A PC CONTROLLED DC DRIVE USING PARALLEL PORT INTERFACING

Amr El Zawawi
Electrical Engineering Department, Faculty of Engineering,
University of Alexandria, Alexandria, Egypt.

ABSTRACT

A dc drive is controlled using an AT286, 12 MHz personal computer equipped by a math coprocessor and a standard data acquisition card - DT2801. Parallel port communication was adopted as it is fast and does not require a complicated I/O hardware structure. On the other hand, it implies more wiring and an involving software. The drive is controlled using a coordinated control structure with a master speed controller and a slave current controller. This structure is fully implemented by software using a PI controller algorithm. A digital firing circuit receives the output of the controller, in straight binary form, via the computer parallel port LPT1. Another parallel port, LPT2, transmits 8 control bits used for protection and interlocking purposes. The data acquisition card is used to input the speed and current feedback signals needed by the software controller. In the other direction, it outputs an interlock signal used to switch on and off the dc power supply to the whole electronic interfacing circuit. This measure proved to be essential for a fail safe operation of the system. The drive was built, commissioned and then tested to investigate its time response. An experimental study was conducted on the effect of the different parameters on the time required for one complete control loop.

Keywords: Drives, computer control, interfacing.

1- INTRODUCTION

Due to their excellent characteristics, dc motors have always been the natural choice for variable speed drives. Nevertheless, their extensive use was hampered by the unavailability of efficient, variable voltage dc supplies needed to control their speed. The development of high power, solid state electronic devices paved the way for the realization of efficient converters capable of providing a variable dc voltage from the widely available constant voltage, constant frequency ac supply. In this respect, it can be said that power electronics has lowered, to a great extent, the main hurdles that faced the wide use of dc motors as variable speed drives. The development of small inexpensive micro-computing systems enhanced the replacement of the rather rigid hardwired logic that characterized the analogue controllers by the flexible software control of their digital counterparts.

Many microprocessor based speed controllers for dc drives have been proposed. Some control strategies relying essentially on Bellman's dynamic programming approach were tested [2]. Other contributions adopted an adaptive structure allowing the switching between different controllers depending on the value of the actual speed of the dc motor relative to its base speed [5]. Model reference adaptive control theory was applied in a try to improve the controller performance [8]. The main stream of interest was focused on the improvement of the performance of the armature current control and/or limiting needed in any dc drive. Two different approaches were followed. The first approach kept the same coordinated control structure that was widely used in the analogue controllers: a

minor current control loop imbedded in a major speed control loop. In this structure, the armature current is continuously controlled and, if necessary, limited [6]. The transition from armature current control mode to armature current limiting mode is quite smooth [7,9,10].

The second approach tried to take advantage from the capabilities of microprocessors to develop a dual mode controller that normally operates in the speed control mode with a current limiter that intervenes only if the armature current exceeds a certain threshold [2,11]. DC drives based on this scheme are simpler to construct and have a quicker response but suffer from a main disadvantage; they allow large current overshoots. This limited their use to small power drives.

The proposed schemes were always based on single board microprocessor systems programmed using an assembly language [1-9]. Therefore, it was of utmost importance in these real time applications to avoid, as much as possible, complex arithmetic operations such as multiply or divide in favor of more simple and less time consuming operations like add, shift, compare, branch, etc. It is was also desirable to use simple integer rather than floating point arithmetic. Recent advances in the field of microelectronics made the use of personal computers to control electric drives both realistic and economically viable. In a previous paper [12], a brief account was given about the design and successful implementation of a PC controlled dc drive. It was proved that a personal computer, as simple as an IBM XT, 4 MHz, equipped with a serial port, was able to control a dc drive fed from a single phase fully controlled B2C converter bridge. An asynchronous serial communication protocol based on the RS 232 C standard was adopted. The system was required to deliver a control byte containing the required delay of the firing pulse to an external digital firing circuit which phase controls the thyristor converter once every half cycle (10 ms). However, this system was not able to process interlock and state monitoring signals within the specified time limit (10 ms).

In this paper, parallel port communication was adopted as it is considerably faster. This will give the PC enough time to process interlocks and state monitoring signals in an attempt to obtain a well protected and fail safe system with a comprehensive on line display of the important parameters pertaining to the system operation. The system can be used with a three phase, six pulse, B6C converter bridge if the complete control loop can be executed in less than one sixth of a cycle (3.3 ms).

2- THE SYSTEM DESCRIPTION

The system under study is that of the speed control of a separately excited dc motor whose armature is fed from a fully controlled, single phase converter B2C bridge. The block diagram shown in Figure (1) depicts a coordinated control scheme made from an inner current control loop lying within an outer speed control loop. This control scheme is fully implemented by software on an AT, 12 MHz personal computer equipped by a math coprocessor, two parallel ports and a DT2801 data acquisition card. The PC receives the current and speed feedback signals from the current feedback module and the tachogenerator respectively.

