Conference Paper

Interfacing PMDC Motor to Data Port of Personal Computer

Laxmikant Ramakrishna, Abdulfattah Mohamed Ali, and Hani Baniodeh

Department of Electrical Engineering, Sirte University, P.O. Box 674, Sirte, Libya

Correspondence should be addressed to Laxmikant Ramakrishna; lrcsirte@gmail.com

Received 27 February 2013; Accepted 12 May 2013

Academic Editors: M. Elmusrati, A. Gaouda, and H. Koivo

This Conference Paper is based on a presentation given by Laxmikant Ramakrishna at “International Conference on Electrical and Computer Engineering” held from 26 March 2013 to 28 March 2013 in Benghazi, Libya.

Copyright © 2013 Laxmikant Ramakrishna et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Procedures and techniques of hardware interfacing to personal computer system through parallel data port to control permanent magnet DC (PMDC) motor and create LabVIEW integrated-development-environments (IDEs) based Virtual Instrument (VI) software are discussed. To test the designed VI software diagram, authors constructed interface hardware without taking support of any commercially available DAQ boards. Hardware resource utilization and performance optimization by creating VI are discussed. Testing the design (Hardware and VI) by varying the set point speed of the motor is concluded. It is observed that the motor speed gradually approaches and locks to the desired or set speed.

1. Introduction

Computer systems are part of modern control engineering. They can be classified into many types; in general they are classified as

- (1) fully dedicated system,
- (2) partially dedicated system,
- (3) nondedicated system.

The concept is to use the nondedicated personal computer (PC) system. PC's use is not only limited to surfing, learning, teaching, documenting, entertainment, social gathering or communication, and so forth; beyond that, it can also be used for many other known-unknown or imagined-unimagined applications, such as interfacing and controlling the physical world parameters, without investing for expensive devices and dedicated interfacing hardware like data acquisition cards (DAQ) or signal conditioning chassis/extension boards, and so forth. Spending more money, buying the task-oriented, dedicated, costly equipment, and getting the work done are laborious. Smart move would be to get the work done by simply existing, used-unused resources with a little or no programming skills and modifications. Here, this is purely a real-time hardware implementation demonstration.

The PC is used as a controller in the complete system in fact forming the open-loop control system [1–3].

1.1. Earlier Techniques. Different researchers [4–7] designed, fabricated/simulated, and studied DC motor speed controllers with different techniques and algorithms. They proposed many methods, by using PC/microcontroller with decoding circuit, and so forth. In general, if any body wants to measure and/or control the hardware with the PC and any programming language or IDE such as LabVIEW, it is essential to have DAQ Board/SCXI/PXI/PCI boards to access the data from outside world or from the PC to outside world [8]. Some of the earlier techniques are shown in Figures 1 and 2 which use below-listed devices [1, 2]:

- (i) PC with LabVIEW IDE,
- (ii) PCI plugged in DAQ board,
- (iii) SCXI-housing chassis,
- (iv) analog I/O modules and mount terminal boxes,
- (v) driver circuit,
- (vi) decoding logic circuit, and so forth (for simple stepper motor angular position control system).

FIGURE 1: Complete schematic of PC-based stepper motor control through LabVIEW software.

FIGURE 2: Components of an SCXI system; DAQ board assembly is used in order to access the analog signal or digital data from external world to PC and vice versa.

Schematic of interfacing stepper motor to PC is shown in Figure 1. The motor is interfaced in the open-loop control system configuration. The motor rotates in the forward direction, that is, in clockwise (CW) when +5 V is applied from the system to the motor and the motor starts rotating in backward direction, that is, counterclockwise (CCW) when zero volt is applied from the system [9]. In order to perform this task a simple I/O VI is developed using LabVIEW Signal Conditioning Extension Instrumentation (SCXI), which is a very expensive and nonefficient approach.

1.2. Motivation for the Work. Motor control can be done by many ways. The conventional way of controlling DC motor is quite expensive, complex and needs more hardware and special expertise.

Here, we are more concerned about “how to interface motor to the PC” without any DAQ board.

PC parallel port has data port, control port, and status port. The data port of the parallel port is used for simple I/O applications. It is an easy and cost-effective approach when compared to multiple numbers of I/O hardware panels.

