Direct Torque Control of Induction Motors by Use of the GMR Neural Network

Giansalvo Cirrincione
Department of Electrical Engineering
University of Picardie-Jules Verne
33, rue Saint Leu
80039 Amiens - France
exin@u-picardie.fr

Maurizio Cirrincione
I.S.S.I.A.-C.N.R. Section of Palermo (former CE.RI.S.E.P.)
(Institute on Intelligent Systems for the Automation)
Viale delle Scienze snc,
90128 Palermo - Italy
nimzo@cerisep.pa.cnr.it

Chuan Lu
Katholieke Universiteit Leuven
Departement Elektrotechniek (ESAT)
Afd. ESAT SCD (SISTA), Kasteelpark Arenberg 10, B-3001
Heverlee,Leuven-Belgium
chuan.lu@esat.kuleuven.ac.be

Marcello Pucci
I.S.S.I.A.-C.N.R. Section of Palermo (former CE.RI.S.E.P.)
(Institute on Intelligent Systems for the Automation)
Viale delle Scienze snc,
90128 Palermo - Italy
pucci@cerisep.pa.cnr.it

***Abstract*— This paper deals with the application of the General Mapping Regressor (GMR) neural network to the direct torque control DTC of an induction motor. In particular it shows that the GMR neural network is able to correctly learn the classical DTC, as well as any other more involved control strategy. A suitable test bench has been set up in order to verify the performance of the neural controller.**

Keywords—Induction machines, electrical drives, direct torque control, neural networks.

I. INTRODUCTION

Modern control strategies of electrical drives generally involve the choice of a reference voltage to be generated by the power converter on the basis of the desired flux and torque of the motor. Accordingly several control strategies can be come up with e.g. the scalar control, the field oriented control (FOC) [1][2][3], the direct torque control (DTC) [2][3][4][5] or other control algorithms aimed to minimise the torque ripple or the losses of the drive or its conducted emissions (EMC = Electro-Magnetic Compatibility). In general this implies, on the basis of the working conditions of the drive, the proper choice of commands to be given to the power devices of the converter. This choice is essentially a pattern recognition problem where the input space is generally made up of the values of actual fluxes and torques. From this standpoint the property of neural networks to solve for pattern recognition problems and function approximation problems can be successfully employed in carrying out the control strategy. This paper deals with the application of GMR (Generalised Mapping Regressor) [6][7][8] neural network to the control of high performance electrical drive with induction motor. In particular a classical DTC control strategy has been learnt by the GMR network and then the performance of the neural controller has been verified both in numerical simulations and experimentally by means of a properly devised test bench. In [9][10][11] the DTC has also been used with “feed-forward neural networks” in order to substitute for the “optimal switching table”. As this is simply a pattern recognition problem, the GMR has been attempted to be used to show also its suitability for such task. Differently from most of the above previous papers, an experimental test bench has been set up and then the good working of the GMR has been experimentally verified.

Section II briefly describes the fundamentals of the GMR neural network. Section III summarises the classical DTC. Section IV shows how the GMR has been applied to the DTC. Section V describes the test bench suitably set up for the experimental verification. Section VI shows and discusses the simulation and experimental results.

Research of C.L. is supported by FWO: G.0407.02,KU Leuven: GOA-Mefisto 666 and DWCT. IUAP V-22.
Research of M.C. and M.P. has been funded by CNR.

II. Brief Description of the GMR Neural Network

GMR, whose architecture is shown in fig. 1, is an incremental self organizing (first layer weights) neural network with chains (second layer weights) among neurons. It transforms the mapping problem in a pattern recognition problem by working in the augmented space whose vectors are created by attaching to the input vector the corresponding output vector. Hence, the input in Fig. 1 is one of these augmented vectors. In this space, the mapping branches become clusters which have to be identified by the GMR linking phase. If, from one side, working in an augmented space is more difficult because of the curse of dimensionality, from the other side it allows to input each possible choice of vector components (reduced space) to the trained network in order to obtain, as output, the remaining components (recall phase): one of the advantages is the possibility to model both the mapping and its inverse. If the mapping is multivalued, GMR can output all values, also specifying to which mapping branch they belong. It is also able to output infinite solutions (as discretized equilevel hypersurfaces).

