Hardware and Software Of A DSP-Based System for the Implementation Of AC Induction Motor Speed Drives

Mario Benedetti, Daniel Carrica and Juan Manuel Lorenzo

Laboratorio de Instrumentación y Control (LIC). Facultad de Ingenieria

Universidad Nacional de Mar del Plata

Juan B. Justo 4302; CP:7600 - Mar del Plata

ARGENTINA

e-mail: mbenedet@fi.mdp.edu.ar

Abstract: In this work a powerful development tool for the experimentation of motor control schemes is presented. It consists of a modular system based on a DSP and an associated algorithms menu.
The objective is to have a system that allows the implementation of new induction motor control strategies with minimal effort. A group of typical motor control routines has been developed so that in order to implement a control system it is only necessary to chain those programs. A group of routines is oriented to Vector Control. Thus, the user is free from programming the DSP in its assembler language and from knowing the structure of the development system in detail.

Keywords: experimentation system, vector control, DSP, motor control.

Mario Benedetti was born in Italy, in 1945. He received the Electronics Engineering degree from the Universidad Nacional de La Plata, La Plata, Argentina, in 1968. From 1968 to 1983, he worked at the Laboratorio de Electronica Industrial, Control e Instrumentación (LEICI) at the Universidad Nacional de La Plata (UNLP), developing electronic instruments. He was also Associate Professor at the Electrical Engineering Department of UNLP from 1970 to 1983, and he worked two years as a fellow at CERN, Geneva, Switzerland. Since 1985 he has been Professor at the Electrical Engineering Department of the Universidad Nacional de Mar del Plata, Argentina.
At present, he is the Director of the Laboratorio de Instrumentación y Control (LIC) at the Universidad Nacional de Mar del Plata. Dr. Benedetti is a member of the National Research Council (CONICET) of Argentina. He served as lecturer of numerous short courses to industry and other universities. His current research interests include Power Electronics and EMI Control.

Daniel Carrica was born in Dolores, Argentina, in 1958. He received the Electronics Engineering degree from the Universidad Nacional de Mar del Plata (UNMDP), Argentina, in 1984. In 1992 he received the MSc degree at the Universidad Politécnica de Madrid, España. From 1983 on, he has been with the Laboratorio de Instrumentación y Control (LIC), Universidad Nacional de Mar del Plata, Argentina, where he works on Power Electronics and Motion Control. He is Associate Professor of Power Electronics at the School of Engineering, UNMDP. From 1990 to 1991 he was at CERN (European Laboratory for Particle Physics), Geneva, Switzerland, as a Scientific Associate. From 1994 to 1996 he was the Director of the Electronics Department at the School of Engineering, UNMDP.
At present, he is the Co-Director of the LIC at the Universidad Nacional de Mar del Plata. Daniel Carrica is a member of the National Research Council (CONICET) of Argentina. His current research interests include Power Electronics and Motion Control.

Juan Manuel Lorenzo was born in La Plata, Argentina, in 1966. He received the Electronics Engineering degree from the Universidad Nacional de Mar del Plata (UNMDP), Argentina, in 1995. Since 1996 he has been working in the Motor Control area at the Laboratorio de Instrumentación y Control (LIC), UNMDP.

1. Introduction

In the past, the use of induction motors in motion control was limited to applications without great demands in their dynamic behaviour. In recent years the area of application of these motors has increased tremendously, also imposing in the areas where the DC motor seemed to be exclusive [1]. This increase has taken place thanks to technological advances in two areas: a) the development in the power semiconductors, specially with the advent of IGBT, MCT, GTO and power MOSFET, and b) an increase in speed and computation capacity of Microprocessors (MP) and Digital Signal Processors (DSP). The great development in both areas has allowed the implementation of control methods that satisfy demanding dynamic requirements [2].

This increase is reflected in a number of publications, conferences and symposia on the subject [3-9].

When a specialist makes a contribution to this area, he will inevitably require an experimental evaluation. Thus, it is necessary to implement a system consisting of an intelligent core, support circuits and acquisition modules. In specific publications of motion control, it can be seen that there is a big effort of experimental hardware development behind any new topology or control strategy.

Consequently, every time the specialist decides to innovate on the subject or evaluate and compare the different known alternatives, he faces the problem of building a new system. This effort, that is not directly related to the goals of the study, takes an important part of the project time. His attention is therefore dispersed in tasks which are not essential for the main objectives of his work.

