Research Paper
Impact Factor - **2.262**
Peer Reviewed Journal

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

SPEED CONTROL OF INDUCTION MOTORS USING PLC AND SCADA

Rani S
Electrical and Electronics dept., MEA College Engineering,Vengoor ,Malappuram, India.

Abstract
Continuous monitoring and control of motors and drives have become a need of process industries, which require operation of various drives in a pre-designed sequence. The programmable logic controllers (PLCs), an intelligent device, can realize automated operation. This paper presents a design and implementation of a monitoring and control system for the three phase induction motor based on PLC technology together with SCADA programming. 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 efficiency of PLC control is increased at high speeds up to 95% of the synchronous speed.

Index Terms—Computer-controlled system, computerized monitoring, electric drives, induction motors, motion control programmable logic controllers (PLCs), variable-frequency drives, SCADA.

1. INTRODUCTION

With the rapidly changes in industries and information technologies in recent years, some traditional bulk electronic appliances have to be monitored for a long time. The control of all equipments has been performed through the use of computers. Most equipments uses PLC to connect with computers to monitor each load and energy consuming devices.

Since technology for motion control of electric drives became available, the use of programmable logic controllers (PLCs) with power electronics in electric machines applications has been introduced in the manufacturing automation [1], [2]. 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 PLCs in automation processes to diminish production cost and to increase quality and reliability [4]–[9]. Other applications include machine tools with improved precision computerized numerical control (CNC) due to the use of PLCs [10]. To obtain accurate industrial electric drive systems, it is necessary to use PLCs interfaced with power converters, personal computers, and other electric equipment [11]–[13]. Nevertheless, this makes the equipment more sophisticated, complex, and expensive [14], [15].

The PLC correlates the operational parameters to the speed requested by the user and monitors the system during normal operation and under trip conditions. Inverter and controlled by PLC prove a higher accuracy in speed regulation as compared to a conventional V/f control system. Tests of the induction motor system driven by system. The efficiency of PLC control is increased at high speeds up to 95% of the synchronous speed. Thus, PLC proves them as a very versatile and effective tool in industrial control of electric drives.

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].

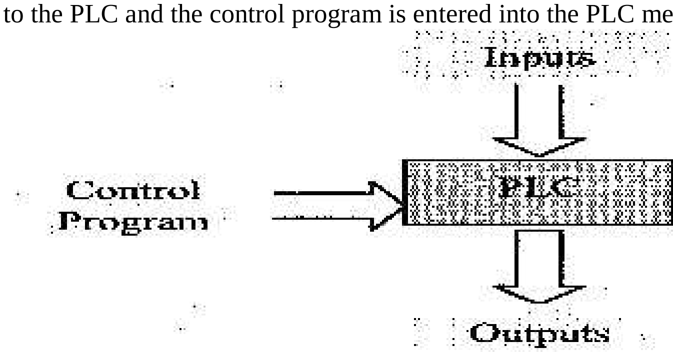

Fig 1.Control Action of a PLC

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

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.

Table I Induction Motor Technical Specifications

Connection type	Δ/Y
Input voltage	380/660 V ac
Input current	1,5/0,9 A
Rated power	0,6 kW
Input frequency	50 Hz
Pole number	4
Rated speed	1400 rpm

Table II,Inverter Technical Specifications

Output voltage	380, 460 V ac
Output frequency	0, 480 Hz
Output current	2,5 A
Output overload	150% for 60s
Power supply voltage	380, 460−10% V ac
Input current	3 A
Dissipated power	46 W

3.SOFTWARE DESCRIPTION

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.