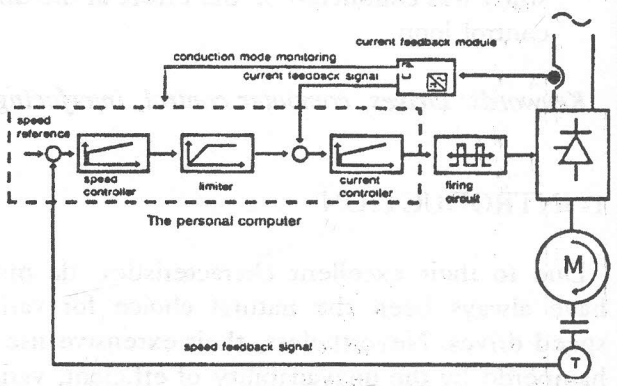

Figure 1. The pc controlled dc drive.

The current feedback module outputs also a conduction mode monitoring signal (DisMC) that detects the discontinuous current conduction mode. In fact, constrair converte Figure current this latt be used analogu

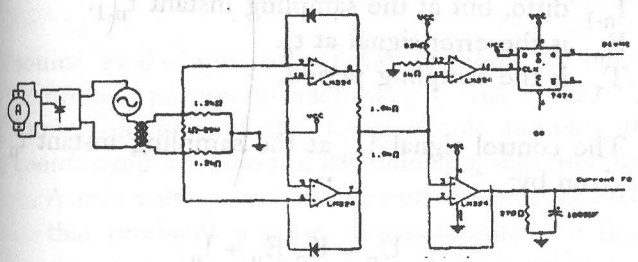

Figure 2-a. The current feedback circuit.

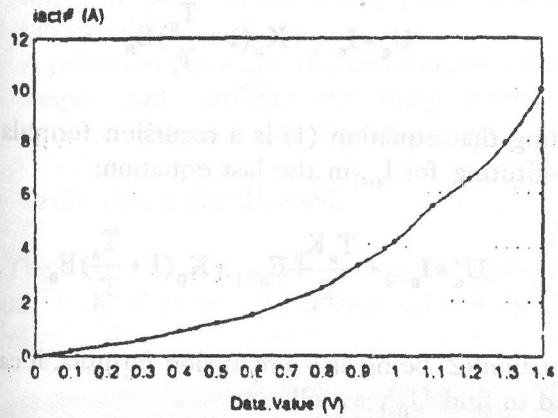

Figure 2-b. Transfer characteristics of the current feedback circuit.

A family of curves were plotted for the emitter voltage V_o as function of V_i, the voltage across the opto-diode in series with a resistance R_i. The best range was experimentally obtained with R_i = 1 k and R_o=2.2 k. The transfer characteristics of the opto-coupler transducer shown in Figure (3-b) indicates its failure to detect armature currents below 0.6 A, and hence cannot detect a transient passage by zero of the armature current. This means that this transducer is unable to detect the conduction mode of the converter. This is why this option was discarded in favor of the indirect method using the ac current input to the converter shown in Figure (2-a). The primary of a 20 A/ 5 A current transformer is connected in series with the ac supply of the converter. Its secondary is connected to a 1 Ω - 25 W resistor. The voltage across this resistor is rectified using a precision rectifier. The output of this latter is buffered by a voltage follower, then smoothed using a 1000 μF electrolytic capacitor. The current feedback module outputs two signals: a digital signal (DisMC) monitoring the converter current conduction mode and an analogue signal representing the current feedback. A look up table was used to obtain the true rms value of the armature current. This step was necessary not only because of the non linearity of the transfer characteristics of the current transducer, as revealed by Figure (2-b), but also because the current wave form is non sinusoidal and its form factor depends on the firing angle. On the other hand, a D type flip-flop is used to latch the current discontinuity signal (DisMC) which will be read by the computer whenever it completes a control cycle. This cycle includes the acquisition of the speed and current feedback signals, then using a digital PI control algorithm to compute the required firing pulse delay. The flip-flop is then reset by the digital signal C# from bit 4 of the parallel port LPT2. This step is necessary for the flip-flop to resume its current discontinuity detection task. Then speed and current feedback signals are processed, under program control, by the data acquisition card DT 2801 for analogue to digital conversion.