Taking into account the situation described, it is of great advantage to have a universal experimentation platform on which different algorithms and control strategies can be tried out. The system should be powerful from the point of view of the hardware and, at the same time, easy to program. To have this tool available facilitates the experiment execution, shortens the testing time and diminishes its cost.

A system of these characteristics is presented in this paper.

2. Implementation

2.1 Hardware

The hardware is based on a previous work performed by Carrica [10], to which some changes have been made in order to increase its flexibility. The hardware of that system was also powerful, but it was conceived as a non-divisible structure. In contrast, this platform was designed on a modular concept. That is, this system is divided into independent modules that communicate with each other. In this way, any module can be modified without affecting the rest. This modular conception allows, for example, to keep the modules technologically updated.

The system is composed of four types of modules:

- DSP Module
- Acquisition Module
- Output Modules
- PC Module

The DSP communicates itself with an acquisition module, three output modules and one module that allows interaction with a PC (Figure 1).

The modular conception of the system increases its flexibility: each module can be replaced or updated without modifying the other ones.

The most salient characteristics of the modules are:

a) *DSP Module*: based on DSP 96002, which works with a 33MHz clock at a speed of 16.5 floating point MIPS and has 32 bits I/O ports. A DSP was chosen because it optimised the execution of MAC operations, that is, operations of the form $y_n = a_1 \cdot y_{n-1} + a_2 \cdot y_{n-2} + \ldots + b_0 \cdot x_n + b_1 \cdot x_{n-1} + \ldots$, very common in control algorithms. The high speed the DSP works with these operations allows it to carry out high-computation effort tasks within the sampling time. In this way, complex algorithms can be implemented, like state observers or sensorless control.

b) *Acquisition Module*: the acquisition system counts with 16 single ended input channels (8 differential pairs), 12 bits resolution and a maximum sampling rate of 500 Ksamples/second (A/D conversion time: 2 μs). This high rate allows sampling all the variables of interest within the execution time of the algorithms.

c) *Output Modules*: the system includes three specially designed output modules, so that the output signals can adapt to most situations. Each module has analog and digital outputs. Analog signals are obtained through 12-bit D/A converters and six TTL digital signals are available in case the converter switches are commanded directly by the DSP (e.g. PWM generation). The output range can be set between ±1.5V, ±2.5V, ±5V and ±10V in the differential mode, and 0V to 2.5V, 5V or 10V in the single ended mode.

d) *PC Module*: the interface with the user was solved using a PC. For example, through the PC the user can program the DSP, or can enter reference signals. The PC can also carry out the algorithms of slower control loops. A specific board was developed for the PC-DSP communication.

