Design and Implementation of PLC-Based Monitoring Control System for Three-Phase Induction Motors Fed by PWM Inverter

YASAR BIRBIR*, H.SELCUK NOGAY
*Marmara University, Technical Education Faculty, Department of Electricity Education
Goztepe, 34722 Istanbul, TURKEY.
ybirbir@marmara.edu.tr, selcuknogay@marmara.edu.tr

Abstract— **This paper presents a design and implementation of a monitoring and control system for the three-phase induction motor based on programmable logic controller (PLC) technology. Also, the implementation of the hardware and software for speed control and protection with the results obtained from tests on induction motor performance is provided. The PLC correlates the operational parameters to the speed requested by the user and monitors the system during normal operation and under trip conditions. Tests of the induction motor system driven by inverter and controlled by PLC prove a higher accuracy in speed regulation as compared to a conventional V/f control system. The efficiency of PLC control is increased at high speeds up to 95% of the synchronous speed. Thus, PLC proves themselves as a very versatile and effective tool in industrial control of electric drives.**

Keywords— **Computer-controlled systems, computerized monitoring, electric drives, induction motors, programmable logic controllers (PLC)**

1. Introduction

With the rapidly changes on industries and information technologies in recent years, some traditional bulk electronic appliances have to be monitored for a long time. All of their control devices such as communication interfaces gradually enter the Intemet information era. The control of all equipment has been performed through the use of computers. Most equipment uses PLC to connect with computers to monitor each load and electricity consuming devices. Programmable Logic Controllers (PLC) are widely used in industrial control because they are inexpensive, easy to install and very flexible in applications. A PLC interacts with the external world through its inputs and outputs. Since technology for motion control of electric drives became available, the use of programmable logic controllers (PLC) with power electronics in electric machines applications has been introduced in the manufacturing automation [1], [14]. This use offers advantages such as lower voltage drop when turned on and the ability to control motors and other equipment with a virtually unity power factor [3]. Many factories use PLC in automation processes to diminish production cost and to increase quality and reliability [4]–[7]. Other applications include machine tools with improved precision computerized numerical control (CNC) due to the use of PLC [7]. To obtain accurate industrial electric drive systems, it is necessary to use PLC interfaced with power converters, personal computers, and other electric equipment. Nevertheless, this makes the equipment more sophisticated, complex, and expensive. A power factor controller for a three-phase induction motor utilizes PLC to improve the power factor and to keep its voltage to frequency ratio constant under the whole control conditions. The vector control integrated circuit uses a complex programmable logic device (CPLD) and integer arithmetic for the voltage or current regulation of three-phase pulse-width modulation (PWM) inverters. Many applications of induction motors require besides the motor control functionality, the handling of several specific analog and digital I/O signals, home signals, trip signals, on/off/reverse commands. In such cases, a control unit involving a PLC must be added to the system structure. This paper presents a PLC-based monitoring and control system for a three-phase induction motor. It describes the design and implementation of the configured hardware and software. The test results obtained on induction motor performance show improved efficiency and increased accuracy in variable-load constant-speed-controlled operation. Thus, the PLC correlates and controls the operational parameters to the speed set point requested by the user and monitors the induction motor system during normal operation and under trip conditions [7, 14].

Manuscript Received April2, 2008; Revised Received September 24, 2008

2. PLC as a System Controller

Programmable logic controllers are modular, industrially hardened computers which perform control functions through modular input and output (I/O) modules. The modularity of PLC allows the user to combine generic I/O modules with a suitable controller to form a control system specific to his is most simply understood needs. The operation of a controller by envisioning that it repeatedly performs three steps:

a) Reads inputs from input modules

b) Solves preprogrammed control logic

c) Generates outputs to output modules based on the control logic solutions. Input devices and output devices of the process are connected to the PLC and the control program is entered into the PLC memory (Fig.1)[1-2]

