Direct Torque Control for Matrix Converter Fed Three-Phase Induction Motor using an Artificial Neural Network Model

Phan Quoc Dzung
Faculty of Electrical & Electronic Engineering
HCMC University of Technology
Ho Chi Minh City - Vietnam

Le Minh Phuong
Faculty of Electrical & Electronic Engineering
HCMC University of Technology
Ho Chi Minh City - Vietnam

Nguyen Hoang Vu
Faculty of Electrical & Electronic Engineering
HCMC University of Technology

***Abstract*— This paper presents a new complete artificial neural network (ANN) of Direct – Torque Control (DTC) for the system Matrix converter (MC)– Induction Motor (IM) to decrease the time of calculation of the conventional control system and to observe another parameters such as motor speed . The ANN – System has one ANN – IM - Model and ANN- Controller- Model which are trained separately. A computer simulation program is developed using Matlab/Simulink together with the Neural Network Toolbox. The simulated results demonstrate the good quality and the robustness of the proposed ANN-DTC for MC-IM.**

Keyword:Direct Torque Control, Matrix Converter, Artificial Neural Network.

I. INTRODUCTION

Three- phase matrix converters (fig.1) have received considerable attention in recent years because they may become a good alternative to voltage- source inverter pulse-width-modulation (VSI-PWM) converters. In reality, the matrix converter provides important benefits such as bidirectional power flow, sinusoidal input current with adjustable displacement angle (i.e. controllable input power factor), and a great potential for size reduction due to the lack of dc- link capacitors for energy storage [3-5]. The direct torque control method (DTC) for induction motors has good behaviors such as robust and fast torque response, no requirements for coordinate transformation, no requirements for PWM pulse generation and current regulators [6-8]. The use of DTC strategy for control of the system MC-IM had been worked out with an excellent performance [1].

Because complicated calculations such as square root and trigonometric functions algorithm are involved, it is difficult to implement DTC using common IC hardware.

The DTC algorithm is usually implemented by serial calculations on a DSP board. However, as a predictive control scheme, the DTC has a steady- state control error produced by the time delay of the lengthy computations, which depends largely on the control algorithm and hardware performance.

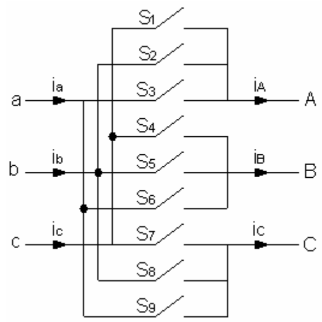


Fig. 1 Schematic representation of a matrix converter

A typical DSP (TMS32010) execution time of the DTC algorithm for a VSI- fed-IM is more than 250us [8]. ANN has faster parallel calculation and more simple circuit structure, so it is superior to a DSP board in execution time and hardware structure. The execution time of neural devices is less than 0.5us (analog) or 0.8us (digital) per neuron [10]. So, DSC of VSI fed Induction Motor based on ANN had been worked out [2, 11].

This paper presents a new ANN controller for MC-IM with one complete ANN- model of Induction Motor and 7 types of subnets for the control system. Comparing with the DSP serial calculations of the DTC system for MC-IM, the control precision of DTC can be significantly improved by using the ANN algorithm. Another hand, the ANN-IM-Model allows the estimation of important parameters of motor such as speed, stator flux, rotor flux, torque without using the sensors.

II. ANN-DTC CONTROL SYSTEM FOR MATRIX CONVERTER – INDUCTION MOTOR

In principle, the control technique of the matrix converter selects, at each sampling period, the proper switching configuration, which allows the compensation of instantaneous errors in flux magnitude, and torque, under the constraint of unity input power factor [1].