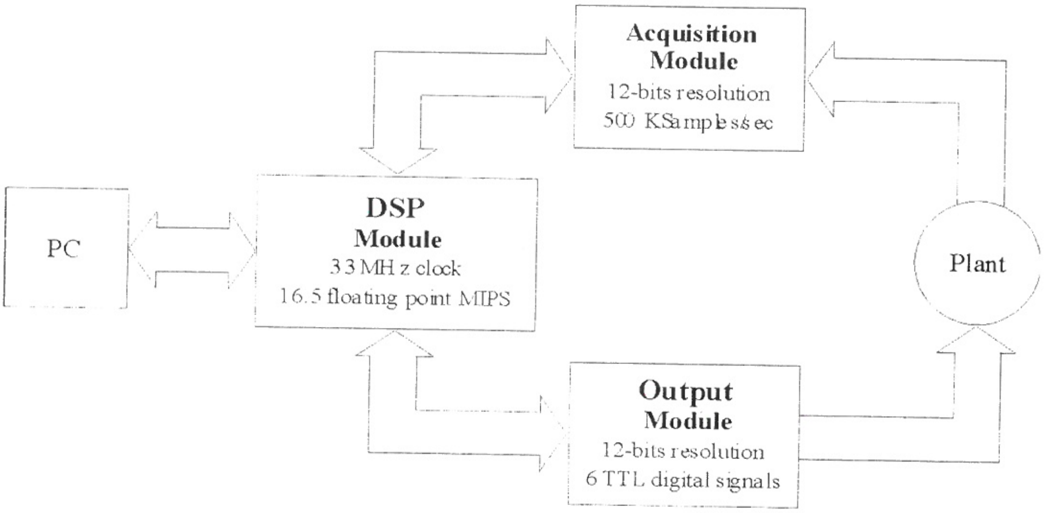

Figure 1. Modules That Form the Hardware of the Experimental Equipment

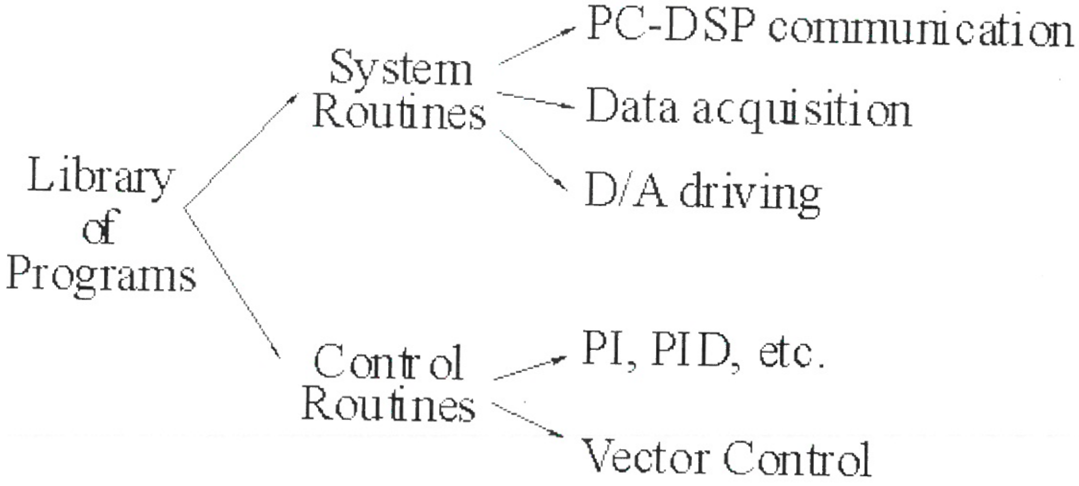

Figure 2. Library of Programs

2.2 Software

A Library of Programs and Algorithms was developed in an attempt to simplify the practice of new strategies and topologies. This library is made up of two groups (Figure 2). On the one hand, routines related with internal operations of the system, such as communication routines between the modules, A/D and D/A conversion-boards setting and driving, signal acquisition and signal conditioning, data format conversion: integer or floating-point. On the other hand, typical algorithms of the control of any plant in general and motor-control in particular.

In the first group, the programs that drive DSP-PC communication constitute a tool which is extremely useful since it allows a) to enter reference inputs to the system, b) to visualise different system variables: acquired or estimated ones, c) to visualise the signals that control the plant and, in general, all those variables of interest, d) to make files with these data for later processing.

The second group includes routines that are commonly found in control loops, from elemental addition or subtraction operations to elaborate algorithms, like PI or PID controllers or general transferences with M poles and N zeros. There were also included typical motor-control algorithms, like coordinate conversions type 2 to 3, 3 to 2, e^{ρ} and $e^{-\rho}$, which are utilised in Vector Control.

3. Use of the Library

The main objective of this Library is to prevent the user from deepening into the programming of a DSP as well as in the way of having access to the different modules that make up the hardware.

It was then decided the realization of independent routines and programs so that they could be linked with each other. In this way, the principal program that implements the algorithm of control to evaluate results in the appropriate succession of these routines. That is to say, that for the evaluation of a certain control algorithm, the user only has to define functional blocks that form its algorithm and assimilate each of them into one of the library routines.

3.1 Structures of the Programs

Basically, the routines that integrate the developed library are macros. It was decided to use this alternative because it is very simple to realise a program linking this type of structures.

As already mentioned, the algorithm to be evaluated should be divided into a series of functional blocks and then each of them is translated into a macro of the library. Thus, the principal program that implements the algorithm in question, results in a succession of macros.

Each of the macros contains a detailed description of the following points (Figure 3):

a) Action carried out by the macro

b) Meaning of the entrance parameters used

c) Registers and memory positions utilised

d) Registers and memory positions where the results are stored

The parameters inform the macro on the values and memory positions which they have to work on.