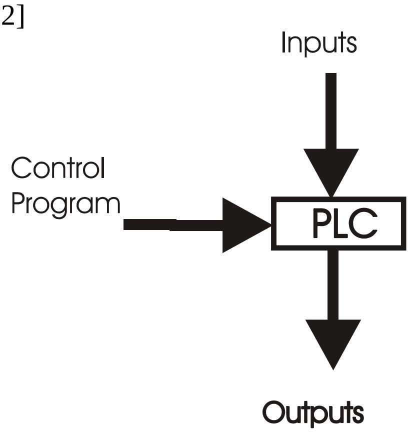

Fig. 1. Control Action of a PLC

In our application, it controls through analog and digital inputs and outputs the varying load-constant speed operation of an induction motor. Also, the PLC continuously monitors the inputs and activates the outputs according to the control program. This PLC system is of modular type composed of specific hardware building blocks (modules), which plug directly into a proprietary bus: a central processor unit (CPU), a power supply unit, input-output modules I/O and a program terminal. Such a modular approach has the advantage that the initial configuration can be expanded for other future applications such as multi machine systems or computer linking [2].

3. Control System of Induction Motor

The software models generated in the Software Requirements Analysis phase of the development project are refined and embellished in the design phase of the project. This phase involves making implementation decisions such as the interfacing between different software modules, the break down of software across multiple processors, assigning inputs and outputs to I/O cards, etc. PLC software, once written must be easy and intuitive to follow. PLCs are an integrated part of the domain system, advances in the technology of the system will effect the requirements of the PLC software. PLC software must therefore be maintainable and extensible [18].

In Fig. 2, the block diagram of the experimental system is illustrated. The following configurations can be obtained from this setup.

a) A closed-loop control system for constant speed operation, configured with speed feedback. The induction motor drives a variable load, is fed by an inverter and the PLC controls the inverter output.

b) An open-loop control system for variable speed and variable frequency operation. The induction motor drives a variable load and is fed by control mode. The PLC is an inverter in constant in activated.

c) The standard variable speed operation. The induction motor drives a variable load and is fed by a constant voltage-constant frequency standard three-phase supply.

The open-loop configuration b) can be obtained from the closed-loop configuration a) by removing the speed feedback. On the other hand, operation c) results if the entire control system is bypassed. [6-7]

PLC’s programming is based on the logic demands of input devices and the programs implemented are predominantly logical rather than numerical computational algorithms. Most of the programmed operations work on a straightforward two-state “on or off” basis and these alternate possibilities correspond to “true or false” (logical form) and“ 1 or 0” (binary form), respectively. Thus, PLCs offer a flexible programmable alternative to electrical circuit relay-based control systems built using analog devices.

The programming method used is the ladder diagram method. The PLC system provides a design environment in the form of software tools running on a host computer terminal which allows ladder diagrams to be developed, verified, tested, and diagnosed. First, the high-level program is written in ladder diagrams. Then, the ladder diagram is converted into binary instruction codes so that they can be stored in random-access memory (RAM) or erasable programmable read-only memory (EPROM). Each successive instruction is decoded and executed by the CPU. The function of the CPU is to control the operation of memory and I/O devices and to process data according to the program. Each input and output connection point on a PLC has an address used to identify the I/O bit. The method for the direct representation of data associated with the inputs, outputs, and memory is based on the fact that the PLC memory is organized into three regions: input image memory (I), output image memory (Q), and internal memory (M). Any memory location is referenced directly using %I, %Q, and %M (Table III).