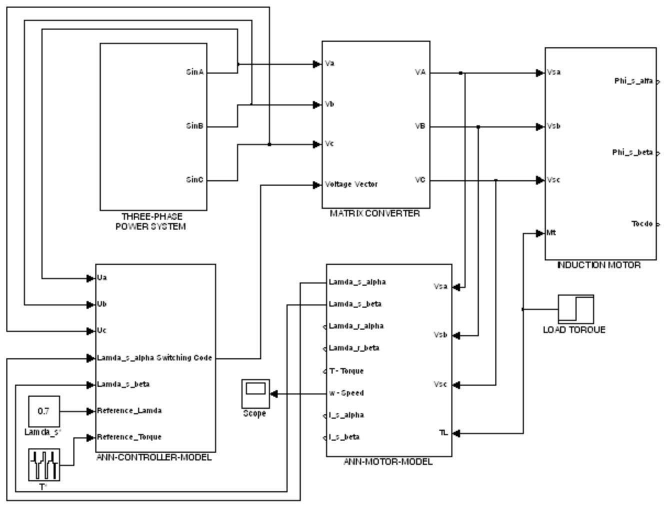


Figure 2. ANN-DTC Control System for Matrix Converter – Induction Motor

The DTC scheme consists typically of 3/2 transformations of current and voltage, torque calculation, flux estimation, flux angle encoder, flux magnitude calculation, hysteresis comparator for flux, torque and average of sinψI, basis DTC switching table, Matrix converter optimum switching table, input voltage angle encoder , switching configuration table.

Fig. 2 presents a ANN System for matrix converter fed induction motor. Based on this schema, the control system is divided in to an ANN-IM-Model and an ANN-Controller-Model, which are trained individually: 1) ANN Motor Model for Flux, Torque estimation and Speed, Rotor Flux and Stator current observer. 2) ANN Controller Model for flux magnitude calculation, hysteresis comparator sub-net, flux (or voltage) angle encoder, space voltage code DTC switching table, Matrix converter optimal switching table, switching configuration table, coordinate transformation.

A. ANN Motor Model(Fig.3)

The purpose of this model is to estimate the stator flux, motor torque and to observe the motor speed.

ANN - motor - model has 9 Sub-Nets for coordinate tranformation of stator voltage V_s, for determining stator flux $\lambda_{s\alpha}$, $\lambda_{s\beta}$, motor torque T, for estimating rotor flux $\lambda_{r\alpha}$, $\lambda_{r\beta}$, motor speed ω, also stator currents $i_{s\alpha}$, $i_{s\beta}$. This model consists of 4 inputs (stator voltages u_{sa}, u_{sb} , u_{sc} and load torque T_L) and 8 mentioned above outputs ($\lambda_{s\alpha}$, $\lambda_{s\beta}$, $\lambda_{r\alpha}$, $\lambda_{r\beta}$, T, ω, $i_{s\alpha}$, $i_{s\beta}$). The model consists of 9 Sub-nets :

1. Coordinate Transform Sub-net (3-2) : 2 purelin
2. Stator Flux $\lambda_{s\alpha}$ Estimation Sub-net (2-1) : 1 purelin
3. Stator Flux $\lambda_{s\beta}$ Estimation Sub-net (2-1) : 1 purelin
4. Rotor Flux $\lambda_{r\alpha}$ Estimation Sub-net (3-5-1) : 5 tansig, 1 purelin
5. Rotor Flux $\lambda_{r\beta}$ Estimation Sub-net (3-5-1) : 5 tansig, 1 purelin
6. Torque Estimation Sub-net (4-10-1) : 10 tansig, 1 purelin
7. Speed Estimation Sub-net (2-10-5-1) : 10 tansig, 5 tansig, 1 purelin
8. Stator current $i_{s\alpha}$ Estimation Sub-net (2-1) : 1 purelin

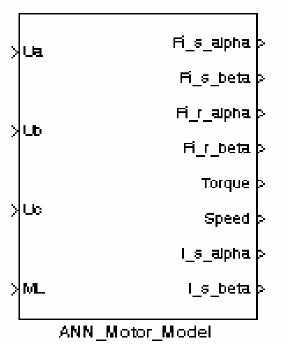


Figure 3. ANN-Motor-Model

9. Stator current $i_{s\beta}$ Estimation Sub-net (2-1) : 1 purelin

The datasheets for training have been received by simulation of motor model with dynamic model of IM (transient equations expressed by [12]). The Levenberg- Marquardt method is used for training (Table 1).