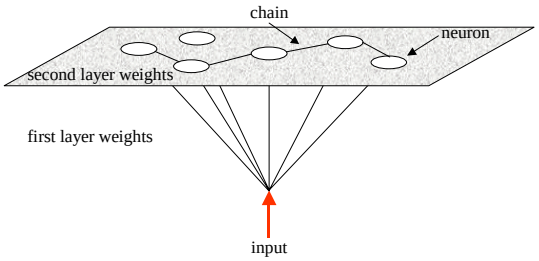


Fig. 1: GMR architecture

The algorithm is here briefly described, in a qualitative way (it is detailed in [8]). It is composed of four phases : training, linking, merging and recall. The training is incremental, competitive and self organizing. The algorithm EXIN SNN [6] is used : it recovers the augmented space by either creating neurons or adapting their weights according to the novelty of the input data. This novelty is quantified by a threshold, called vigilance threshold, which determines the resolution of the modelling. The neurons created after the first epochs (presentations of the whole training set, TS, to the network) are called object neurons (coarse quantization); then, data are labelled according to the nearest (in modulus) object neuron (object). In the next epochs (fine quantization) as many EXIN SNN's as objects are trained, by using data with the same label as TS's (multiresolution approach). A pool of neurons (final neurons) is found. Once trained, all data are relabelled according to the nearest final neuron. All data labelled according to the same neuron are defined as the neuron domain. For each domain, the principal direction (domain PD) of data is determined by using a neural approach for Principal Component Analysis [8]. In the linking phase, a link between two neurons is found at the presentation of each TS data in this way: the nearest (winning) neuron is linked to the neuron whose PD is the most similar to the winner's PD. A geometrical additional rule prevents from linking too far neurons. In the merging phase it is checked if different objects are linked: if the case, these objects are merged. The recall phase, which is explained in [8] replaces the neurons in the reduced space with Gaussians representing the domain. Their parameters are estimated by maximum likelihood. Simple tests and a Gaussian kernel interpolation determine all the possible outputs of the network. As a consequence of linking and merging, the corresponding mapping branches are also output.

III. The Direct Torque Control (DTC)

Nowadays in induction motor drives, along with field oriented controlled systems (FOC), the instantaneous torque control can be carried out by the use of direct torque control (DTC). This technique permits the direct and independent control of both electromagnetic torque and the flux linkage by selecting the optimum inverter switching modes so that the torque reference and the flux reference are tracked by the application of suitable stator voltages [2][3][4]. The implementation of the direct torque control, in order to perform the decoupling between the torque and flux control, requires information on the magnitude of the stator flux linkage space vector and on its angle with respect to the stationary reference frame. In classical DTC the knowledge of the sector in which the stator flux linkage lies is necessary while its exact angular position is not required. Even if the stator flux linkage can be directly measured, it is usually computed by a suitable flux model, which is based on the mathematical model of the motor. Any flux model requires, at each instant, the measurements of some stator quantities (stator voltages and currents) and the knowledge of some electrical parameters of the machine. In the following Fig.2 the classical DTC scheme with an induction machine is shown. The block "inverter optimal switching table" operates the generation of the stator voltage space vector, with a suitable choice of the switching pattern of the inverter, on the basis of the knowledge of the sector (supplied by the stator flux model block) in which the stator flux lies, and of the amplitudes of the stator flux and the torque.

In this scheme the closed loop control both of stator flux and rotor speed is performed. Speed control is performed by a PI controller whose output is the torque reference t_e^*. Flux and torque control is performed by hysteresis comparators whose bands are suitably chosen in order to reduce the flux and torque ripple. The flux model employed is based on the stator equations of the machine. The electromagnetic torque is estimated by the estimated stator flux components and the stator current components.

