

A NOVEL ALGORITHM FOR USING A MICROCONTROLLER IN CONTROLLING ADJUSTABLE SPEED DRIVE

¹Ahmad Ehteshamul Islam, ²Syed Md. Jaffery Al-Kadry and ³A.N.M. Zainuddin

^{1,3}Lecturer, Department of Electrical & Electronic Engineering, BUET, Dhaka 1000, Bangladesh

²Associate Design Engineer, Power IC Ltd., Baridhara, Dhaka 1212, Bangladesh

¹Email: ehteshamulislam@eee.buet.ac.bd

ABSTRACT

The controlling of adjustable speed drive (ASD)s using microcontroller/ microprocessor is very popular now-a-days. Here an efficient algorithm is suggested and tested for operating a induction motor adjustable speed drive using a microcontroller (AT89C51). The scheme is designed to offer best performance, with minimal starting current and also ensuring different protections.

1. INTRODUCTION

Commercial and industrial firms use ASD systems for a variety of applications, including pumps, fans and blowers, hoists, cranes, machine tools, film lines, extruders, and textile-fiber spinning machines. Drive vendors have been able to respond to the unique demands and characteristics by producing a variety of drives. The variety of drives and a multitude of possible applications make the selection of the optimum drive a challenge.

Different research groups' work on ASDs operating with different types of ac motors. Different PWM techniques are available in the literature developed by the researchers for the control of voltage, frequency and spectral distribution of static inverters ([1], [3], [4]), used for driving the ASD. Such PWM techniques are applied in cases up to several thousand hp. Some manufacturers still build current-source-inverter and variable-voltage-inverter units in their higher hp sizes, but competition is starting to drive them to develop pulse-width-modulated designs to meet demand for improved efficiency and performance.

Three-phase induction motors fed ASD are popular in the industry for its reliable operation. A microprocessor (μP) or a microcontroller (μC)

along with other interface circuits normally controls the inverter of ASDs [2]. And an efficient algorithm is required here for driving that ASD using the processor so that-

- Proper voltage and frequency can be set so that the motor runs at set speed.
- The starting current of the ASD is minimized.
- The achieved speed and set speed are closely matched.
- The current taken by ASD at a particular speed is well within some limits, along with some over-current protection mechanism.

In this paper, we are presenting an algorithm, which was tested in driving a 500-Watt three-phase induction motor at wide range of speed (50 rpm to 5000 rpm) using a single-phase AC supply. The scheme fulfils all the above-mentioned properties without reasonable deterioration in performance, having full protection and safety measures.

2. THE CONTROL SCHEME

The scheme developed in the project was mainly targeted to improve the control system of the three-phase induction motor ASD. For the tight control of the speed, several conventional methods were adopted as appropriately as possible and connected at the proper places of the control circuitry. The feedback network played a very important role in the scheme. Both current and speed feedback were taken for input adjustment. As usual, both the parameters were converted into voltages prior to the comparison with some value of the concerning parameter (e.g. reference current for current feedback and set speed for speed feedback). The block representation of the overall scheme is shown in Fig. 1.