Table III

	Available	Used
Discrete Inputs (%I)	32	8
Discrete Outputs (%Q)	16	9
Analogue Inputs (%AI)	8	7
Analogue Outputs (%AQ)	8	6
Register Memory (%M)	540	

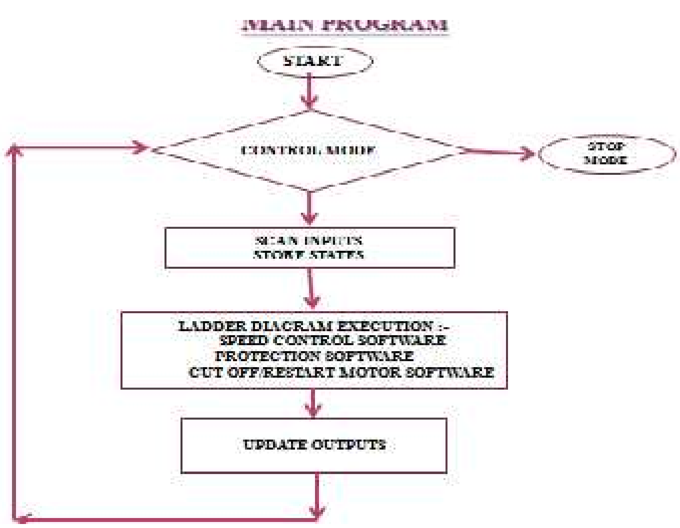

Fig. 2. Flowchart of the main program.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

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, [33], [34]. 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. 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 inactive control of the machine and monitors any data to check for correct operation

A.PLC Speed Control Software

In Fig. 3, 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 cannot keep the speed constant without the control loop with feedback and PLC. From the control panel, the operator selects the speed set point and the forward/backward direction of rotation.

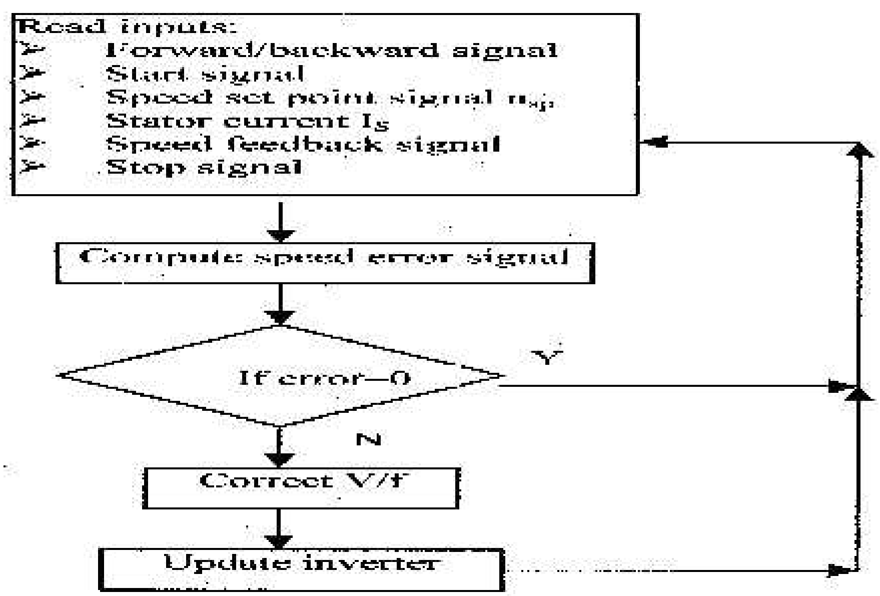

Fig 3. Flowchart of speed control software.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Then by pushing the manual start pushbutton, the motor begins the rotation. If the stop button is pushed, then the rotation stops. The corresponding input signals are interfaced to the DIM and the output signals to the DOM as shown in Table IV. The AIM receives the trip signal from the stator current sensor, the speed feedback signal from the tachogenerator, 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, 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 variations. At the beginning, the operator selects the gain 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).