The employed control strategy is that shown in Tab. 1, where are shown the voltage space vectors to be applied when the stator flux linkage lies in the k^{th} the sector.

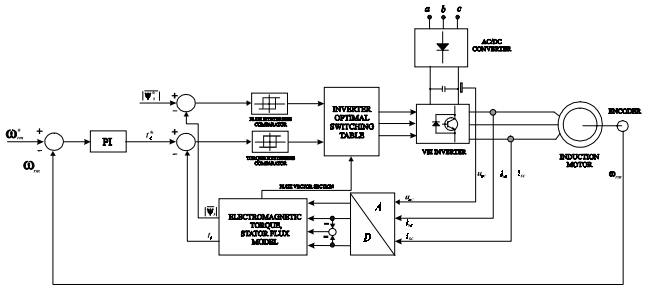


Fig.2: Block diagram of the classical DTC

TABLE I. ADOPTED CONTROL STRATEGY

	$t_e \Uparrow \mid \psi_s \mid \Uparrow$	$t_e \Uparrow \mid \psi_s \mid \Downarrow$	$t_e \Downarrow \mid \psi_s \mid \Uparrow$	$t_e \Downarrow \mid \psi_s \mid \Downarrow$
Control strategy	u_{k+1}	u_{k+2}	u_{k-1}	u_{k-2}

This control strategy has been chosen as it permits the best dynamic performance even in the breaking phase. The adopted sampling frequency of the control system is 10 kHz.

IV. THE GMR BASED DTC

As described in the introduction the target of this paper is to prove the suitability of the GMR neural network in applications to the control of induction motors. In general any control strategy which involves a pattern recognition problem can be implemented by a proper neural network as for example the GMR. In particular it is possible to implement any particular control strategy in dependence on predefined targets. In the case presented in this paper the DTC control strategy shown on Table I has been implemented. A GMR neural network has been devised having as inputs the torque error, the stator flux linkage error and the sector in which it lies, and as output the voltage space vector to be generated by the inverter. The GMR then replaces the "inverter optimal switching table" block of Fig. 2 as well as the hysteresis comparators, thus resulting in an easy implementation (Fig.3). However any other more involved control algorithm can be likewise implemented, e.g. predictive control, ripple torque minimisation, optimal efficiency algorithms, field oriented control, adaptive inverse control.

V. DESCRIPTION OF THE TEST BENCH

The DTC algorithm has been tested in simulation and experimentally. A test bench has been built for this purpose. The test bench consists of [12]:

- A three-phase induction motor with rated values shown in Table II;
- An electronic power converter (three-phase diode rectifier and VSI composed of 3 IGBT modules without any control system) of rated power 7.5 kVA.
- An electronic card with voltage sensors (model LEM LV 25-P) and current sensors (model LEM LA 55-P) for monitoring the instantaneous values of the stator phase voltages and currents;
- A voltage sensor (Model LEM CV3-1000) for monitoring the instantaneous value of the DC link voltage;
- An electronic card with analog 4^{th} order low-pass Bessel filters and cut-off frequency of 800 Hz;
- An incremental encoder (model RS 256-499, 2500 pulses per round);
- A dSPACE card (model DS1103) with a floating-point DSP.

Special care has been taken for the signal processing of the signals because anti-aliasing filters, low-pass filters and differentiators are necessary.