The PLC program uses a cyclic scan in the main program loop such that periodic checks are made to the input variables (Fig.3). The program loop starts by scanning the inputs to the system and storing their states in fixed memory locations (input image memory I). The ladder program is then executed rung-by-rung. Scanning the program and solving the logic of the various ladder rungs determine the output states. The updated output states are stored in fixed memory locations (output image memory Q). The output values held in memory are then used to set and reset the physical outputs of the PLC simultaneously at the end of the program scan. For the given PLC, the time taken to complete one cycle or the scan time is 0, 18 ms/K (for 1000 steps) and with a maximum program capacity of 1000 steps. The development system comprises a host computer (PC) connected via an RS232 port to the target PLC. The host computer provides the software environment to perform file editing, storage, printing, and program operation monitoring. The process of developing the program to run on the PLC consists of: using an editor to draw the source ladder program, converting the source program to binary object code which will run on the PLC' s microprocessor and downloading the object code from the PC to the PLC system via the serial communication port. The PLC system is online when it is in active control of the machine and monitors any data to check for correct operation.

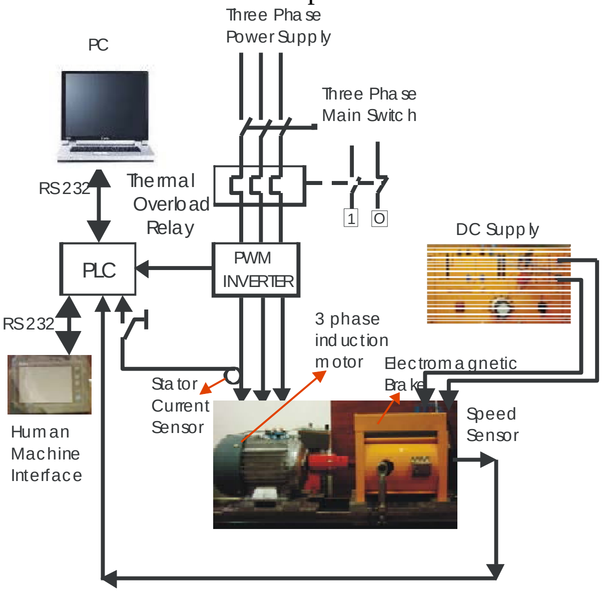

Fig. 2. Experimental setup

Table 1. Induction Motor Technical Specifications

Volt	Hz.	A	kW	Cosϕ	1/min.
Δ/220	50	4,7	1,1	0,8	1380
Y/380	50	2,7	1,1	0,8	1380

Table 2. Inverter Technical Specifications

Output Voltage (V)	Output Current (A)	Output Frequency (Hz)	Output Overload
380-460	1-6	0–150 Hz	150% 60s

Table 3. PLC Configuration

	Available	Used
Discreet Inputs (%I)	32	8
Discreet Outputs (%Q)	16	9
Analogue Inputs (%AI)	8	7
Analogue Outputs (%AQ)	8	6
Register Memory (%M)	500	

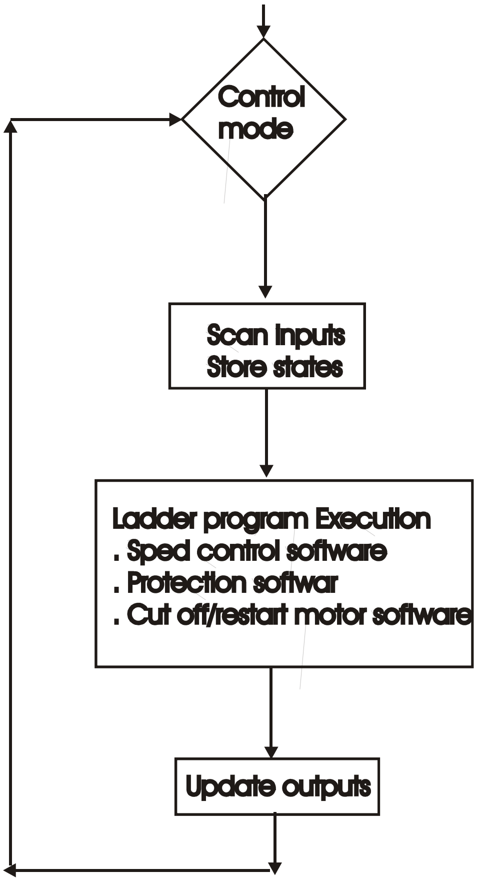

Fig.3. Flowchart of the main program

As a microprocessor-based system, the PLC system hardware is designed and built up with the following modules [7]:

- central processor unit (CPU);
- discrete output module (DOM);
- discrete input module (DIM);
- analog outputs module (AOM)
- analog inputs module (AIM)
- power supply.