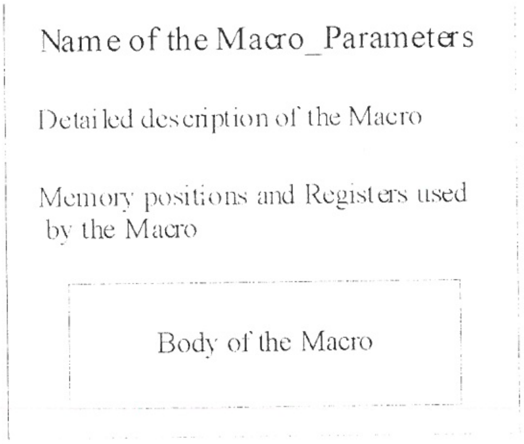

Figure 3. Structure of the Macros

4. Vector Control

Given the comparative advantages of the induction motors with respect to the dc ones, it is desirable to use them in the majority of possible applications. Control of these motors is more complicated than control of dc motors due to the fact that an induction motor is a non-linear, multivariable and with coupled system variables [11], [12], [13]. Certain algorithms, like Vector Control, simplify the control of induction motors. Specifically, Vector Control allows the controlling of the induction motor torque as in a dc motor, that is, through orthogonal and therefore mutually independent variables [14].

The inconvenience of these algorithms is that they require an important computational effort. This fact did not foster the use of induction motors in applications with demanding dynamic performance.

However, thanks to the development of fast processors, such as DSPs, those time-consuming motor control algorithms can now be implemented. Due to this, it was decided to develop a group of macros for the specific area of Vector Control.

4.1 Macros for Vector Control

The correct implementation of Vector Control depends on a crucial issue: the knowledge of the angle between the rotoric flux and the stator. This angle, usually called "ρ" rho (Figure 4), is not directly measured. It is calculated from the following expression:

$$\frac{\partial \rho}{\partial t} = \omega + \frac{i_{sq}}{\tau_R i_{mR}} \tag{1}$$

The current i_{mR} is calculated from expression (2):

$$\tau_R \frac{\partial i_{mR}}{\partial t} + i_{mR} = i_{sd} \tag{2}$$

All the variables of expression (1), and therefore "ρ", can be calculated starting from the stator currents and the motor angular speed. One of the created macros performs these calculations.

The speed can be directly sensed or can be calculated by an estimator. In the same way, the macros that perform Vector Control typical coordinate transformations have been developed (see Figure 4):

a) Decomposition of the stator current vector, whose original components are the three stator windings currents, in two orthogonal axes, one of them being aligned with one of the stator windings. This coordinate transformation is usually called *3 to 2 conversion.*

$$\begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix} = \begin{bmatrix} 3/2 & 0 & 0 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} i_{s1} \\ i_{s2} \\ i_{s3} \end{bmatrix}$$

b) Inverse conversion to that just mentioned, called *2 to 3 conversion*, that is to say, the macro calculates the three stator currents starting from the two orthogonal stator components.

$$\begin{bmatrix} i_{s1} \\ i_{s2} \\ i_{s3} \end{bmatrix} = \begin{bmatrix} 2/3 & 0 \\ -1/3 & 1/\sqrt{3} \\ -1/3 & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix}$$

c) Change from stator to rotor flux coordinates, or $e^{-j\rho}$ transformation:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho \\ \cos\rho & -\sin\rho \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix}$$

d) Change from rotor flux to stator coordinates, or $e^{j\rho}$ transformation:

$$\begin{bmatrix} i_{sa} \\ i_{sb} \end{bmatrix} = \begin{bmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}$$

To sum up, macros are included that implement all the functional blocks which are typical of Vector Control.

5. An Example of Application

It is the control system shown in Figure 5 that will be implemented. It corresponds to an induction motor speed control using Vector Control. The main program will result from identifying the basic functional blocks and replacing them by some adequate library macros. The program presented below (or a very similar one) will result:

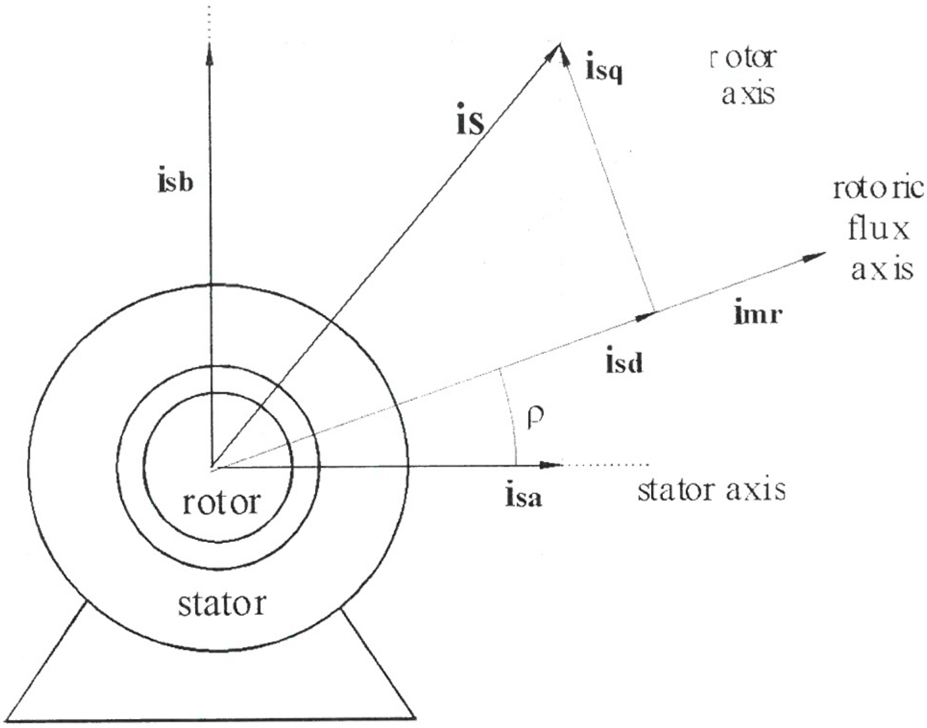

Figure 4. Vectors Used in Vector Control

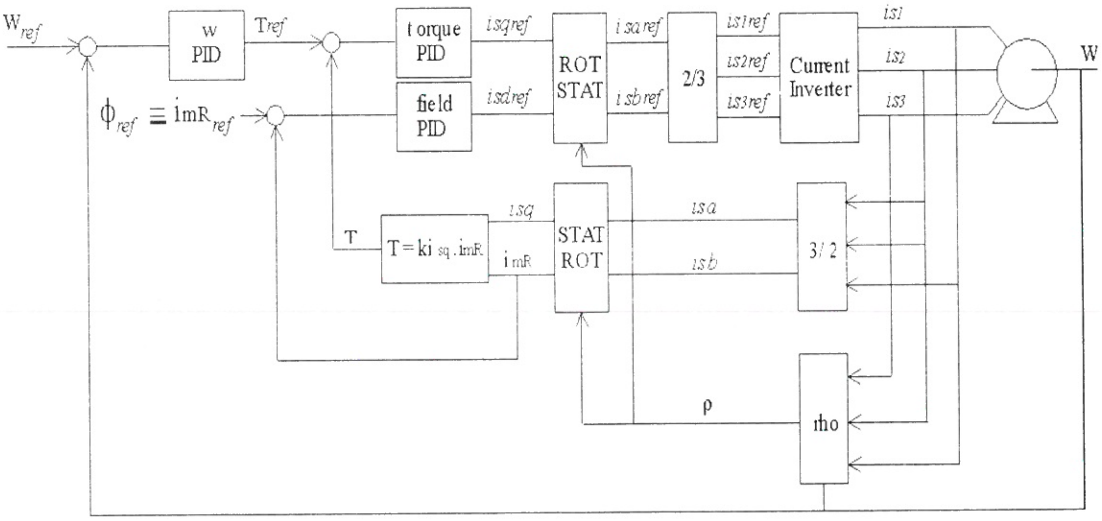

Figure 5. Control Scheme Implemented in the Example of Application

```
      INIT       ; user indicates to program in
                 what memory position each
                 variable will be stored.

start A/D 1      ; acquisition of one of the
                 stator currents (supposed to
                 be present in channel 1). The
                 start label indicates the
                 beginning of the loop.

      A/D 2      ; acquisition of second stator
                 current.

      IS3        ; this macro calculates the 3rd
                 stator current.

      A/D 7      ; acquisition of the angular
                 speed.

      THREE      ; stator current vector 3/2
      TO         conversion.
      TWO I

      RHO        ; rho calculus.

      STAT-      ; stator to rotor flux
      ROT        coordinate transformation.

      PID W      ; angular speed compensator.

      PID M      ; magnetic flux compensator.

      PID T      ; torque compensator.

      ROT-       ; rotor flux to stator
      STAT       coordinate transformation.

      TWO        ; stator current vector 2/3
      TO         conversion.
      THREE
      I

      D/A 1      ; is1 sent to D/A.

      D/A 2      ; is2 sent to D/A.

      D/A 3      ; is3 sent to D/A.

      jmp start  ; loop starts again from label
                 start.
```

This example clearly shows that implementing a control algorithm using the routines of the developed library is much simpler than programming in the DSP assembler language.

