized. With the random values of the weights and biases the output values are compared with the desired values that were gathered from simulated values of the open-loop induction machine (Haykin, 1999). With this comparison, the output layer neurons generate a sum-squared error. The sum-squared error for a set of N patterns is given as (Haykin, 1999;