InFig.4, the flowchart of the speed control software is illustrated. The software regulates the speed and monitors the constant speed control regardless of torque variation. The inverter being the power supply for the motor executes this while, at the same time, it is controlled by PLC' s software. The inverter alone can not keep the speed constant without the control loop with feedback and PLC. From the control panel, the operator selects the speed set point n_{sp} and forward/backward direction of rotation. Then, by pushing the manual start pushbutton, the motor begins the rotation. If the stop button is pushed, then the rotation stops.

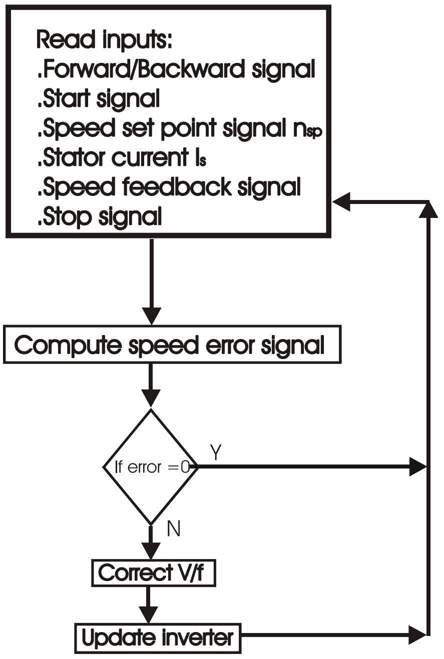

Fig. 4. Flowchart of speed control software

The corresponding input signals are interfaced to the DIM and the output signals to the DOM. The AIM receives the trip signal from the stator current sensor, the speed feedback signal from the tacho-generator, and the signal from the control panel. In this way, the PLC reads the requested speed and the actual speed of the motor. The difference between the requested speed by the operator and the actual speed of the motor gives the error signal. If the error signal is not zero, but positive or negative, then the PLC according to the computations carried out by the CPU decreases or increases the *V/f* of the inverter and, as a result, the speed of the motor is corrected.

The implemented control is of proportional and integral (PI) type (i.e., the error signal is multiplied by gain K_p, integrated and added to the requested speed). As a result, the control signal is sent to the DOM and connected to the digital input of the inverter to control *V/f* variations. At the beginning, the operator selects the gain K_p, by using a rotary resistor mounted on the control panel (gain adjust) and the AIM receives its voltage drop as controller gain signal (0–10 V).

The requested speed n_{sp} is selected using a rotary resistor and the AIM reads this signal. Its value is sent to the AOM and displayed at the control panel (speed set point display). Another display of the control panel shows the actual speed computed from the speed feedback signal. In Fig. 5, the flowchart of this software is shown. During motor operation, it is not possible to reverse its direction of rotation by changing the switch position. Before direction reversal, the stop button must be pushed. For motor protection against overloading currents during starting and loading, the following commands were programmed into the software.

i) Forward/backward signal is input to DIM.

ii) Speed set point signal n_{sp}, the stator current I_S , and the speed feedback signal are input to AIM.

iii) At no load, $I_S \le$ 2,5 A, if the speed set point is lower than 20% or n_{sp}< 300 r/min, the motor will not start.

iv) At an increased load over 3,2 Nm (40% of rated torque), $I_S \ge ,3\ A$ and a speed set point lower than 40% or n_{sp}< 600 r/min, the motor will not start.

v) If the load is increased more than 8,0Nm (rated torque) $I_S \ge$ 4 A and if the speed set point exceeds 100% or $n_{sp} \ge 1500$ r/min, the motor enters the cut off procedure.

vi) In all other situations, the motor enters in the speed control mode and the speed control software is executed as described in Subsection A.