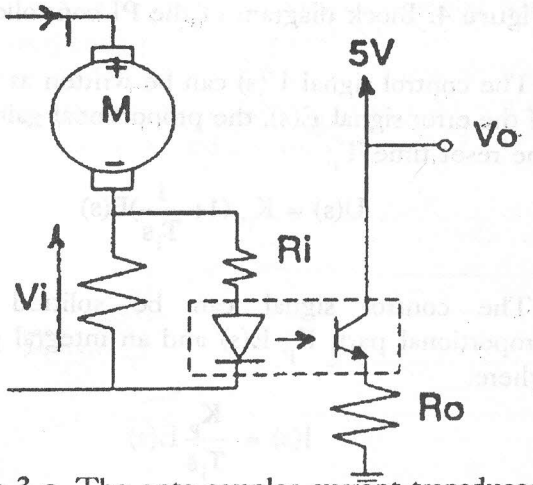

Figure 3-a. The opto-coupler current transducer.

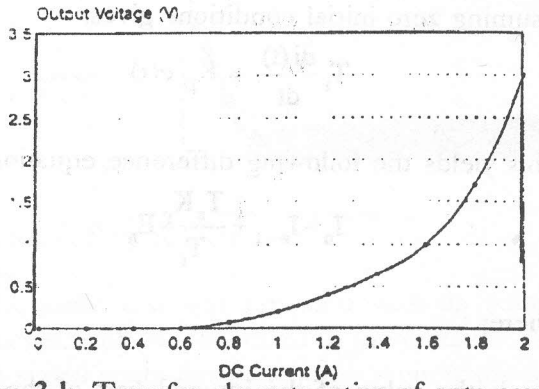

Figure 3-b. Transfer characteristics of opto-coupler current transducer.

3- THE DIGITAL CONTROL ALGORITHM

PI controllers have always been the natural choice for dc drives to ensure accurate steady state response. Figure (4) gives the block diagram of such a controller. In order to implement a computer controlled drive, we need first to derive an algorithm for a PI controller that can be used, with different parameters, for both speed and current controllers.

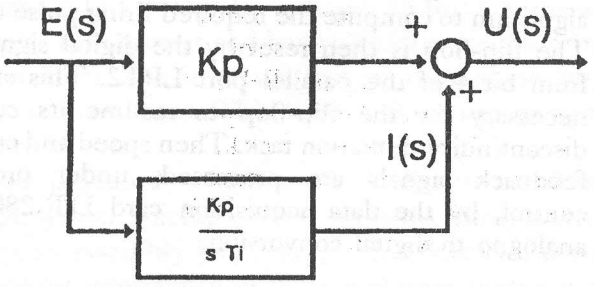

Figure 4. Block diagram of the PI controller.

The control signal U(s) can be written as function of the error signal E(s), the proportional gain K_p and the reset time T_i;

$$U(s) = K_p\,(1+\frac{1}{T_i s})E(s)$$

The control signal can be splitted into a proportional part, K_p E(s) and an integral part I(s), where;

$$I(s) = \frac{K_p}{T_i s}E(s)$$

Transforming the last equation to the time domain, assuming zero initial conditions gives;

$$T_i\frac{di(t)}{dt} = K_p\,e(t)$$

This yields the following difference equation;

$$I_n = I_{n-1} + \frac{T_s K_p}{T_i}E_n \qquad (1)$$

where;

I_n is the value of the integral part of the control signal at the sampling instant t_n.

I_{n-1} ditto, but at the sampling instant t_{n-1}.

E_n is the error signal at t_n.

T_s is the sampling period.

The control signal U_n at the sampling instant t_n is given by;

$$U_n = K_p\,E_n + I_n$$

Substituting for I_n from equation (1);

$$U_n = I_{n-1} + K_p(1+\frac{T_s}{T_i})E_n \qquad (2)$$

Noting that equation (1) is a recursion formula and substituting for I_{n-1} in the last equation;

$$U_n = I_{n-2} + \frac{T_s K_p}{T_i}E_{n-1} + K_p(1+\frac{T_s}{T_i})E_n \qquad (3)$$

Equation (2) being also a recursion formula, it can be used to find U_{n-1} as follows;

$$U_{n-1} = I_{n-2} + K_p(1+\frac{T_s}{T_i})E_{n-1} \qquad (4)$$

Making use of the last equation, equation (3) can be rewritten as;

$$U_n = U_{n-1} - K_p E_{n-1} + K_p(1+\frac{T_s}{T_i})E_n \qquad (5)$$

Recursion formula (5) will be used to implement the PI controller. Two remarks seem in order:

1- The control signal U_n at the sampling instant t_n is function of its previous value U_{n-1}, the error signal E_n and its previous value E_{n-1}.
2- The control signal depends on the sampling period T_s; a dependency that has its implications on the software as well as the hardware.

4- THE DIGITAL FIRING CIRCUIT

A phase control strategy was implemented using a digital firing circuit. This strategy uses the switching characteristics of the thyristors to connect the ac

source to the armature of the dc motor for only a controlled portion of each supply half cycle. The implementation of the firing circuit requires the coordinated operation of the following four modules:

- A zero voltage crossing detection circuit (ZVCD) that produces a pulse to synchronize the firing circuit to the supply voltage of the converter.
- A delay circuit based on two cascaded synchronous 4 bit counters that count the required delay of the firing pulse in a straight binary code.
- A pulse conditioning and distribution circuit that shapes and amplifies the firing pulses before delivering them to the thyristor gates.