TABLE II. PARAMETERS OF THE INDUCTION MOTOR

Rated power P_{rated} [kW]	2.2
Rated voltage U_{rated} [V]	220
Rated frequency f_{rated} [Hz]	50
Pole-pairs	2
Stator resistance R_s [Ω]	3.88
Stator inductance L_s [mH]	252
Rotor resistance R_r [Ω]	1.87
Rotor inductance L_r [mH]	252
3-phase magnetizing inductance L_m [mH]	236
Moment of inertia J [Kg·m^2]	0.0266

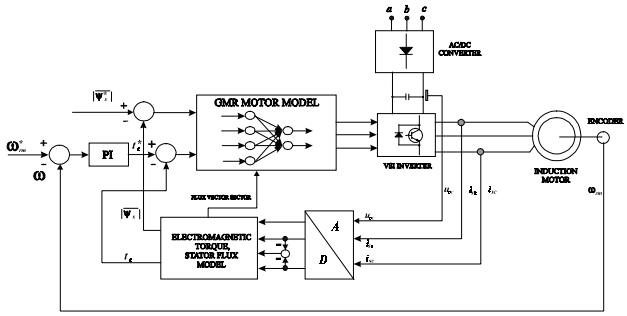


Fig. 3: GMR based DTC scheme

VI. SIMULATION AND EXPERIMENTAL RESULTS

The GMR based algorithm has been verified both in simulation and in experimentation by means of the test bench described in section V. Simulations have been performed by the Matlab®-Simulink® software, while in the experimental test the Dspace 1103 DSP (Digital Signal Processor) board has been programmed in the Matlab®–Simulink®-Real-Time-Workshop® environment.

The training set to give to the GMR has been chosen in order to comply with the "persistent excitation" theorem, which means that the frequency spectrum of the data should be as informative as possible both in frequency and in magnitude. In particular the network has as its inputs the difference between the desired value of the torque and estimated one, the difference between the actual amplitude of the stator flux and its reference, and the sector in which the stator flux linkage lies (discrete value ranging from 1 to 6). The output of the network is the voltage space vector to be generated (discrete value ranging from 1 to 6), which is directly linked to the switching pattern of the power devices of the inverter. More specifically the training set which has been used in experimentation is the one shown in Fig.4. The reference speed consists of negative and positive steps ranging from –150 rad/s to 150 rad/s. For each speed a load ranging from the rated load (± 10 Nm) to no-load has been given. The flux has been set to the rated value 0.8 Vs. Fig. 5 shows the corresponding torque as obtained by the test bench. By giving these references of speed and load to the drive, the corresponding data of the flux error, the sector in which the stator flux linkage lies, the torque error and the corresponding generated voltage space vector have been obtained. By choosing this last variable as the target and the other three variables as the three inputs, a collection of input/output data for the GMR has been obtained. The three input variables have been rescaled into [-1,1] before training and testing, while the output variable voltage vector still keeps the original scale value from 1 to 6. The 60% of these data have been picked up randomly to create the training set while the rest has been used as a test set. Fig. 6 shows the 3-dimensional input space of all the data of the training set. Fig. 7 shows the corresponding plot of the data of the test set. The GMR has been then trained with the following parameters: vigilance threshold of the coarse quantization = 0.3, vigilance threshold of the fine quantization = 0.02. The object neurons obtained are 130 and the pool of neurons after the fine quantization is composed of 1026 neurons. By using a merging threshold of one the objects are reduced to 75 as shown in Fig. 8. The GMR gives good results in the recall phase as shown in Fig. 9 where the data are well represented by the different branches. The GMR network can provide multiple solutions given the input data. 20 % of the output given by GMR are multi-valued, mostly 2-valued, which is also a character of the training set. Thus the GMR response is judged correct if the output voltage values given by the GMR network have included the target value. In this case an accuracy of 85% is achieved. Here no interpolation is used in the GMR recall phase: only the weights of the closest neurons to the input are considered. Though in this case the accuracy on the training set is still as high as 90%, the accuracy however on the test set goes down to 76%. This is obviously due to the wealth of information of the training set, but also to the fact that no interpolation has been used to speed up the algorithm. Future work will deal with the accuracy of the results. Fig. 10 shows the error on the output as obtained with the whole training set. Fig.11 shows the linking between neurons made by the GMR, with threshold 1, in the subspace obtained by using the three first principal component vectors obtained on the test set: here the linking within each cluster is apparent.