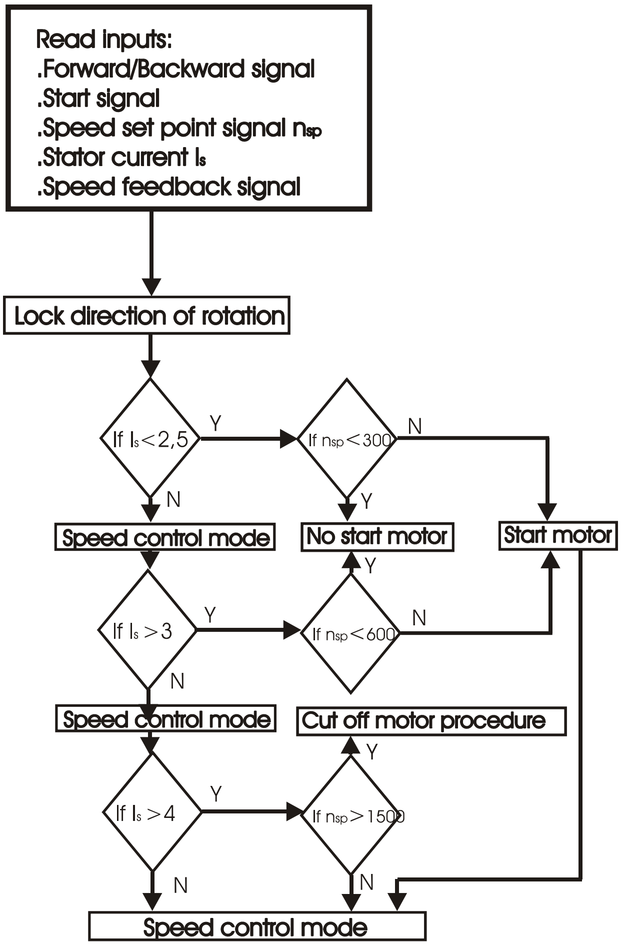

Fig. 5. Flowchart of monitor and protection Software

In Fig. 6, the flowchart of this software is shown.

- In overloading situations, the motor is cut off and the trip lamp (yellow) is lit. The operator must release the thermal relays and then must turn off the trip lamp by pushing trip or stop button. The thermal relays are set to the motor rated current 4 A. Following this, the motor can be started again.
- The motor can be cut off by the operator pushing the stop button: the display of the actual speed is set to zero, the start lamp (green) turns off, and the stop lamp (red) turns on and remains lit for 3 s.
- The load must be disconnected immediately after the motor cuts off and before the drive system is restarted. The motor will not start before 3 s after cutoff even if the start button is pushed.

4. Results

The system was tested during operation with varying loads including tests on induction motor speed control performance and tests for trip situations. The PLC monitors the motor operation and correlates the parameters according to the software. At the beginning, for reference purposes, the performance of induction motor supplied from a standard 380V, 50-Hz network was measured. Then, the experimental control system was operated between no load and full load (8, 0 Nm) in the two different modes described in Section 3:

a) induction motor fed by the inverter and with PLC control;

b) induction motor fed by the inverter.

The range of load torque and of speed corresponds to the design of the PLC hardware and software as described in the previous sections. The speed versus torque characteristics were studied in the range 500–1500 r/min and are illustrated in Fig. 7. The results show that configuration b) operates with varying speed-varying load torque characteristics for different speed set points n_{sp}. Configuration a) operates with constant-speed-varying load torque characteristics in the speed range 0–1400 r/min and 0–100% loads. However, in the range of speeds higher than 1400 r/min and loads higher than 70%, the system operates with varying-speed-varying-load and the constant speed was not possible to be kept. Thus, for $n_{sp} \geq 1400$ r/min both con-figurations a) and b) have a similar torque-speed response. This fact shows that PI control for constant speed as implemented by the software with PLC is effective at speeds lower than 93% of the synchronous[4-7].