2. Hardware Details of the System

2.1. Block Diagram of the System. Figure 3 shows the block diagram of the system setup. It consists of PC with LabVIEW

software, buffer, digital-to-analog converter (DAC), current-to-voltage (I/V) converter, voltage follower (VF), the current amplifier (CA), and PC’s parallel port (PP).

Data byte is buffered, the buffered data is converted into analog voltage and current through DAC, I/V converter, VF, and finally, the CA is used to drive the motor forming simple open-loop control system.

2.2. Working of the System. When a command from VI in PC is given, the control signal will flow into 4 bits of DAC-1408 through 4-bit parallel ports (D0-D3, selection of D0-D7 will provide different voltage magnitudes) of data byte and buffer. The buffered output signals are visualized via light-emitting diodes (LEDs). The DAC will convert the 4-bit digital data into equivalent analog signal and the converted analog signal will be in the form of current. But the motor needs both voltage and the current. The current is converted into voltage by using current-to-voltage converter; the output of I/V converter is not enough to drive the motor, so it is fed to voltage follower. The output of voltage follower is given to Darlington current amplifier, the Darlington current amplifier drives the DC motor to rotate. The control data outflow from PC-VI; to PMDC-Motor, is maintained till the desired voltage magnitude is achieved and same voltage is maintained till next command for desired speed. This forms

FIGURE 3: Block diagram of proposed DC motor interfaced to PC with LabVIEW IDE.

FIGURE 4: Circuit schematic of the system.

the simple open-loop control system for controlling the speed of PMDC motor. The schematic diagram is shown in Figure 4 and photographs of the system setup are shown in Figures 5(a) and 5(b).

The DC motor unit is interfaced to the computer through the D25-pin parallel port connector cord on interface board. To drive interface circuits the analog voltages are applied through the power supply constructed.

3. Experimental Implementation

Simple ON-OFF control technique is implemented. ON/OFF voltage push button switches in VI are used to regulate the motor speed. If, more speed is required then, more rated

voltage LEDs are switched ON, if less speed is required then, rated voltage LEDs are switched OFF respectively. The corresponding rated voltage is added to the existing voltage across motor or the voltage is reduced from the motor changing in the speed of the PMDC motor.

4. Virtual Instrumentation Software Details

The software is used to apply and change the voltage to the DC motor with respect to the real-time needs of the user and rotate the DC motor for required speed. In turn this is done by the VI. It imitates the appearance and operation of any other designed physical instrument. VI is defined as a process of combining hardware and software

FIGURE 5: (a) Photograph of the experimental setup, (b) photograph of the experimental setup.

FIGURE 6: Front panel diagram of user interface VI diagram.

with industry-standard computer technology to create a user-defined instrumentation solution, because their appearance and operation imitate physical instruments, such as switch, LED, oscilloscope, and multimeter, and so forth.

5. LabVIEW

LabVIEW GUI uses terminology, icons, and ideas familiar to technicians, scientists, and engineers. They rely on GUI rather than Character User Interface (CUI) language to describe programming actions. LabVIEW programs are called virtual instruments (VI). Each VI consists of two main parts:

- (a) front panel or front end,
- (b) block diagram [4, 8].

5.1. Design of Voltage Controller Front Panel. The front panel VI reads the voltage (to be applied to the motor via the front-end hardware user interface) entered by the user and sets the voltage sequences in data nibble through PCs parallel port and controls the motor voltage as per the front-panel VI command menu. The control function forms the programming part as per the user requirement and the hardware

FIGURE 7: Block diagram or source code of VI diagram for ON state and applying different control voltages to the out port.

will be the same for any type of control function. Figure 6 shows the front panel diagram which contains ON/OFF switch, motor-voltage-level-controls menu, time delay for data output, magnitude (LED) indicators, and so forth.

5.2. Block Diagram (Source Code) Designing. The block diagram or the source code involves *for-loop*, *while-loop*, sequence-nested structures, Boolean control bit which represents binary digit, array builder, binary to 32-bit digit converter and other icons, which are self-explanatory. The function diagrams or the source code diagrams are in graphics when case selector switch is ON and OFF which are shown in Figures 7 and 8, respectively.