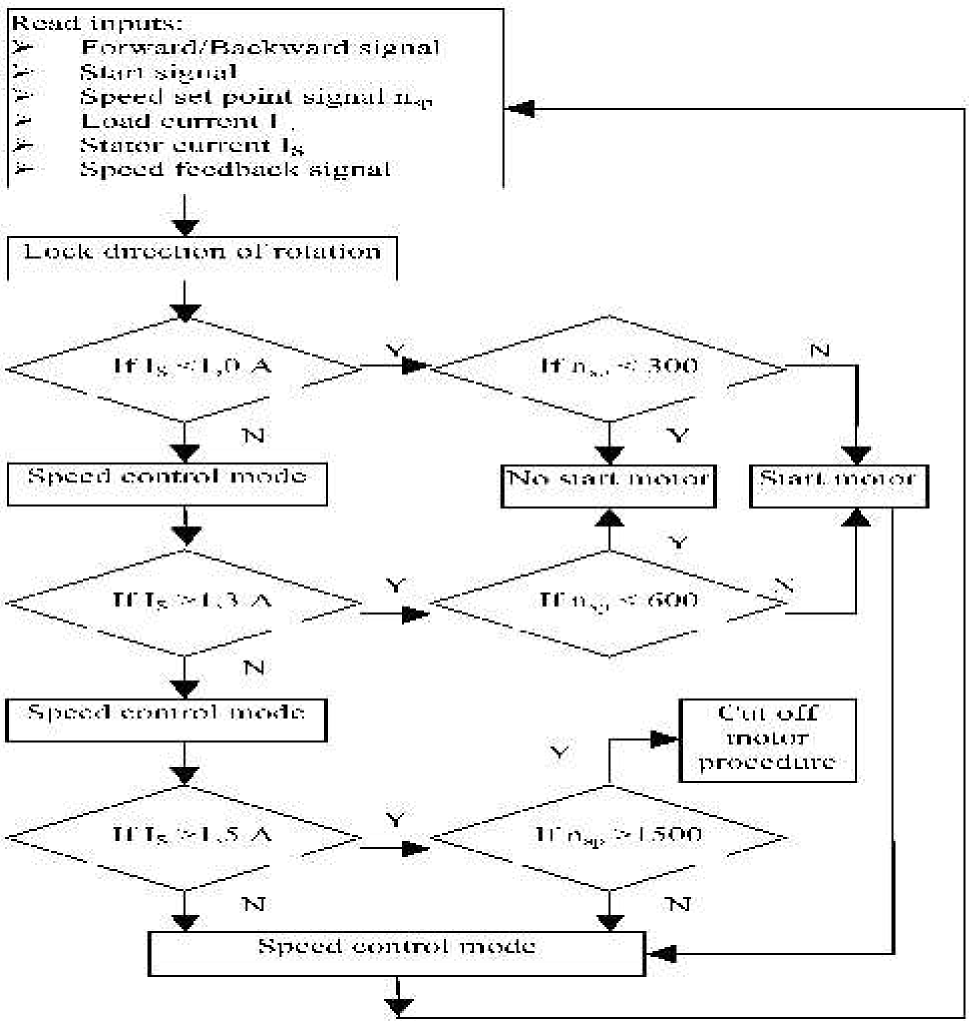

Fig 4. Flowchart of monitor and protection software.

The requested speed 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. A third display shows the load torque computed from the load current signal in Newton-meters (N m). Their corresponding signals are output to the AOM (Table IV).

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Table IV

Motherboard				
Module 1	Module 2	Module 3	Module 4	Module 5
	Analog input module-AIM	Discrete input module-DIM	Analog output module-AOM	Discrete output module-DOM
1. CPU	1. Speed feedback signal (input)	1. Start pushbutton signal	1. Speed feedback signal (display)	1. Relay_1
2. Power supply	2. Load current feedback signal	2. Stop pushbutton signal	2. Speed set point signal (display)	2. Start Lamp (run)
	3. Stator current signal (input)	3. Trip pushbutton signal	3. Load torque signal (display)	3. Relay_2
	4. Speed set point signal	4. Forward switch signal	4. Inverter frequency reference	4. Stop Lamp
	5. Controller gain signal	5. Backward switch signal	5. Load relay	5. Relay_3
	6. Controller time constant	6. Trip pushbutton signal	6. Stator relay	6. Trip Lamp
	7. Inverter analog port	7. 24 V dc		7. Inverter digital port
		8. 0 V dc		8. 24 V dc
				9. 0 V dc

B. Monitor and Protection Software

In Fig. 4, 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, the load current , the stator current , and the speed feedback signal are input to AIM.
III. At no load, if the speed set point is lower than 20% or r/min, the motor will not start.
IV. At an increased load over 0, 4 N m (40% of rated
V. torque), , and a speed set point lower than 40% or r/min, the motor will not start.
VI. If the load is increased more than 1, 0N m(rated torque) and if the speed set point exceeds 100% or r/min, the motor enters the cutoff procedure.
VII. In all other situations, the motor enters in the speed control mode and the speed control software is executed as described in Subsection A.