TABLE I. 9 SUB-NETS OF ANN-MOTOR-MODEL

Sub-net	Input(s)	Output(s)	Method	Epochs max	Error
1	V_{sa}, V_{sb}, V_{sc}	$V_{s\alpha}$, $V_{s\beta}$	Levenberg Marquardt	15000	$1e^{-15}$
2	$V_{s\alpha}$, $\lambda_{r\alpha}$	$\lambda_{s\alpha}$	L-M	15000	$1e^{-15}$
3	$V_{s\beta}$, $\lambda_{r\beta}$	$\lambda_{s\beta}$	L-M	15000	$1e^{-15}$
4	$\lambda_{r\beta},\lambda_{s\alpha},\omega$	$\lambda_{r\alpha}$	L-M	15000	$1e^{-15}$
5	$\lambda_{r\alpha},\lambda_{s\beta},\omega$	$\lambda_{r\beta}$	L-M	15000	$1e^{-15}$
6	$\lambda_{s\alpha},\lambda_{s\beta},\lambda_{r\alpha},\lambda_{r\beta}$	T	L-M	15000	$1e^{-15}$
7	T, T_L	ω	L-M	15000	$1e^{-15}$
8	$\lambda_{s\alpha}$, $\lambda_{r\alpha}$	$i_{s\alpha}$	L-M	15000	$1e^{-15}$
9	$\lambda_{s\beta}$, $\lambda_{r\beta}$	$i_{s\beta}$	L-M	15000	$1e^{-15}$

B.ANN Controller Model (Fig.4)

The purpose of this model is to realise the DTC method for controlling of MC-IM. ANN - controller - model has 7 Sub-Nets for Flux magnitude calculation, Hysteresis comparator, Flux Angle Encoder, Space Vector Code DTC switching Table, Matrix Converter Optimal Swiching Table, Decoder of Switching Configuration and Coordinate Transformation. This model consists of 7 inputs (power voltages u_a, u_b , u_c , $\lambda_{s\alpha}$, $\lambda_{s\beta}$, Referenced torque T^*, Referenced Startor Flux λ_s^*) and 1 output (The code of switching configuration for Matrix Converter).

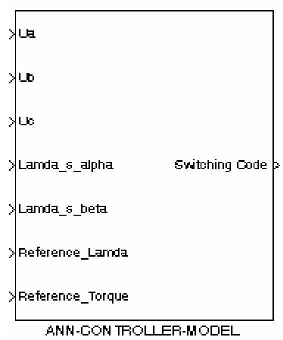
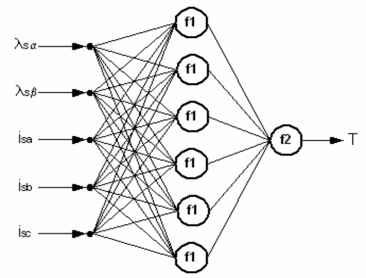


Figure 4. ANN-CONTROLLER- MODEL

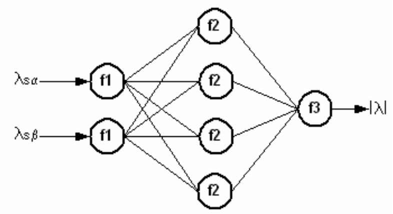
1) Flux magnitude calculation sub-net [2]

It is possible to use the training method that proposed by authors in [2] : Neural net includes 2 input square neurons, 4 tansig neurons and 1 purelin neuron. The back-propagation learning rule is used to train this net until it can approximate the square root function (Fig.5).



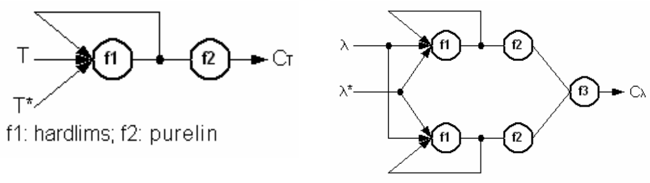
f1: square; f2: purelin

Figure 5. Torque calculation sub-net



f1:square; f2: tansig; f3: purelin

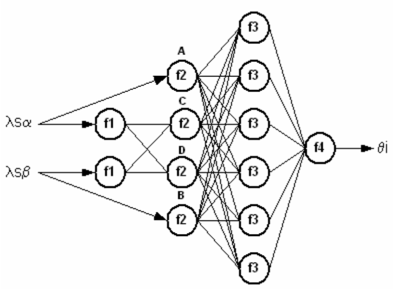
Figure 6. Flux magnitude calculation sub-net



f1: hardlims; f2: purelin

f1: hardlim; f2: purelin;f3: purelin

Figure 7. Hysteresis comparator sub-net



f1: square; f2: hardlims; f3: logsig; f4: purelin

Figure 8. Flux angle (or voltage angle) encoder sub-net

2) Hysteresis comparator Sub-Net [2]