5- THE TIME RESPONSE

The experimental setup shown in Figure (5) was built in the Electrical Machines Laboratory of Qatar Uni

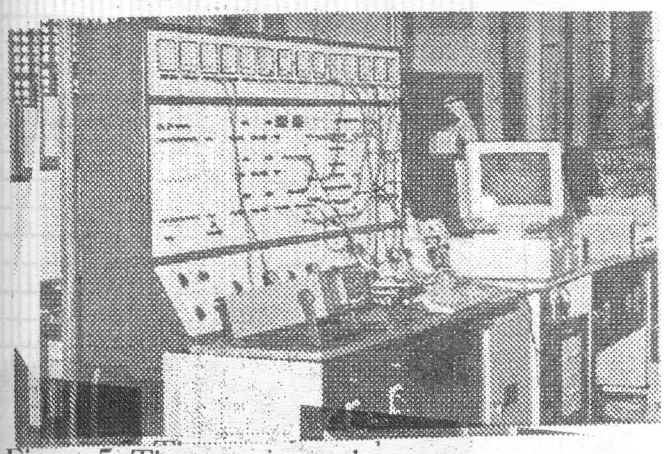

Figure 5. The experimental setup.

Figure (6) shows the back plane of the personal computer with connections made to LPT1, LPT2 and the data acquisition card inserted in an expansion slot.

The time response of the drive was investigated by applying a step input corresponding to a speed reference of 1000 rpm to a 1.5 kW, 200V, 1400 rpm, 8.7 A dc motor. The controller parameters were experimentally tuned to yield a response close to that characterizing an error modulus optimum [13]. The controller parameters were:

$$K_{ps}= 2,\ T_{is}= 50 \text{ ms},\ K_{pi} = 0.8,\ T_{ii}=80 \text{ ms}$$

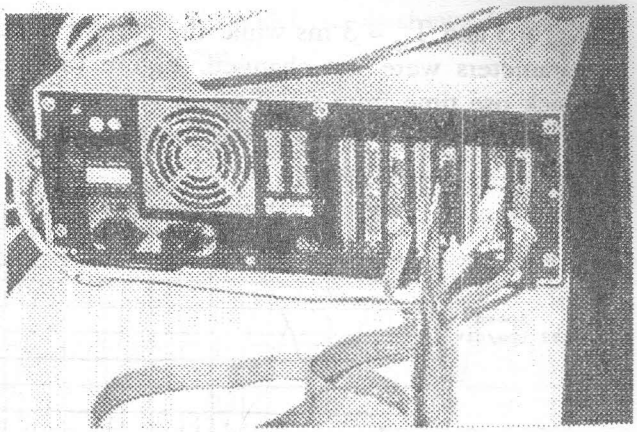

Figure 6. The back plane of the personal computer.

where K_{ps} and K_{pi} are the proportional gains of the speed and current controllers respectively while T_{is} and T_{ii} are their reset times. The current limiter was set at 10 A. The controller sampling period was 5.5 ms corresponding to case v of Table (1).

The response given by Figure (7) reveals a quick rise time from zero to 1000 rpm in 0.45 s, a peak overshoot of 18 % and a settling time of 1.8 s.

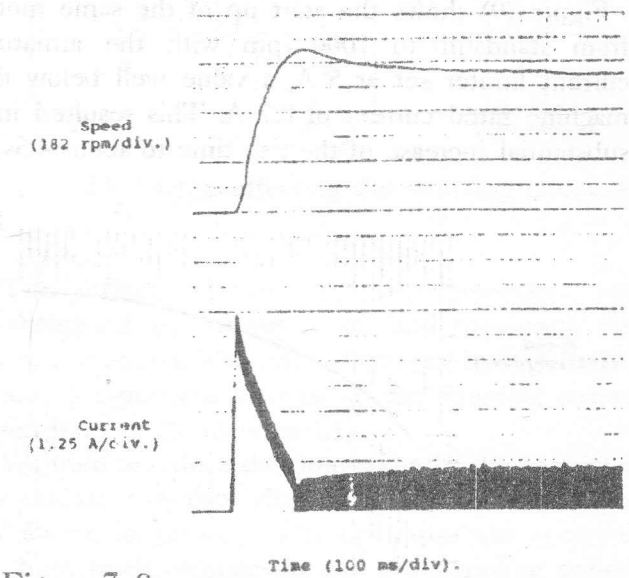

Figure 7. Step response.