C. Cutoff and Restart Motor 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 1.5 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.

The ladder diagram for controlling the entire system is shown in fig. 5. The inputs and outputs are represented by NO/NC contacts and coils respectively

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

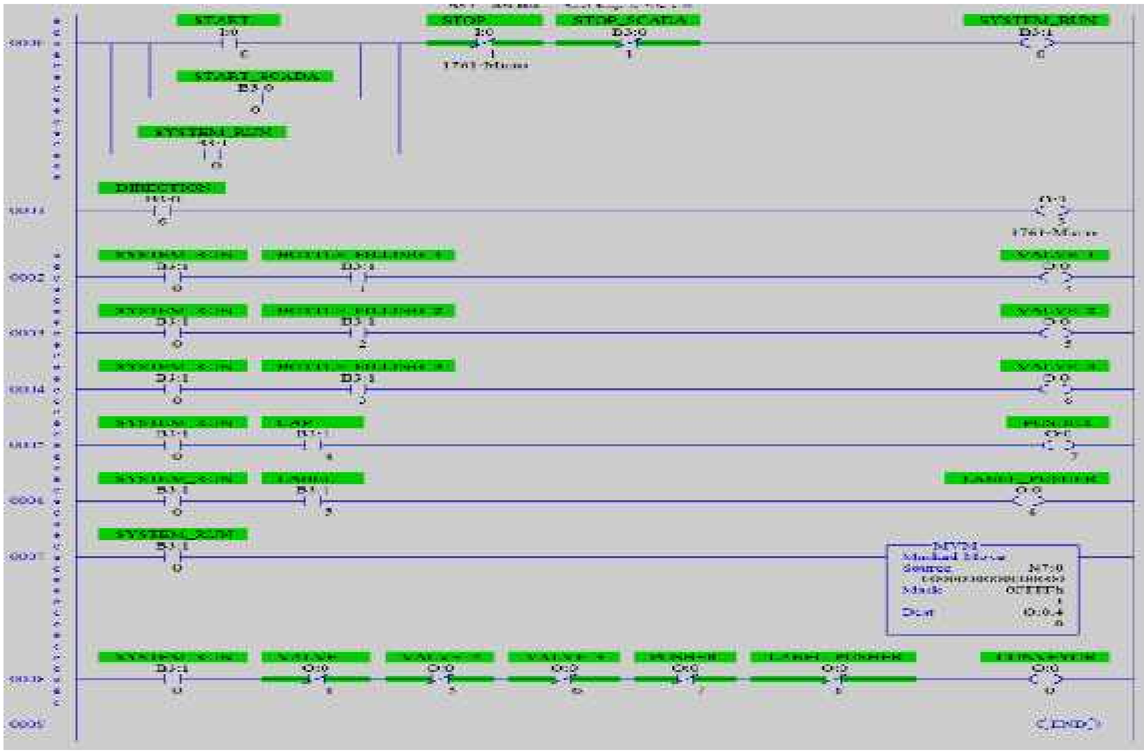

Fig 5 Ladder diagram.