The flux error $\varepsilon\lambda$, (torque error εT) between stator flux λ and its command λ^* (motor torque T and the torque command T*) can be limited within $\pm\Delta\lambda$ ($\pm\Delta T$) using the hysteresis comparator. The flux hysteresis comparator is implemented by a recurrent network with hardlim and purelin neurons (fixed – weight) (Fig.6).

3) Flux angle (or voltage angle) encoder sub-net

The purpose of this sub-net is to define the sector where the vector (flux, voltage) lies (Fig.7).

Neural net consists two networks that are trained individually. The first one includes 2 inputs ($\lambda_{s\alpha}$, $\lambda_{s\beta}$) and 4 sub-subnets for determining the codes A, B, C, D where the total number of neurons is 6 (2 square, 4 hardlims) (with fixed-weight) (Table 2).

The second network has 4 inputs (A, B, C, D), 1 hidden layer (6 logsig neurons) and 1 output (purelin). The m-file in Matlab program will be used for programming the algorithm (*trainlm* function – Levenberg –Marquardt algorithm). The code number of defined sector Θ_i (code) is output (table 3) (Θ_i=0.1 means sector 1).

TABLE 2. THE CODES OF FIRST LAYER

Code	A	B	C	D
Function	Sign($\lambda_{s\alpha}$)	sign($\lambda_{s\beta}$)	$sign\left(-\lvert\lambda_{s\alpha}\rvert+\sqrt{3}\lvert\lambda_{s\beta}\rvert\right)$	$sign\left(-\sqrt{3}\lvert\lambda_{s\alpha}\rvert+\lvert\lambda_{s\beta}\rvert\right)$
Activation function neuron	hardlims	hardlims	Square, hardlims	Square, hardlims
Weight	1	1	(-1, 3)	(-3,1)
Bias	0	0	(0,0)	(0,0)

TABLE 3 : THE CODE NUMBER OF DEFINED SECTOR

Inputs				**Output**
A	B	C	D	Θ_i
0,1	-1,0,1	-1	-1	0.1
0,1	0,1	0,1	-1,0,1	0.2
-1	0,1	0,1	-1,0,1	0.3
-1	-1,0,1	-1	-1	0.4
-1	-1	0,1	-1,0,1	0.5
0,1	-1	0,1	-1,0,1	0.6

Then, decrease the neural number of hidden layer continuously with fixed training program and fixed training error, at last the minimum number of neurons of hidden layer is 6 *logsig* neurons. So, the optimal total number of neurons is 7. After 124 training epochs, the sum squared error E is less than 10^{-15}.

4) Space voltage code DTC switching table sub-net

A two-layer network is employed to implement the optimum switching table. Torque control signal C_T, Flux control signal C_λ, number of sector θ_i (where the flux space vector lies) are inputs of network. The code of space vector voltage (i) is output (table 4).

The sub-net is obtained by training (supervised) with *trainlm* function – Levenberg –Marquardt algorithm, the acceptable for training squared error is 10^{-20}. The optimal number of neurons of 1st layer is 8 *logsig* neurons, the 2nd layer has 1 *purelin* neurons. So, the total number of neurons is 9 neurons (convergence obtained for 4357 epochs) (Fig.8).