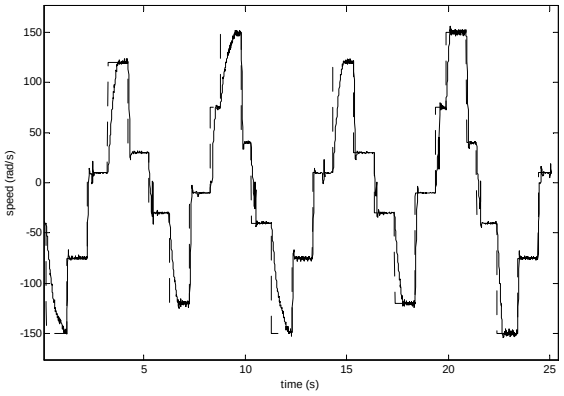


Fig 4: Reference and measured speed during the training phase (experiment)

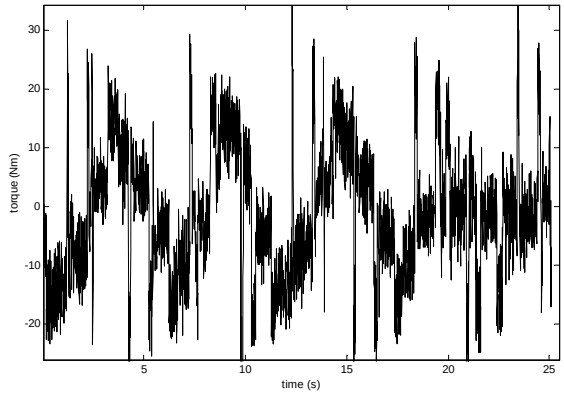


Fig 5: Electromagnetic torque during the training phase (experiment)