The mathematical expression for PMDC motor speed controller used is given as following [9]:

$$\begin{aligned} \text{output voltage} = & \left(\frac{V_{ref}}{5.12k} \right) * \left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} \right. \\ & \left. + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{256} \right), \end{aligned} \quad (1)$$

where A is “digit indication of LED.” It is the magnitude of applied voltage to the motor. If LED is ON, “rated-high” voltage magnitude is applied to the motor. If LED is OFF, “low” voltage magnitude is applied to the motor. A1 is the LSB

TABLE 1: Applied voltage levels.

SL No.	Data applied				Voltages				Equivalent voltage theoretically	Voltage practically
	D3	D2	D1	D0	2.7 V	1.5 V	0.9 V	0.6 V		
1	0	0	0	0	0	0	0	0	0	0.28 V
2	0	0	0	1	0	0	0	0.6	0.6 V	0.59 V
3	0	0	1	0	0	0	0.9	0	0.9 V	0.89 V
4	0	0	1	1	0	0	0.9	0.6	1.5 V	1.20 V
5	0	1	0	0	0	1.5	0	0	1.5 V	1.50 V
6	0	1	0	1	0	1.5	0	0.6	2.1 V	1.8 V
7	0	1	1	0	0	1.5	0.9	0	2.4 V	2.11 V
8	0	1	1	1	0	1.5	0.9	0.6	3 V	2.42 V
9	1	0	0	0	2.7	0	0	0	2.7 V	2.47 V
10	1	0	0	1	2.7	0	0	0.6	3.3 V	3.02 V
11	1	0	1	0	2.7	0	0.9	0	3.6 V	3.33 V
12	1	0	1	1	2.7	0	0.9	0.6	4.2 V	3.63 V
13	1	1	0	0	2.7	1.5	0	0	4.2 V	4.02 V
14	1	1	0	1	2.7	1.5	0	0.6	4.8 V	4.25 V
15	1	1	1	0	2.7	1.5	0.9	0	5.1 V	4.55 V
16	1	1	1	1	2.7	1.5	0.9	0.6	5.7 V	4.86 V

FIGURE 8: Block diagram or source code of VI diagram for OFF-state and applying zero control voltage to the out port.

and A4 is MSB. The rated proportional voltage is applied as below [9]:

$$\text{real-time speed control equation} \\ = \left(\frac{V_{ref}}{5.12k} \right) * \left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} \right). \quad (2)$$

When Digit-1 to Digit-4 any digits are enabled the rated voltage magnitude levels (0.6 V, 0.9 V, 1.5 V, and 2.7 V) from the front panel menu, corresponding TRUE/FALSE bit is enabled in the block diagram. Since the TRUE or FALSE bit is single bit, it is connected to the Boolean array builder which is used to build 8-bit array. Then the 8-bit array is converted into 32-bit number and the 32-bit number is converted into 32-bit integer. The 32-bit integer is applied to the out-port-byte through which, finally, the command word exits out of the PCs parallel port to the motor. State of the motor depends upon the control word. The combination of the digits gives different equivalent voltages. Table 1 gives the magnitudes of the applied voltages to the motor. The result is change

in speed proportional to the applied voltage. In turn it is command word generated from the front panel user interface. The speed data (voltage V/S speed) shown in Table 1 can be stored in a file for further analysis or use. The VI provides on-line variation of voltages or set point, which facilitates the system to study for step, set-point, and random voltage variations. The behavior of motor on adding/subtracting different magnitudes can be monitored.

A complete VI developed using the icons, in block diagram for proper functioning of motor control is shown in Figure 6.

6. Experimental Observations

The experimental studies are carried out to verify the feasibility of the VI for different conditions. The VI is subjected to keep the motor continuously halted to take the path of complete FALSE cases. The performance indices of the controller (in terms of all zeros) are seen. The VI is subjected to keep the motor continuously with maximum speed rotating to take the path of complete TRUE cases. The performance indices of the controller (in terms of all ones) are seen.

The experiments are carried out to test the performance of the VI and circuit constructed. Selection of DAC data line selection makes lots of differences across the output.