TABLE 4 : BASIC OPTIMUM SWITCHING TABLE (SPACE VECTOR VOLTAGE CODE SELECTION)

Sector		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
C_T= +1	C_λ=+1	0.2 (V2)	0.3 (V3)	0.4 (V4)	0.5 (V5)	0.6 (V6)	0.1 (V1)
	C_λ=-1	0.3 (V3)	0.4 (V4)	0.5 (V5)	0.6 (V6)	0.1 (V1)	0.2 (V2)
	C_λ= 0	0.7 (V7)	0 (V0)	0.7 (V7)	0 (V0)	0.7 (V7)	0 (V0)
C_T= -1	C_λ=+1	0.6 (V6)	0.1 (V1)	0.2 (V2)	0.3 (V3)	0.4 (V4)	0.5 (V5)
	C_λ=-1	0.5 (V5)	0.6 (V6)	0.1 (V1)	0.2 (V2)	0.3 (V3)	0.4 (V4)
	C_λ= 0	0.7 (V7)	0 (V0)	0.7 (V7)	0 (V0)	0.7 (V7)	0 (V0)

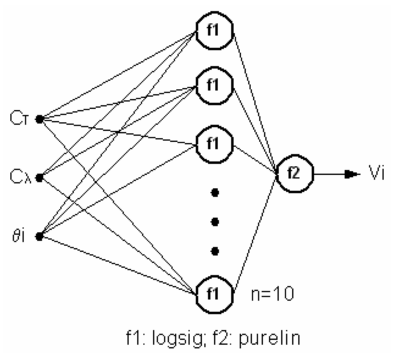


Figure 9. Space voltage code DTC switching table sub-net

TABLE 5. MATRIX CONVERTER OPTIMUM SWITCHING TABLE

	0.1		0.2		0.3		0.4		0.5		0.6	
C_ψ	+1	-1	+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
V_1	-0.3	0.1	0.2	-0.3	-0.1	0.2	0.3	-0.1	-0.2	0.3	0.1	-0.2
V_2	0.9	-0.7	-0.8	0.9	0.7	-0.8	-0.9	0.7	0.8	-0.9	-0.7	0.8
V_3	-0.6	0.4	0.5	-0.6	-0.4	0.5	0.6	-0.4	-0.5	0.6	0.4	-0.5
V_4	0.3	-0.1	-0.2	0.3	0.1	-0.2	-0.3	0.1	0.2	-0.3	-0.1	0.2
V_5	-0.9	0.7	0.8	-0.9	-0.7	0.8	0.9	-0.7	-0.8	0.9	0.7	-0.8
V_6	0.6	-0.4	-0.5	0.6	0.4	-0.5	-0.6	0.4	0.5	-0.6	-0.4	0.5
V_0, V_7	0	0	0	0	0	0	0	0	0	0	0	0

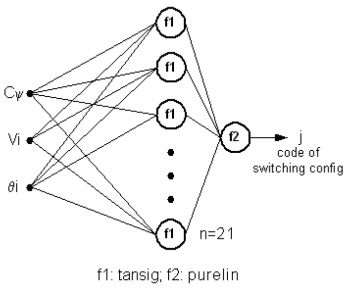


Figure 10. Matrix converter optimal switching table sub-net

5) *Matrix converter optimal switching table sub-net*

We can assume that V1 is the VSI output voltage vector selected by the DTC algorithm in the given switching period. In order to generate a voltage vector similar to V1, one of the matrix converter switching configurations ±1, ±2, ±3 must be chosen. Among the six vectors, those having the same direction of V1 and the maximum magnitude are considered. If the input line – to – neutral voltage lies in sector 1, then the switching configurations, which can be utilized, are +1 and –3. Both these switching configurations satisfy the torque and flux requirements. These configurations determine input current vectors lying on the directions adjacent to sector 1 and 4. Then, if the average value of sin(ψi) has to be decreased, the switching configuration –3 has to be applied. On the contrary, if the average value of sin(ψi) has to be increased, the switching configuration +1 has to be applied. The switching table based on these principles is shown in Table 5 [1]

A two-layer network is employed to implement the matrix converter optimum switching table, similar to previous sub-net. Control signal C_Ψ , space vector voltage V_i , code of θ_i are inputs of network. The code of switching configurations (j) is output (Fig.9).

The sub-net is obtained by training (supervised) with *trainlm* function – Levenberg –Marquardt algorithm, the acceptable for training squared error is 10^{-20}. The optimal number of neurons of 1st layer is 21 *tansig* neurons, the 2nd layer has 1 *purelin* neuron. So, the total number of neurons is 22 neurons (convergence obtained for 2148 epochs).

6) *Decoder of Switching configuration table sub-net*

This sub-net is implemented for the purpose of determining which switch to be fired for three phase, according to switching configuration (Table 6).