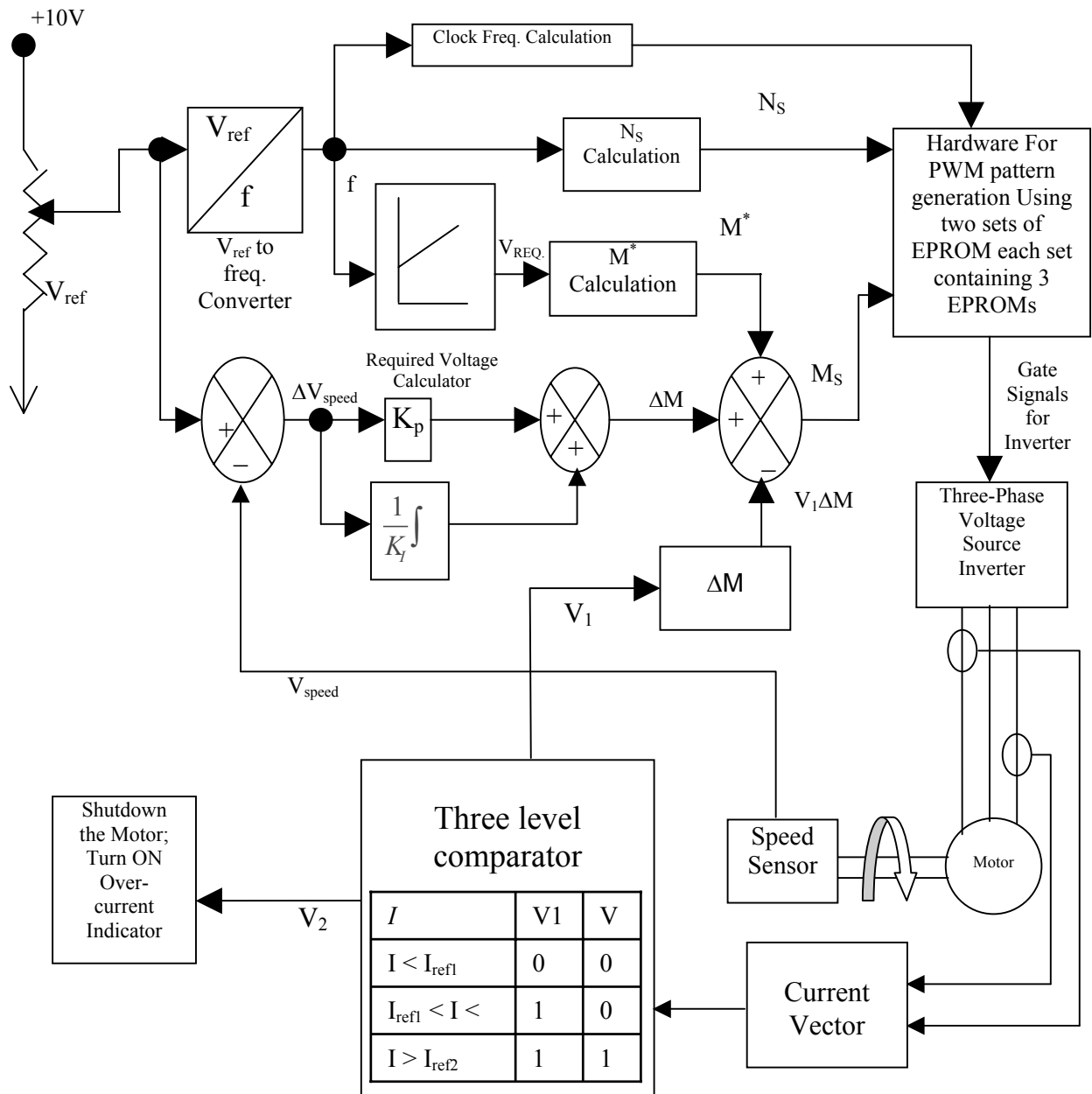


Fig. 1. The Control Scheme Used for Driving ASD

A speed-varying knob is used for varying the speed and setting speed at a definite position. An input potentiometer, connected to this speed-varying knob gives a proportional analog voltage (V_{ref}), corresponding to the set speed. With this set command of speed, a frequency is chosen. For this chosen value of frequency, decision is made about N (number of carrier pulses in a fundamental period), M^* (modulation index tag) and carrier frequency. These three parameters are fed to the EPROM's to determine the proper addresses, which is required for the generation of appropriate PWM patterns for generating

required voltage at required frequency, for speeding up the motor at the targeted speed. The detail of the addressing technique is discussed elsewhere [5]. With the generated PWM patterns, the three-phase inverters are switched through proper isolations and amplifications. The motor rotates according to the frequency and modulation index of the PWM waves.

This is how the forward path of the scheme works (open loop). To actuate and stabilize the speed of the motor, a feedback system must be placed between the output and input (i.e. closed loop system) to enable the microcontroller in

continuous monitoring of the system. Several standard feedback combinations are used in the scheme. Of them, PI controller (proportional and integral), associated with a speed sensor, three level current controller worth mentioning.

The speed is converted into a voltage (digital in our case) and is compared with the input reference voltage, found from the input potentiometer. The difference voltage is processed using PI controller technology and is used for increasing M (if speed is lower than the set value) or decreasing M (if speed is higher than the set value).