6.1. Comparisons of Conventional DAQ-Based Motor Controllers and Our Interfaced Circuit. Table 2 gives the comparison for the conventional DC motor control system and our interfaced DC motor circuit. From the table it is also observed that when output sampling time is increased, the operating efficiency of the system changes as per the PC's processing speed. That is, if we consider that one conventional DAQ board setup and its housing arrangement are taking the processing time of 250 mSec, and our interfaced circuit is able

TABLE 2: Comparison.

SL No.	Parameter of Interest	Our system	Existing system	Outcome
1	Installation and housing space requirements	Compact system	More space is required	Our system is good because it is more compact than the other and it can be picked and installed as required by the user and no fixed hardware
2	Power requirement	Less power consumption	More power is required	Our system is good because it consumes less power with less hardware
3	Cost or price to build the system	Very less hardware, hence less price	More hardware, specified by the manufacturer	Our system is good because it operates with less hardware
4	Software	LabVIEW IDE	LabVIEW IDE and hardware drivers are required	Our system is good because just IDE is required to develop the VI

to change the state within 25 mSecs, so the difference is 1:10 and the percent of error occurring chances is more. Naturally our system is good. If such systems are exposed to operate for more time, then the percentage of error operation would be more. And the end product would be more erroneous.

7. Conclusions

The designed and developed system is unique and the first in its own kind, that is, without any DAQ or interfaces like PCI/SCXI/PXI/USB based boards. In order to control any physical parameter, investments in vendor software and hardware are a must. From small- to large-scale applications, data from external world to the PC can be transmitted. LabVIEW-based Parallel Port Data Acquisition System for laboratory/small experiments is non-expensive. The dedicated system can be designed and constructed using only the software. This would be a cost-effective solution and a great exercise for the new bees.

8. Future Scope

Definitely there is a scope for further development of this system. In future, application might require fewer changes in code or even someone may consider developing a totally new closed-loop data acquisition package using PID, fuzzy logic neural network, and AI. Since the O/P is different voltage level control, different PMDC motors may be used and their behavior can be studied, for irregular voltage change condition. Potential divider technique may be employed for direct speed variation implementation in VI instead of different magnitude voltage control.

References

- [1] R. Laxmikant, N. Katte, A. B. Kulkarni, P. Bhaskar, and C. S. Parvathi, "PC based position control system," in *Proceedings of the National Symposium on Instrumentation (NSI '05)*, CP-247, Cochin University of Science and Technology, Cochin, Kerala, November-December 2005.
- [2] R. Laxmikant, N. Katte, A. B. Kulkarni, P. Bhaskar, and C. S. Parvathi, "Study of the performance of PID controller for angular position control of a DC motor in presence of load and noise," *Journal of the Instrument Society of India*, vol. 37, no. 3, pp. 189–198, 2007.
- [3] L. Ramakrishna, *Design and development of fuzzy logic controllers and integrated fuzzy logic controllers and their applications [Ph.D. thesis]*, Gulbarga University, Gulbarga, India, 2009.
- [4] Y. H. Bharathi, B. R. Rekha, P. Bhaskar, C. S. Parvathi, and A. B. Kulkarni, "Multi-input fuzzy logic controller for brushless dc motor drives," *Defence Science Journal*, vol. 58, no. 1, pp. 147–158, 2008.
- [5] H. Wu, X. Chen, and L. Hu, "Embedded system of DC motor speed control based on ARM," in *Proceedings of the ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM '08)*, pp. 123–126, August 2008.
- [6] U. Maheswararao Ch., Y. S. Kishore Babu, and K. Amaresh, "Sliding mode speed control of a DC Motor," in *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT '11)*, pp. 387–391, June 2011.
- [7] S. Masuda, "A direct PID gains tuning method for DC motor control using an input-output data generated by disturbance response," in *Proceedings of the 20th IEEE International Conference on Control Applications (CCA '11)*, pp. 724–729, September 2011.
- [8] National Instruments, *LabVIEW User Manual*, National Instruments, Austin, Tex, USA, 2000.
- [9] C. T. Killian, *Modern Control Technology*, West Publishing Company, St. Paul, Minn, USA, 1996.

The Scientific World Journal

Hindawi Publishing Corporation
<http://www.hindawi.com>

Volume 2013

Hindawi

- ▶ Impact Factor **1.730**
- ▶ **28 Days** Fast Track Peer Review
- ▶ All Subject Areas of Science
- ▶ Submit at <http://www.tswj.com>