To have an idea of the speed of the system, the algorithm in the previous example, which includes three PID compensators, the complex calculation of rho, three changes of coordinates and two changes of coordinate system, takes about 40μs.

6. Conclusion

What was developed was a tool that largely facilitated the experimentation of Vector Control algorithms. To a very powerful and flexible system from the point of view of the hardware, a library of programs is added, that implements functional blocks usually found in control loops. A branch of this library was specially designed to implement induction motors Vector Control. All the programs have the characteristic of being modular, that is to say, they realise complete and independent actions. This modularity is fundamental, for it allows chaining the programs as if they were instruction lines. What was thus achieved was to simplify the experimentation of algorithms to a great extent, freeing the researchers from knowing not only the DSP programming but also the detailed way of access to the different boards that make up the hardware. This library can be perfectly amplified and updated.

Acknowledgments

The authors would like to thank CONICET and the National University of Mar del Plata, for their special support in the publication of this work.

REFERENCES

1. CARRICA, D., **Sistema Experimental de Control Vectorial de CA** (MSc. Thesis), Universidad Politécnica de Madrid, Madrid, 1992.

2. BOSE, **Power Electronics and Motion Control Technology Status and Recent Trends**, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 29, No. 5, 1993, pp. 902-909.

3. MURATA et al, **Vector Control for Induction Machine on the Application of Optimal Control Theory**, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 37, No. 4, August 1990, pp. 283-290.

4. CHEN et al, **Design of A Neural Fuzzy Controller for Induction Motor Speed Control**, IECON'94, 20th International Conference on Industrial Electronic Control and Instrumentation, Bologna, Italy, Vol.1, 1994, pp. 611-616.

5. CHEN et al, **Implementation of A High-Performance Induction Motor Drive Using A Three-Phase Inverter Module, A PWM ASIC and A DSP**, APEC '95, 10th Annual Applied Power Electronics Conference and Exposition, Dallas, USA, Vol.1, 1995, pp. 217-223.

6. HOLTZ, J., **On the Spatial Propagation of Transient Magnetic Fields in AC Machines**, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 32, No. 4, July/August 1996, pp. 927-937.

7. TAKANO, A., **Quick-Response Torque-Controlled Induction Motor Drives Using Phase-Locked Loop Speed Control With Disturbance Compensation**, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 43, No. 6, December 1996, pp. 640-646.

8. BURTON et al, **Identification and Control of Induction Motor Stator Currents Using Fast On-line Random Training of A Neural Network**, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol. 33, No. 3, May/June 1997, pp. 697-704.

9. MUTOH et al, **A Torque Controller Suitable for Electric Vehicles**, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 44, No. 1, February 1997, pp. 54-63.

10. CARRICA, D. et al, **Flexible System for the Experimentation of Field-oriented Control (FOC) Schemes in Induction Machines**, ECLA'94 Workshop on Advanced Automation, Madrid, Spain, 1994.

11. HOLTZ, J., **The Induction Motor - A Dynamic System**, IECON'94, 20th International Conference on Industrial Electronic Control and Instrumentation, Bologna, Italy, Vol.1, 1994, pp.1-6.

12. HOLTZ, J., **The Representation of AC Machine Dynamics by Complex Signal Flow Graphs Motor**, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 42, No . 3, June 1995, pp. 263-271.

13. BENEDETTI, M. et al, **Dinámica del motor trifásico de inducción controlado con orientación de campo**, AADECA'96 (Asociación Argentina de Control Automático) -7th Latin-American Automatic Control Congress, Buenos Aires, the Argentine, Vol.2, 1996, pp. 862-868.

14. BENEDETTI et al, **Basic Schemes for Vector Control of AC Induction Motors**, STUDIES IN INFORMATICS AND CONTROL, Vol.3, Nos. 2-3, September 1994, pp. 261-268.