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:

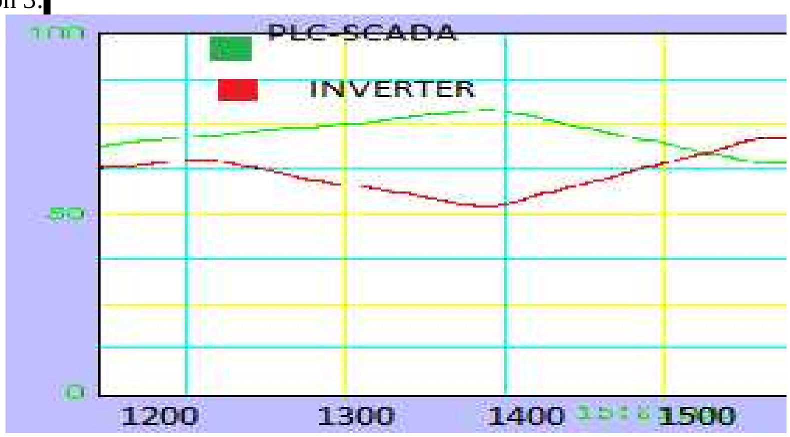

Fig 6. Comparison of result

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. 6.The results show that configuration b) operates with varying speed varying load torque characteristics for different speed set points *nsp*. 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 *nsp* ≥ 1400 r/min both con-figurations.

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.

REFERENCES

1. Mini Sreejeth, Pannod Kumar and Madhusudan Singh, "Development of Supervisory Control for Distributed Drives System", 978-1-4244-7781-4110/$.26.00 ©2010 IEEE.
2. Maria G. Ioannides, "Design and Implementation of PLC-Based Monitoring Control System for Induction Motor", 0885-8969/04$20.00 © 2004 IEEE.
3. Yaser Birbir, H Selcuk Nogay, "Design and Implementation of PLC-Based Monitoring Control System for Three-Phase Induction Motors Fed by PWM Inverter" International Journal of Systems Application, issue 3, vol.2,2008.
4. Frede Blaabjerg, Florin Lungeanu, Kennath Skaug, and Andreas Aupke 'Comparison of variable speed drives for single phase induction motors", 0-7803-7 1569/02/$10.000 2002 IEEE.
5. K. T . Erickson, "Programmable logic controllers," *IEEE Potentials*, vol.15, pp. 14–17, Feb./Mar. 1996.
6. A. Mader and H. Wuper, "Timed automation models for simple programmable logic controllers," in *Proc. 11th Euromicro Conf. Real-Time Systems*, 1999, pp. 106–113.
7. J. Marcos, E. Mandado, and C. M. Penalver, "Implementation offail-safe control systems using programmable logic controllers," in*Proc. IEEE/IAS Int. Conf. Industrial Automation and Control*, 1995, pp. 395–400.
8. Z. Futao, D. Wei, X. Yiheng, and H. Zhiren, "Programmable logic controller applied in steam generators water levels," in *Proc. IEEE/IAS 31stAnnu. Meeting Conf. Rec.*, vol. 3, 1996, pp. 1551–1556.
9. K. Dong-Il, S. Jin-Il, and K. Sungkwun, "Dependence of machining accuracy on acceleration/deceleration and interpolation methods in CNC machine tools," in *Proc. Conf. Rec. IEEE Industry Applications Soc. Annu. Meeting*, vol. 3, 1994, pp. 1898–1905.
10. D. P. Eng, "Diesel generation control system modernization," in *Proc. IEEE Can. Elect. Comput. Eng. Conf. Rec.*, vol. 1, 1998, pp. 125–128.
11. J. J. Harris, J. D. Broesch, and R. M. Coon, "A combined PLC and CPU approach to multiprocessor control," in *Proc. 16th IEEE/NPSS Symp. Fusion Engineering*, vol. 2, 1995, pp. 874–877.
12. T. Krairojananan and S. Suthapradit, "A PLC program generator incorporating sequential circuit synthesis techniques," in *Proc. IEEE Asia-Pacific*
13. *Conf. Circuit and Systems*, 1998, pp. 399–402.
14. M. Fabian and A. Hellgren, "PLC-based implementation of supervisory control for discrete event systems," in *Proc. 37th IEEE Conf. Decisionand Control*, vol. 3, 1998, pp. 3305–3310.
15. P. Marino, F. Poza, and J. B. Noguira, "Industrial LAN's with real-time communication servers," in *Proc. IEEE Int. Symp. Industrial Electronics*, vol. 1, 1997, pp. 23–28.