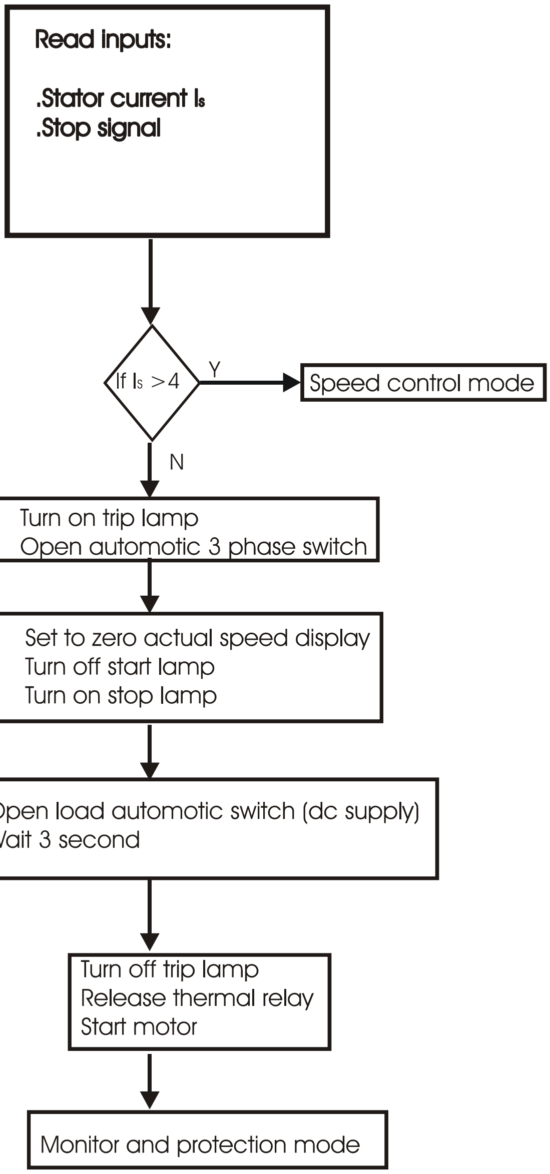

Fig. 6. Flowchart of cutoff/restart motor software

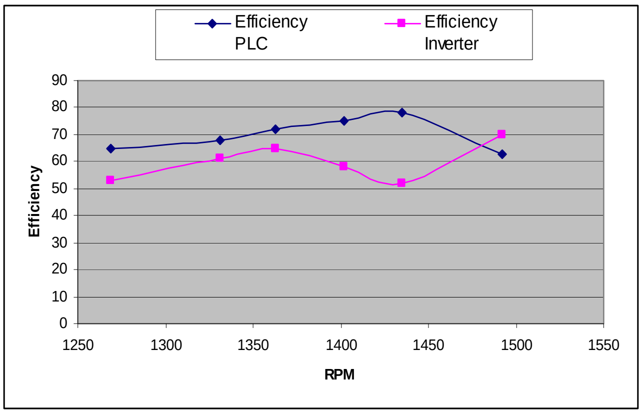

Fig. 8. Efficiency of controlled system

The efficiency for different values of n_{sp} was also studied. As depicted in Fig. 8, the results show that configuration a) in all cases has a higher efficiency than configuration b). Also, at operation with loads higher than 70%, the normalized efficiency is $\eta(pu)>1$, meaning that the obtained efficiency with PLC control is higher than the efficiency of induction motor operated from the standard 380-V, 50-Hz network without the control of PLC and without the inverter. According to this figure, the efficiency of PLC-controlled system is increased up to 10–12% compared to the standard motor operation. From a theoretical point of view, if we neglect magnetizing current, an approximate value for the efficiency is