The same test was repeated with an additional inductance of 20 mH added to the armature circuit. The speed controller parameters were readjusted to K_{ps} = 3 and T_{is} = 3 ms while the current controller parameters were not changed. Figure (8) shows a

K_{ps} = 3 and T_{is} = 3 ms while the current controller parameters were not changed. Figure (8) shows a quick rise time from standstill to 1000 rpm in 0.4 s, a peak overshoot of 14.5% and a settling time of 1.3 s.

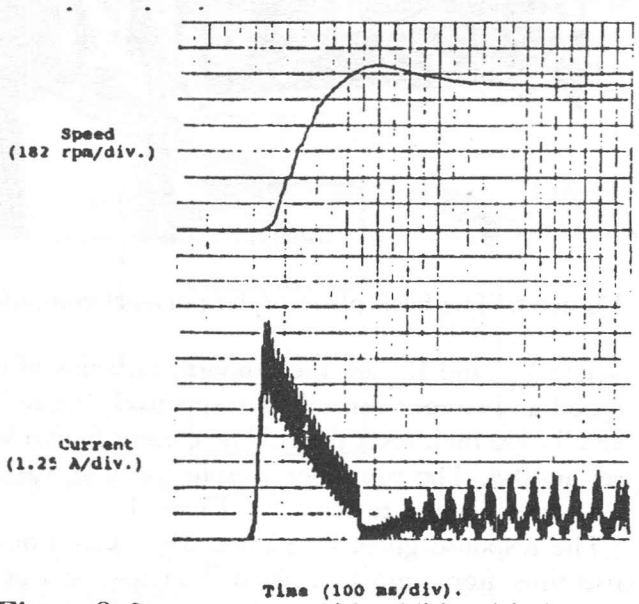

Figure 8. Step response with additional inductance.

Figure (9) shows the start up of the same motor from standstill to 1000 rpm with the armature current limiter set at 5 A, a value well below the machine rated current of 8.7 A. This resulted in a substantial increase of the rise time to about 4.5s.

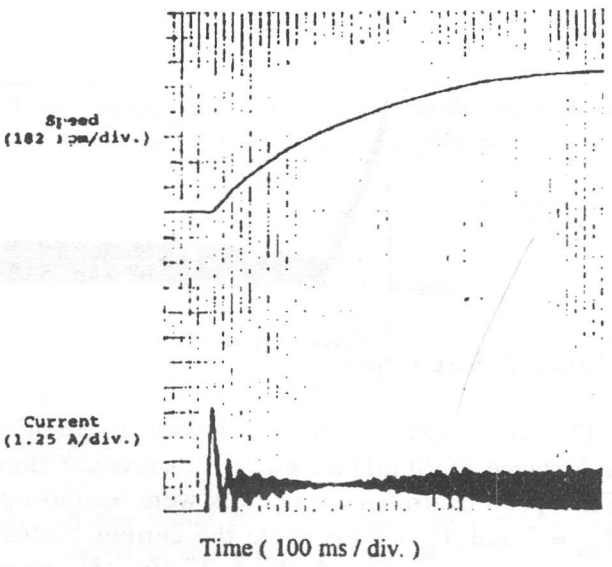

Figure 9. Start up with reduced current limit.

The controller parameters tuned at the error modulus optimum give an overshoot close to 4 % and a quick rise time but cause a rather slow leveling after a load disturbance [13]. In order to obtain a compromise between the drive starting characteristics (step response) and a quicker leveling after a load disturbance, the reset times used with the controllers were less than those corresponding to the modulus optimum. This explains the larger overshoots observed in the step response.

In order to investigate the immunity of the drive to load disturbances, it was subjected to a severe load torque step from 2.4 Nm to 6.5 Nm. The motor speed and torque were recorded. Figure (10) illustrates the ability of the drive to recover from the load disturbance within 0.7 s with a damped frequency of oscillation of 1.43 rad/s.

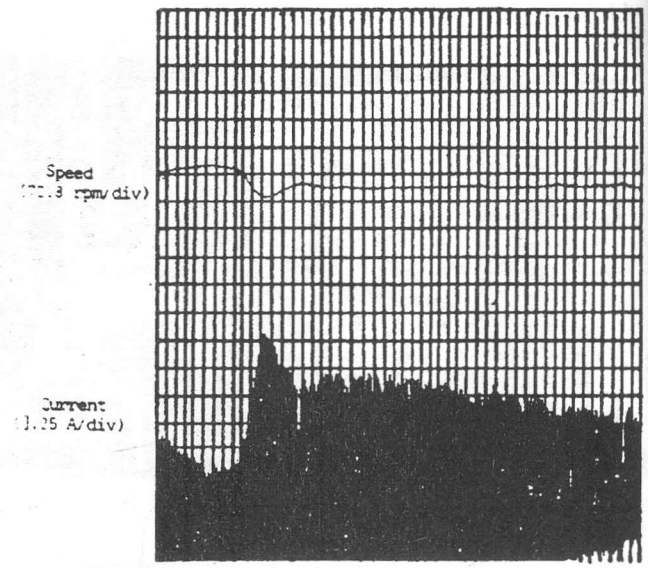

Figure 10. Load disturbance.