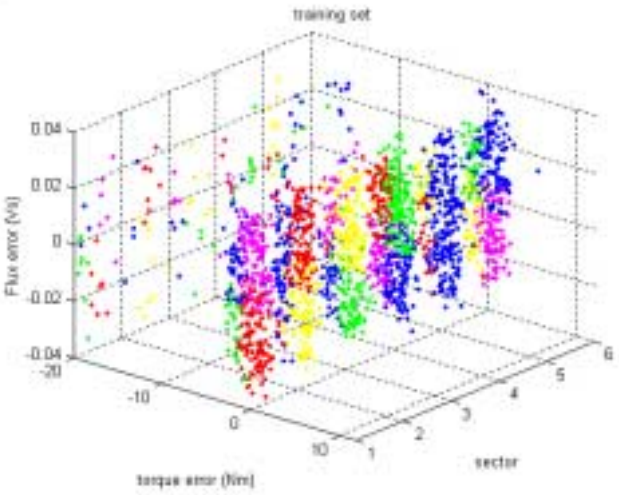


Fig 6: Input space of the training set

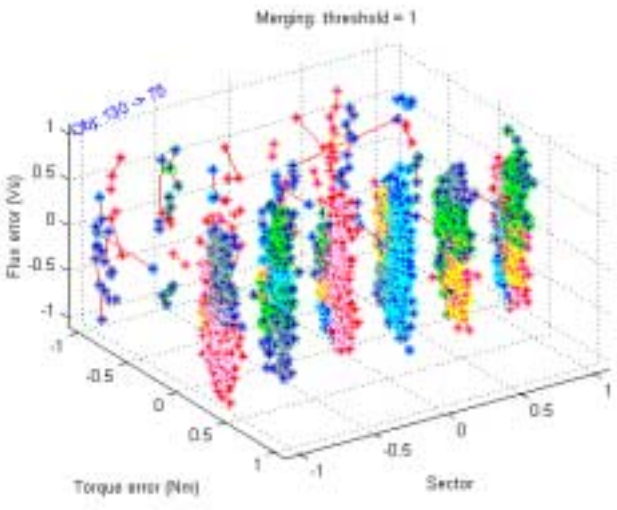


Fig 8: Input space after merging

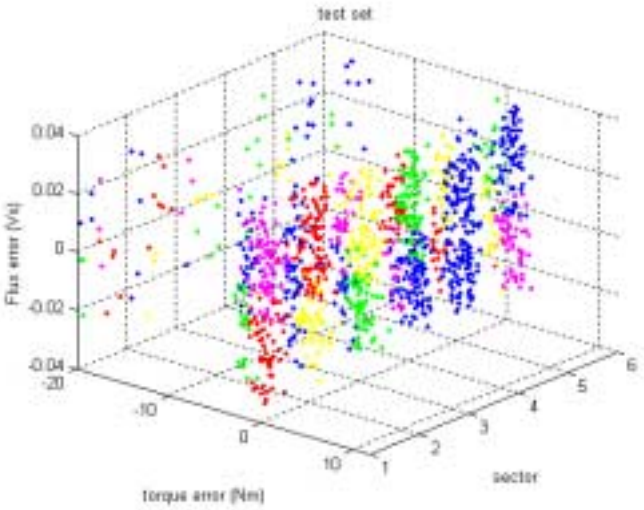


Fig 7: Input space of the test set

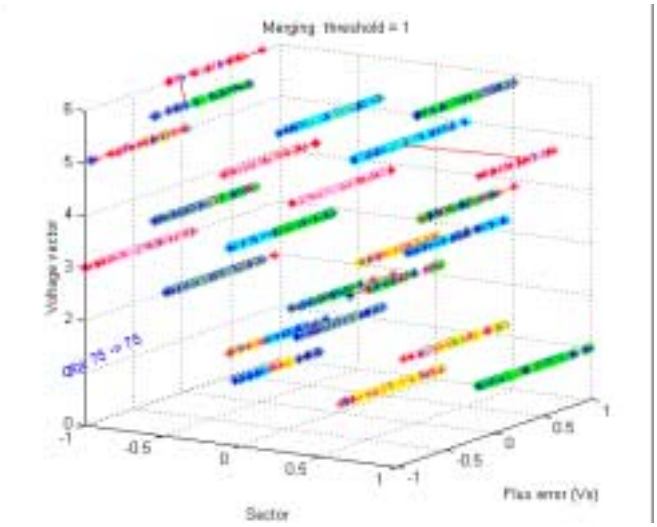


Fig 9: GMR mapping

VII. CONCLUSIONS

In this paper an application of the GMR to the control of an electrical drive with an induction motor has been presented. More specifically a particular strategy of the direct torque control has been taught to the GMR and the goodness of its learning capability has been verified by using experimental data obtained on a purposely developed test bench. The results show that this novel neural network gives at least the same results as those obtained in literature with other neural networks; but whereas those works aimed at solving an error minimisation problem by using well known feedforward neural networks, this work instead, considering that the task to be accomplished is the learning of a simple look-up table, exploits the pattern recognition capability of the GMR.

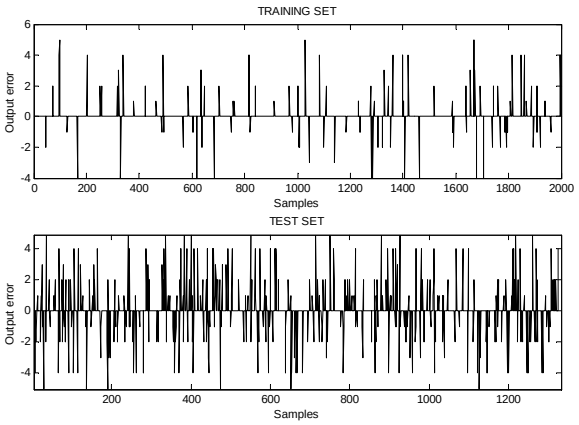


Fig. 10: Training and test set: output error between the output voltage vector with classical DTC and the GMR DTC