$$\eta_s = \frac{1-s}{1-s\,R_S\,/\,R_R} \qquad (1)$$

where is the slip and R_S and R_R are the stator and rotor winding resistances, respectively. As can be seen from Fig. 7, the PLC controlled system a) works with very low slip values, almost zero. In all speed and load torque conditions, the configuration a) has a smaller slip than configuration b), thus the higher values of efficiency can be justified and especially at high speeds and frequencies. At lower frequencies, the magnetic flux increases and, thus, there is an increase in magnetizing current resulting in increased losses. This system presents a similar dynamic response as the closed-loop system with *V/f* speed control. Its transient performance is limited due to oscillations on torque and this behavior restricts the application of this system to processes that only require slow speed variation.

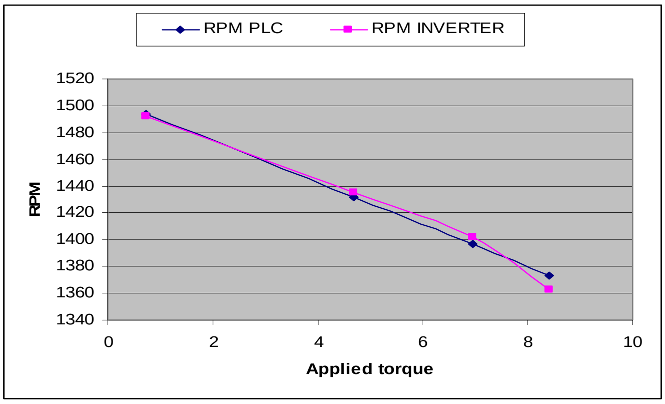

Fig. 7. Experimental speed torque characteristics with PLC and inverter

5. Conclusion

Successful experimental results were obtained from the previously described scheme indicating that the PLC can be used in automated systems with an induction motor. The monitoring control system of the induction motor driven by inverter and controlled by PLC proves its high accuracy in speed regulation at constant-speed-variable-load operation. The effectiveness of the PLC-based control software is satisfactory up to 96% of the synchronous speed. The obtained efficiency by using PLC control is increased as compared to the open-loop configuration of the induction motor fed by an inverter. Specifically, at high speeds and loads, the efficiency of PLC-controlled system is increased up to 10–12% as compared to the configuration of the induction motor supplied from a standard network. Despite the simplicity of the speed control method used, this system presents:

- constant speed for changes in load torque;
- full torque available over a wider speed range;
- very good accuracy in closed-loop speed control scheme;
- higher efficiency;
- overload protection.

Thus, the PLC proved to be a versatile and efficient control tool in industrial electric drives applications.

Acknowledgment
This research was supported by Marmara University Institute of Scientific Research Projects. Project Number: FEN-DKR-151105-0226

References

[1] S. S. Peng, M. C. Zhou, Ladder Diagram and Petri-Net-Based Discrete-Event Control Design Methods, *IEEE Transactions on Systems, Man, and Cybernetics- Part C, Applications and Reviews*, Vol.34 No.4 Nov. 2004 pp. 523-531

[2] J. G. GiIberl, G. R. Diehl, Application of Programmable Logic Controllers to Substation Control and Protection, *IEEE Transactions on Power Delivery*, Vol. 9, No. 1, January 1994, USA, pp. 384-388

[3]A. Hossain, M. H. Rashid, The Hardware and Softwar Interface of a Programmable Logic Grade Process Controller to an Industrial Control System, *IEEE Conferance*, 1990, USA, pp. 1862-1868

[4] T. Krairojananan, S. Suthapradit, A PLC Program Generator Incorporating Sequential Circuit Synthesis Techniques. *IEEE*, 399-402, USA

[5] K. Ji, Y. Dong, Y. Lee, J, Lyoul, Reliability Analysis Safety Programmable Logic Controller, *SICE-ICASE International Joint Conference* , Oct. 18-21, 2006 Korea