The same test was repeated with an additional inductance of 20 mH in the armature circuit. Figure (11) shows a speed leveling within 0.9 s with a damped frequency of oscillation of 1.11 rad/s.

6 - FACTORS AFFECTING THE SAMPLING FREQUENCY

The sampling period of the controller is determined by the time required to execute once

the software control loop. The control cycle must be completed in less than 10 ms. This requirement stems from the fact that the single phase converter thyristor bridge represents a sampling element that offers a chance to control the firing angle once every half cycle. In other words, if a 50 Hz ac supply is used, this means that any single phase full wave converter cannot be controlled more than once every 10 ms. This time was measured by software using the BASIC command TIMER, and by hardware by setting then resetting bit 6 of the parallel port LPT2 once each time the control loop was executed. Complete agreement was observed between the two methods. Figures (12-16) give the logic state of bit 6 of LPT2 as recorded for different operating conditions.

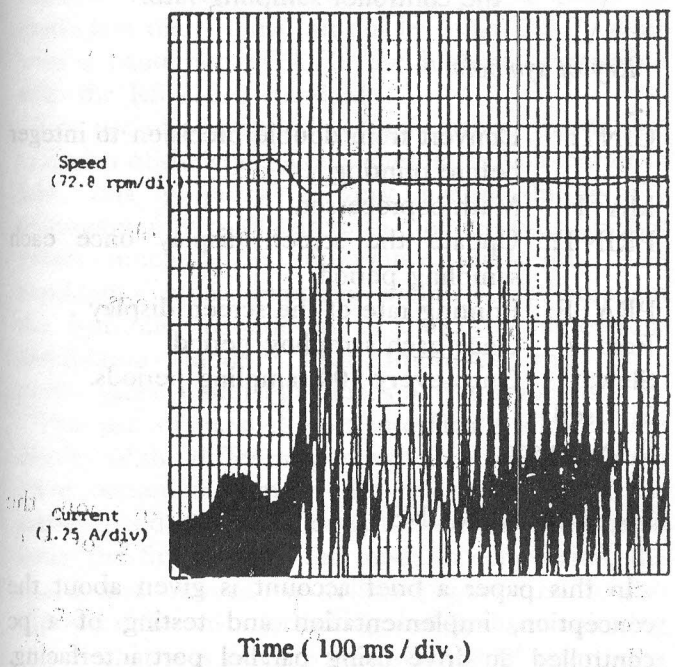

Figure 11. Load disturbance, with additional inductance in the armature circuit.

The personal computer used is an AT 286, 12 MHz equipped by a math coprocessor and a 1 MB RAM. The control cycle and hence the sampling period was found to be 9 ms as shown in Figure (12). In order to investigate the effect of the different parameters on the sampling period, the speed, current and firing angle data were converted from double precision to integer before printing them. This resulted in an insignificant reduction in the sampling period by 0.1 ms as shown in Figure (13) representing case II.

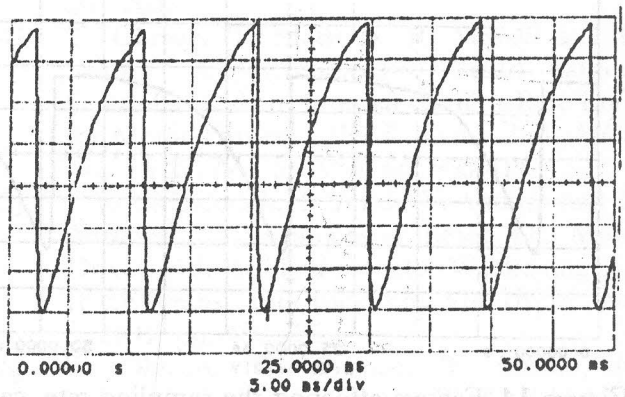

Figure 12. Factors affecting the sampling rate, case I.

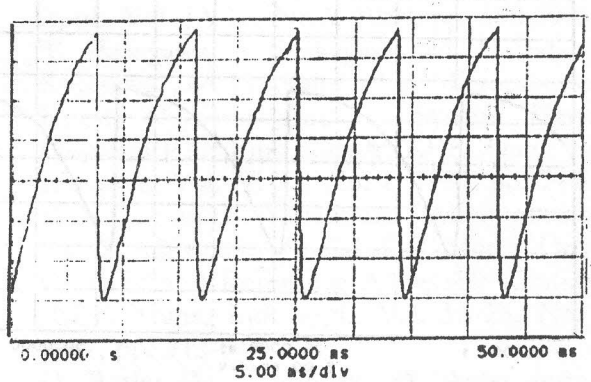

Figure 13. Factors affecting the sampling rate, case II.