There is also a feedback network for abnormal current protection. The line currents of the motor are sensed and are compared with a preset reference value (I_{ref1}). If the motor current exceeds the expected level (which is equal to the motor current at 1500 rpm rotation with full load, 500W, condition), an error voltage will be generated. This error voltage then is used for a reduction of modulation index (ΔM) from the original computed one (M^*). As the modulation index determines the input voltage of the motor, reduction in it will obviously decrease speed in that overload condition. Therefore, the output power requirement will be lower; hence, there will be a reduction in motor current. Thus the motor current will be brought back to normal level if exceeded. Reduction of the modulation index is done on the basis of principal of integral (I) controller.

If however the motor current exceeds the specified limit (150% of the rated value), it becomes imperative to shut down the motor. This is done by comparing the current with another preset reference value, I_{ref2} .

Thus, we have applied three-level controller mechanism in this current feedback path. If the error is within the dead band (in the normal current condition), no control is necessary. However, if it crosses first limit, voltage is reduced, for bringing the motor current back to the safe limit, and if current crosses a second limit, a mechanism is applied to shut the motor down instantly by removing the gate pulses. An over current indication is also provided when current exceeds this limit given.

3. ALGORITHM USED FOR CONTROLLING ASD

In this paper, only the algorithm involved in the controlling of ASD will be discussed. As mentioned earlier, the others are reported

elsewhere. The algorithm given is used in an AT89C51 microcontroller. Various predefined variables [6] are used here.

3.1 Algorithm

1. Set SP = 00H;
2. Set S = 1 [S indicates the Present slot number, S = 1 for 0 ~1500 rpm and 2 for 1500~5000 rpm]
 - N = 0 [Frequency Tag]
 - M = 0 [Modulation Index Tag]
 - PS = 1; [Past 'table position Index'; higher PS means higher speed]
 - ADC_upper_thres = 92H [Holding the ADC upper threshold value of Speed Table Position, initialized for position 1]
 - ADC_lower_thres = 00H [Holding the ADC lower threshold value of Speed Table Position, initialized for position 1]
3. Set T2MOD = 02H
4. Set TIM = FF00H; ['TIM' holding the register value to be loaded for Timer 2 of the microcontroller]
5. Call Set timer 2 subroutine.
6. Output N, M sequentially through Port 0. [Motor will speed up to around 50 rpm]
7. Set Past = 0; ['Past' holding the past ADC value]
8. Read data from ADC.
9. If ADC = Past, Call delay subroutine; Set S =1 and N = 0, M = 0; Output N, M sequentially through Port 0; go to Step 8
10. Read ADC.
11. If ADC = Past, Call delay subroutine; go to Step 10.
12. If ADC >1228; S = 2
13. If ADC <1228; S = 1
14. Go to the position of Speed Table indicated by PS.
15. If ADC > ADC_upper_thres, move down by one step (increase speed) through the table; else go to Step 21.
16. Output N, M value in that position of the table, call delay subroutine.
17. If REQD = TIM, go to Step 20. ['REQD' holding the required register value for Timer 2 of the microcontroller in the present speed position]
18. If REQD > TIM, increase TIM by 1; Call set timer 2 subroutine; call delay subroutine, go to Step 17 else go to Step 19.
19. If REQD < TIM, decrease TIM by 1. Call set timer 2 subroutine, call delay subroutine, go to Step 17.

20. Go to Step 26.
21. If $ADC < ADC_lower_thres$, move up by one step (decrease speed) through the table; else go to Step 27.
22. Output **N**, **M** value in that position of the table, call delay subroutine.
23. If $REQD = TIM$, go to Step 26.
24. If $REQD > TIM$, increase **TIM** by 1; Call set timer 2 subroutine; call delay subroutine, go to Step 23; else go to Step 25.
25. If $REQD < TIM$, decrease **TIM** by 1, Call set timer 2 subroutine; call delay subroutine, go to Step 23
26. Save the present table position Index to 'PS'.
27. Save Past = ADC. ; "Past" holding past set ADC value
28. Read Speed Data and save it on 'SPD'.
29. Bring SPD and ADC on same scale.
30. If $ADC = SPD$, go to Step 10.
31. If $ADC > SPD$, increase **M**. Call Delay Subroutine. go to Step 30.
32. If $ADC < SPD$, decrease **M**. Call Delay Subroutine. go to Step 30.