[6] A. R. Alae, M. M. Negm, M. Kassas, A PLC Based Power Factor Controller for a 3-Phase Induction Motor, *IEEE Transactions on Energy Conversion*, USA, 1065-1071,2000

[7] M. G. Ioannides, Design and Implementation of PLC-Based Monitoring Control System for Induction Motor, *IEEE Transactions on Energy Conversion*, 19, No:3 USA, 2004

[8] S. Gulpanich, Implementation Programmable Logic Controller for THAILAND Industries, *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce IEEE* (CIMCA-IAWTIC'05) USA, 2005

[9] M. Chmiel, E. Hrynkiewicz, The Way of Ladder Diagram Analysis for Small Compact Programmable Controller, *Information System and Technologies IEEE (KORUS' 2002)* page(s): 169-173, USA 2002

[10] A. M. Graham, M. Etezadi-Amoli, Design, Implementation, and Simulation of a PLC Based Speed Controller Using Fuzzy Logic, *Power Engineering Society Summer Meeting, 2000, IEEE,* Volume 4, 16-20 July 2000 Page(s): 2475 – 2480

[11] H. A. Barker, J. Song, A graphical simulation tool for programmable logic controllers, *Discrete Event Dynamic Systems - A New Generation of Modelling, Simulation and Control Applications,*
IEE Colloquium on 9 Jun 1992 Page(s): 4/1 - 4/4

[12] D.S. Evans, , Condition monitoring of sequential fluid power systems using programmable logic controllers**,** *Application of Advanced PLC (Programmable Logic Controller) Systems with Specific Experiences from Water Treatment,* IEE Colloquium on (Digest No.1995/112) 29 Jun 1995 Page(s):4/1 - 4/4

[13] E. Bundesmann, R. Thompson, Interfacing programmable logic controllers to electrical distribution monitoring systems, *Electrical Engineering Problems in the Rubber and Plastics Industries, 1993.*, IEEE Conference Record of 1993 Forth-Fifth Annual Conference of 993 Page(s):46 - 55

[14] L. Wang, K. Liu, Improvement of power factor and voltage for renewable energy systems using PLC's new fuzzy module, *Circuits and Systems, 2004. Proceedings.* The 2004 IEEE Asia-Pacific Conference on Volume 2, 6-9 Dec. 2004 Page(s):957 - 960 vol.2

[15] K. Anti, J. Markku, A. Jero, N. Markku, Real-Time Induction Motor Speed Control with a Feedback Utilizing Power Line Communications and a Motor Feeder Cable in Data Transmission *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on* Nov. 2006 Page(s):638 – 643

[16] M. Cade, Improvement of induction machine stability by modulation techniques, *Electric Power Applications, IEE Proceedings*, Volume 141, Issue 6, Nov. 1994 Page(s): 347 – 352

[17] M. B. Younis, G. Frey, Formalization of PLC programs to sustain reliability, *Robotics, Automation and Mechatronics, 2004 IEEE Conference* on Volume 2, 1-3 Dec. 2004 Page(s): 613 - 618

[18] C.M. Davidson, J. McWhinne, Engineering the control software development process *Factory 2000 - The Technology Exploitation Process, Fifth International Conference* on (Conf. Publ. No. 435) 2-4 April 1997 Page(s):247 – 250

[19] D. Hansen, Review of EMC Main Aspects in Fast PLC Including Some History, *Electromagnetic Compatibility, 2003. EMC '03. 2003 IEEE International Symposium*, 11-16 May 2003 Page(s):184 - 192 Vol.1

[20] M. Ogawa and Y. Henmi, Recent Developments on PC+PLC based Control for Systems, Brewery Process Automation Applications, *Korea Beer, SICE-ICASE International Joint Conference 2006* Oct. 18-21, 2006 in Bexco, Busan,