The effect of the math coprocessor was investigated by turning it off and measuring the sampling period. The result displayed by Figure (14) shows a significant increase of the sampling period from 9 ms to 29 ms (case III).

We tried to reduce the time taken by the computer to update the data displayed on the screen, by inhibiting its printing. The computer was operated without math coprocessor and the sampling period dropped from 29 ms to 14 ms, as shown in Figure (15) representing case IV. The sampling period is still more than 10 ms and hence unacceptable.

In order to obtain a lower sampling period, the math coprocessor was turned on and a counter was created that updates the screen information once

every 100 complete control cycles. This resulted in a sampling period of 5.5 ms as shown in Figure (16) displaying case V.

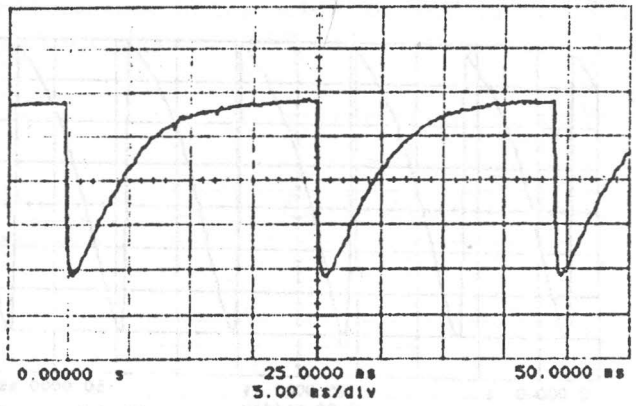

Figure 14. Factors affecting the sampling rate, case III.

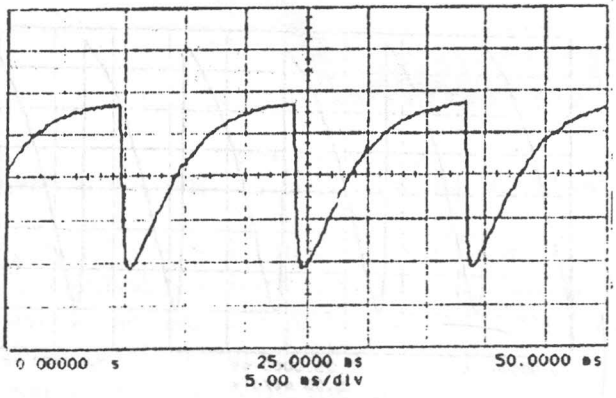

Figure 15. Factors affecting the sampling rate, case IV.

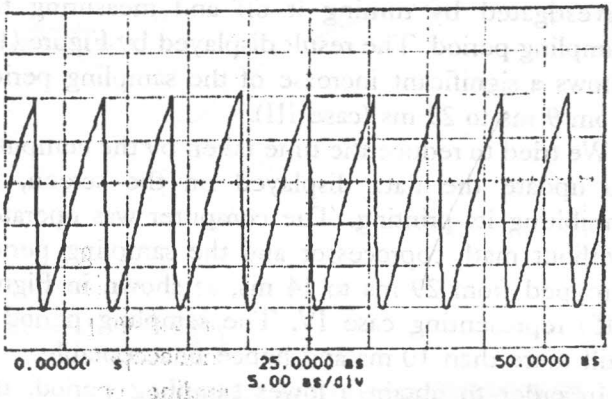

Figure 16. Factors affecting the sampling rate, case V.

Finally a compiled version of the software packag was tried and the sampling period was observed t drop to 3 ms; a value that allows the use of thi package to control efficiently a 6 pulse, 3 phas converter drive.

Table 1. gives the key for the different factors affecting the controller sampling rate.

case#	CINT	MATH	PRINT	PRATE
I	0	1	1	Fast
II	1	1	1	Fast
III	0	0	1	Fast
IV	0	0	0	
V	0	1	1	Slow

Table (1) - Key for the different factors affectin the controller sampling rate.

Key of symbols:

CINT: Convert from double precision to intege before printing.
MATH: Math coprocessor.
PRINT: Update the screen display once eac sampling period.
PRATE: Printing rate of the screen display .
Fast: Once each sampling period.
Slow: Once every 100 sampling periods.
1: Used.
0: Not used.

7 - CONCLUSIONS

In this paper a brief account is given about th conception, implementation and testing of a p controlled dc drive using parallel port interfacing The pc is a sampling element that may deteriorat the drive performance. However, the dc driv contains an inherent discrete element, the thyristo bridge, which operates with a sampling period tha can be calculated from the following relation;

$$T_c = (f. p)^{-1}$$

where;
T_c is the sampling period of the thyristor converte
p is the number of pulses of the converter.
f is the frequency of the converter ac supply.