Timer 2 Setup Subroutine:

33. Clear T2CON
34. Set $RACAP2 = TIM$
35. Set T2CON = 04H

Delay Subroutine:

36. Calculate f using:

$$f = \frac{24 \times 10^6}{4 \times (65536 - RACAP2) \times 256 \times N}$$

37. Set delay for 5/f times.

Current Interrupt Subroutine:

38. Decrease **M**.
39. Output **M** through Port 0.

4. FEATURES OF THE ALGORITHM USED

As mentioned earlier, to be efficient in driving an ASD, an efficient algorithm is required. The algorithm is used for selecting proper voltage and frequency so that the induction motor runs at set speed. For minimizing the starting current, the set speed is achieved gradually. The change in speed with change in command was also done steadily. And under any condition, the current is not allowed to

cross a well-defined limit, when the ASD is operating at a particular speed.

5. RESULTS

Table 1

Input Source	240 Volt; 50 Hz			
Speed Control	Done with a Knob of a potentiometer resulting in 0~10V variation corresponding to 0~5000 rpm speed setting			
Speed Regulation and Output Torque	Set Speed	Output Speed	% Deviation	Torque
	150	159	6	2.5
	500	494	1.2	3.2
	1000	990	1	3.1
	1500	1490	0.67	3.4
	2000	1953	2.4	2.3
	2500	2429	2.8	1.9
	3000	3040	1.3	1.5
	3500	3550	1.4	1.4
	4000	4051	1.25	1.2
	4500	4550	1.11	1.0
	5000	5100	2.00	0.9

Speed Deviation & Output Torque

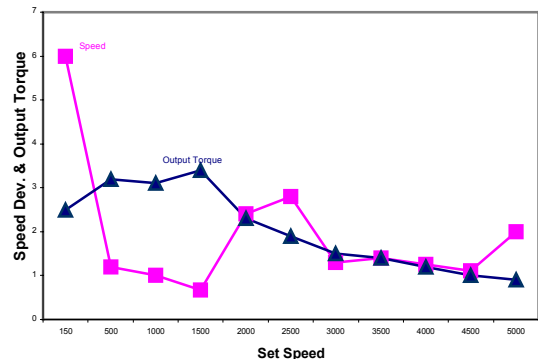


Fig. 2. Speed Deviation and Output torque at different set speeds

6. CONCLUSION

A fully functional ASD is designed and tested. The implementation of the whole scheme is performed with low cost circuits and components. The performance of the drive is tested for speeds ranging from 50 rpm to 5000 rpm. The scheme adopts a novel strategy for high-resolution PWM pattern generation and is implemented with an 8-bit microcontroller (AT89C52) at a wide speed range.

ACKNOWLEDGEMENT

This work was sponsored by BUET, Rahim Afroze Battery Co., and Siemens Bangladesh Limited.

REFERENCES

- [1] S. R. Bowes, "Advanced Regular Sampled PWM Control Techniques for Drives and Static Power Converters," *IEEE Transactions on Industrial Electronics*, Vol. 42, No. 4, August 1995, pp. 367-373.
- [2] S. R. Bowes and M. J. Mount, "Microprocessor control of PWM inverters," *Proc. IEE, B-Elect. Power Appl.*, vol. 128, no. 6, pp. 293-305, 1981.
- [3] S. R. Bowes, "Regular Sampled harmonic Elimination PWM control of Inverter Drives," *IEEE Transactions on Power Electronics*, Vol. 10, No. 5, September 1995, pp. 521-531.
- [4] S. R. Bowes, S. Grewal, D. Holiday, "Regular Sampled Harmonic Elimination Control of Single Phase Two Level Inverters," *IEE Proceedings-Electrical Power Applications*, 148 (2), 2001, pp. 309-314.
- [5] A. E. Islam, et al., "Low cost implementation of High resolution PWM Scheme for ASD", Proceedings of ICECE 2004, pp 385-389, December 28-30, 2004
- [6] Datasheet of Atmel AT89C52 Microcontroller.