On the other hand the linking ability of the GMR as well as its potentiality in learning the inversion of relationships opens up also the possibility that it can be used also for more complex input-output control functions, e.g. adaptive inverse control, maximisation of efficiency, minimisation of torque ripple,. With this last regard the GMR can be used to control the duty-ratio of the power converter, which is a nonlinear function of the electromagnetic torque error, stator flux-linkage error and the position of the stator flux-linkage space vector. This nonlinear function can be easily implemented by GMR, which can determine the duty ratio during every switching cycle by proper recognition of the input space clusters so that the torque ripple can be minimised. Current work is on progress about all these last topics.

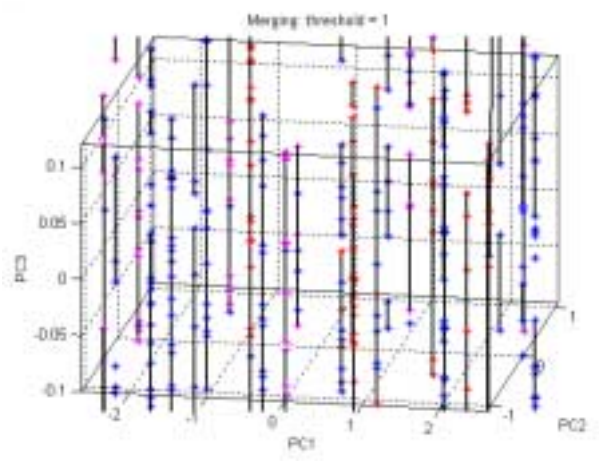


Fig.11 Linking of GMR on the test set in the principal component space

REFERENCES

[1] W. Leonhard, "Control of electrical drives", Springer, 1996.

[2] P. Vas, "Sensorless Vector and Direct Torque Control", Cambridge University Press, 1998.

[3] P. Vas, "Artificial-Intelligence-Based Electrical Machines and Drives", Cambridge University Press, 1999.

[4] Takahashi, Noguchi, "A new quick response and high-efficiency control strategy of an induction motor", IEEE Transactions on Industry Applications, 1986, 22.

[5] Depenbrock, "Direct self-control (DSC) of inverter fed induction machine", IEEE Transactions on Power Electronics, 1988, 3.

[6] G. Cirrincione, "Neural structure from motion", Ph.D. Thesis, LIS INPG, Grenoble, France, 1998.

[7] Chuan Lu, "The Generalised Mapping Regressor (GMR) Neural Network for Inverse Discontinuous Problems", Master of Artificial Intelligence Thesis, Katholieke Universiteit Leuven, Belgium, 2000.

[8] G. Cirrincione, M. Cirrincione, C. Lu, S. Van Huffel, "A Novel Neural Approach to Inverse Problems with Discontinuities (the GMR Neural Network)", accepterd in IEEE IJCNN 03.

[9] M. P. Kazmierkowski, "Control Strategies for PWM Rectifier/Inverter fed Induction Motors", IEEE ISIE 2000, vol. 1.

[10] L. A. Cabrera, M. E. Elbuluk, D. S. Zinger, "Learning Techniques to Train Neural Networks for Inverter-Fed Induction Machines Using Direct Torque Control", IEEE Transactions on Power Electronics, vol. 12, no.5, September 1997.

[11] C. Z. Cao, H. P. Li, "An Application of Fuzzy-Inference-Based Neural Network in DTC System of Induction Motor", 1st International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002.

[12] M. Pucci, Novel Numerical Techniques for the Identification of Induction Motors for the Control of AC Drives: Simulations and Experimental Implementations (in Italian), PhD Thesis, University of Palermo (Italy), January 2001.

ACKNOWLEDGMENT

All sections have been equally and jointly developed by the authors.