This means that as long as the digital control loop period T_s is less than or equal to T_c, the system will not experience any performance deterioration if compared to its analogue counterpart. Experimental investigation of the factors affecting the controller sampling period T_s revealed that the use of a math coprocessor is a must and that updating the screen information is time consuming. It has been shown that refreshing the screen information once every 100 T_s , almost twice every second, gives an acceptable display quality and results in a controller sampling period T_s less than the converter sampling period T_c for a single phase 2 pulse converter bridge. It was found that the use of a compiled version of the software controller results in a controller sampling period of 3 ms, which can be used with a 3 phase, 6 pulse converter bridge. This was not possible with the serial port interfacing [12] because the controller sampling period could be made less than 10 ms but not less than 3.3 ms even with a baud rate of 9600, the highest permissible with the RS 232 C standard.

The use of a compiled version restricts the on-line variation of the controller parameters. As a matter of fact, this property is very useful during the commissioning of the drive to tune its controller but looses much of its importance afterwards. The conduction mode detection circuit is used to adapt the controller parameters to the armature current conduction mode. This improves the drive performance under light loads and low speeds.

The use of parallel port interfacing allowed the display of the different parameters pertaining to the drive operation namely; the reference speed, the actual speed, the actual armature current, the current limit, the firing angle, the mode of conduction, the speed and current controllers parameters. It also allowed different interlocking and health monitoring signals that was not possible with serial interfacing [12], making this system more reliable and fail safe.

REFERENCES

[1] T. Ohmae, T. Matsuda, K. Kamiyama and M. Tachikawa. "A Microprocessor Controlled High Accuracy Wide Range Speed Regulator for Motor Drives", IEEE Trans. Ind. Electron., vol. IE-29, No. 3, pp. 207-211, 1982.

[2] B. White, R. Lipczynski and A. Daniels: " A Simple Digital Control Scheme for a DC Motor "; IEE Proc., Vol. 130, Pt. B, No. 2, 1983, pp. 143-147.

[3] D. Divan, G. Hancock, G. Hope and T. Barton: " Micro programmable Sequential Controller "; IEE Proc., Vol. 131, 1984, pp. 201-208.

[4] T. Ohmae, T. Matsuda, R. Masaki and K. Saito: " A Microprocessor Based Current Controller with an Internal Current Rate Loop for Motor Drives "; IEEE Trans. Ind. Appl., Vol. IA-22, No. 5, 1986, pp. 805-811.

[5] J. Kettleborough, I. Smith, V. Vadher and F. Antunes: " Microprocessor Based DC Motor Drive with Spillover Field Weakening " ; IEEE Trans . Ind. Electron., Vol. IE-38 , No. 6, 1991, pp. 469-475.

[6] T. Ohmae, T. Matsuda, T. Suzuki, N. Azusawa, K. Kamiyama and T. Konishi : " A Microprocessor Controlled Fast Response Speed Regulator with Dual Mode Current Loop for DCM Drives "; IEEE Trans. Ind. Appl., Vol. IA-16, No.3, 1980, pp. 388-394.

[7] T. Hasegawa, T. Nakagawa, H. Hosoda, R. Kurasawa and H. Naito : " A Microprocessor Based Thyristor Leonard System Having Powerful RAS Functions "; IEEE Trans. Ind. Electron., Vol. IE-31, No. 1, 1984, pp. 74-78.

[8] H. Naitoh and S. Tadakuma: " Microprocessor Based Adjustable Speed DC Motor Drives Using Model Reference Adjustable Control"; IEEE Trans. Ind Appl., Vol. IA-23, No. 2, 1987, pp. 313-318.

[9] D. Butler, M. Churches , Y. Anbe and H. Naito: "Compensation of a Digitally Controlled Static Power Converter for the Damping of Rolling Mill Torsional Vibration " ; IEEE Trans. Ind. Appl. Vol. IA-28, No. 2,1992, pp. 427-433.

[10] P. C. Sen : " Thyristor DC Drives " ; Wiley, 1981, pp. 19-54.

[11] K. Zelenka and T. Barton: " A Fast Acting Current Limit for a DC Motor Drive" ; IEEE Trans. Ind. Appl., Vol. IA-22, No. 5, 1986, pp. 798-804.

[12] A. El Zawawi: " The Design and Implementation of a Micro-Computer Controlled DC Drive "; Proc. Intern. Conf. on Signals and Systems, Al Ain, Jan. 29-31, 1990, Vol. 2, pp. 129-136.

[13] F. Frohr and F. Ottenburger: " Introduction to Electronic Control Engineering "; Heyden & Son, 1982, pp. 256-278.