TABLE 6. DECODER OF SWITCHING CONFIGURATION TABLE

Switching configuration	Switch to be fired		
	Phase A	Phase B	Phase C
0.1	0.3	0.5	0.8
-0.1	0.2	0.6	0.9
0.2	0.2	0.4	0.7
-0.2	0.1	0.5	0.8
0.3	0.1	0.6	0.9
-0.3	0.3	0.4	0.7
0.4	0.2	0.6	0.8
-0.4	0.3	0.5	0.9
0.5	0.1	0.5	0.7
-0.5	0.2	0.4	0.8
0.6	0.3	0.4	0.9
-0.6	0.1	0.6	0.7
0.7	0.2	0.5	0.9
-0.7	0.3	0.6	0.8
0.8	0.1	0.4	0.8
-0.8	0.2	0.5	0.7
0.9	0.3	0.6	0.7
-0.9	0.1	0.4	0.9
0a	0.3	0.6	0.9
0b	0.2	0.5	0.8
0c	0.1	0.4	0.7

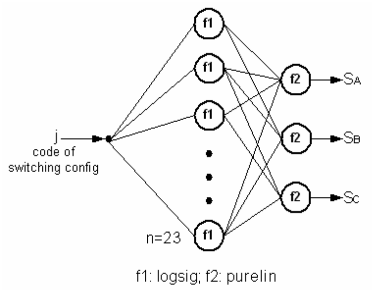


Figure 11. Decoder of switching configuration table sub-net

A two-layer network is employed to implement the decoder of switching configuration table, similar to previous sub-nets. The code of switching configuration is input of network. The numbers of switch configurations (k) are outputs (for example, k = 0.1 means S_1 in Fig.1).

Method of training is similar to the previous sub-net, the acceptable for training squared error is 10^{-20}. The optimal number of neurons of 1^{st} layer has 19 log*sig* neurons, the 2^{nd} layer has 3 *purelin* neurons. So, the total number of neurons is 22 neurons (convergence obtained for 3252 epochs) (Fig.10).

7) Coordinate transformation sub-net

The purpose of this sub-net is to transform the coordination abc to αβ. Neural net has one layer of 2 purelin neurons with fixed weights, where V_a, V_b, V_c are inputs, V_α, V_β are ouputs (Fig.11).

w_1 = [1 -1/2 -1/2]; θ_1 = [0]

w_2 = [0 sqrt(3)/2 -sqrt(3)/2]; θ_2 = [0]

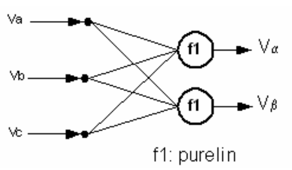


Figure 12. Coordinate transformation sub-net

III. SIMULATION OF ANN-DTC FOR MC-IM

A Simulink/Matlab program with the toolbox of neural – network is used to simulate the ANN-DTC for MC-IM with the above-mentioned sub-nets.

The induction motor model for the simulation studies has the parameters as following:

Type: three-phase, squirrel-cage induction motor.

20HP, R_s = 0.1062 (Ω), R_r = 0.0764 (Ω), X_s = 0.2145 (Ω), X_r = 0.2145 (Ω), X_m = 5.8339 (Ω), p = 2, J = 2.8 (kgm^2).

Reference flux λ_s* = 0.86 Wb.

Reference torque: T^* = 40 Nm when 0 s ≤ t ≤ 0.5 s; T^* = 100 Nm when 0.5 s ≤ t ≤ 0.8s; T^* = 20 Nm when 0.8 s ≤ t ≤ 1.2 s; T^* = -100 Nm when 1.2 s ≤ t ≤ 1.4s; T^* = 20 Nm when 1.4 s ≤ t ≤ 1.7s.

Time of simulation t = 1.7s.

Case study 1: Load torque TL = 20Nm.

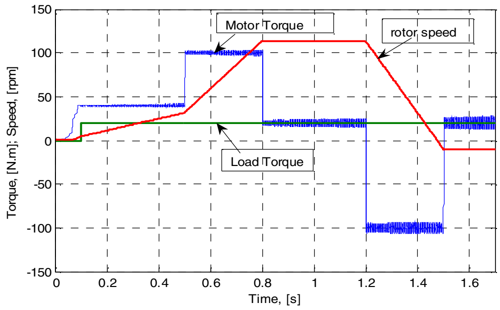


Figure 12. Torque and speed responses (case study 1)

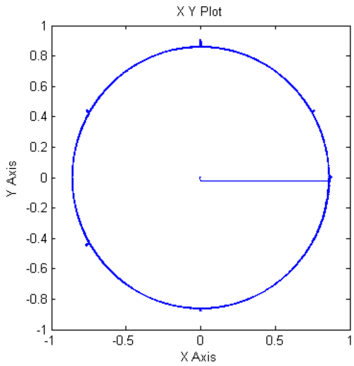


Figure 13. Locus of stator flux

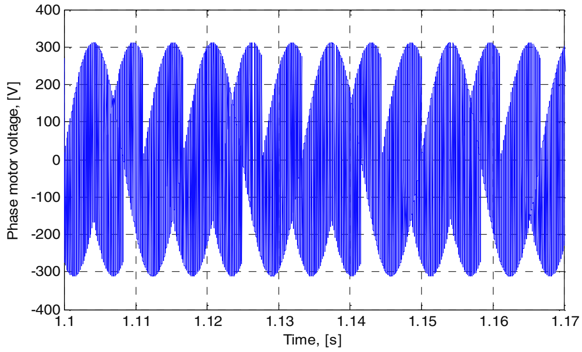


Figure 14. Phase motor voltage

Case study 2: Change of load torque : TL = 0 Nm when 0 s ≤ t ≤ 0.1 s; TL = 20 Nm when 0.1 s ≤ t ≤ 1s; TL = 40 Nm when 1 s ≤ t ≤ 1.5 s; TL = 20 Nm when 1.5 s ≤ t ≤ 1.7s.

In this case, the locus of stator flux is identical to Fig.13.

Case study 3: Load torque: TL = 20 Nm, rotor constant τr increases or decreases by 30%. The torque, flux and speed responses are identical as Fig. 12 -13.

Case study 4: Study of flux error caused by delay time of controller. Tdelay = 36μs for proposed ANN-controller (3μs to process each layer [10] for 12 layers in total) and Tdelay = 100 μs for typical DSP-controller) (Fig. 16, 17)

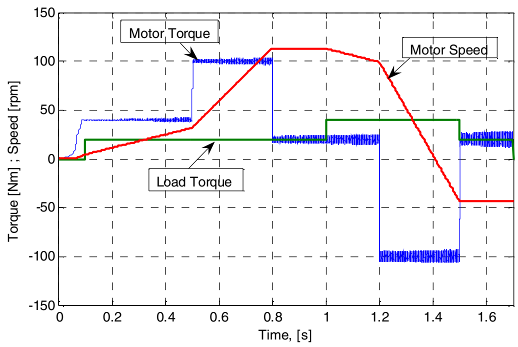


Figure 15. Torque and speed responses (case study 2)

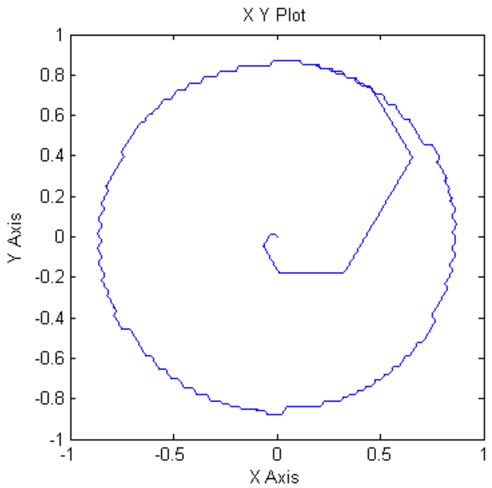


Figure 16. Flux response of ANN DTC with 36µs delay (time = 0s – 0.018s)

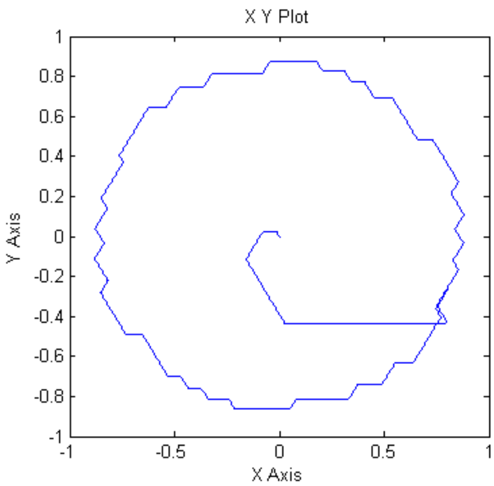


Figure 17. Flux response of DSP DTC with 100µs delay (time = 0s – 0.018s)

Simulation results demonstrate the excellent performance of the proposed ANN_DTC for MC-IM, while the good responses of the flux, torque, and speed are obtained (case study 1, fig.12 - 14). The proposed ANN-DTC also has the good robustness when the load torque T_L is changed (case study 2, fig.15); the rotor constant τ_r is changed (for example the value of τ_r is increased or decreased by 30%, case study 3). The simulation results also show that DSP-DTC produces more flux error than ANN-DTC (case study 4, Fig.16-17).

IV. Conclusion

This paper presents a new complete artificial-neural-network based direct–torque–control (ANN-DTC) scheme for an Matrix converter–fed Three-phase induction motor. Based on the understanding of DTC complexity (dynamic, recurrent, and nonlinear), the fixed weight and supervised methods with the training individually strategy are implemented for the controller design. Complete ANN system for MC-IM may be implemented by ASIC chip in the future.

Compared with the DSP based DTC, the proposed ANN-DTC scheme for Matrix converter –Induction motor incurs much shorter execution times and, hence, the flux and torque errors caused by control time delays are minimized.

Reference

[1] D. Casadei, G. Serra, A. Tani “The Use of Matrix Converters in Direct Torque Control of Induction Machines”, IEEE Trans. on Ind. Electron., vol.48, no. 6, December 2001.

[2] K. L. Shi, T. F. Chan, Y. K. Wong “Direct Self Control of Induction Motor Based on Neural Network”, IEEE Trans. on Ind. Appl., vol.37, no. 5, September/October 2001.

[3] A. Alesina, M.G.B. Venturini, “Analysis And Design of Optimum-Amplitude Nine – Switch Direct AC-AC Converters”, IEEE Trans. on Power. Electron., vol.4, Jan. 1989.

[4] P. W. Wheeler, J. Rodriguez, J. C. Claire, L. Empringham, A. Weinstein “Matrix Converters : A Technology Review” , IEEE Trans. on Ind. Electron., vol.49, no. 2, April 2002.

[5] D. Casadei, G. Serra, A. Tani, L. Zari “Matrix Converters Modulation Strategies : A New General Approach Based on Space-Vector Representation of the Switch State”, IEEE Trans. on Ind. Electron., vol.49, no. 2, April 2002.

[6] I. Takahashi, T. Noguchi, “ A New Quick- Response and High-Efficiency Control Strategy of an Induction Motor “, IEEE Trans. on Ind. Appl., vol. 1A-22, n0 5, pp. 820 –827, 1986.

[7] M. Depenbrock, “ Direct Self Control of Inverter – fed Induction Machine”, IEEE Trans. on Power. Electron., vol.3, Oct. 1988.

[8] T. G. Habetler, F. Profumo, M. Pastorelli, L. M. Tolbert, “Direct Torque Control of Induction Machines Using Space Vector Modulation”, IEEE Trans. on Ind. Appl., vol. 20, 1992.

[9] A. Delgado, C. Kambhampati, and K. Warwick, “Dynamic Recurrent Neural Network for System Identification and Control,” Proc. IEEE-Control Theory Applicat., vol.142, no.4, pp.307-314, 1995.

[10] M. E. Zaghloul, J. L. Meador, and R. W.Newcomb, “Silicon Implementation of Pulse Coded Neural Networks”, Kluwer Academic Publishers, 1994.

[11] P.Q.Dung, H.T.N.Thuong: Direct Torque Control for Induction Motor using ANN. The 2004 International Symposium on Advanced Science and Engineering, HCM City, Vietnam, pp. 60 - 63, 2004 .

[12] Peter Vas: Electrical Machines and Drives – A space-vector theory approach. Clarendon Press- Oxford. 1992.