A Simple Speed Feedback System for Low Speed DC Motor Control in Robotic Applications

R. V. Sharan, G. C. Onwubolu, R. Singh, H. Reddy, and S. Kumar
School of Engineering and Physics
University of the South Pacific, Suva, Fiji.
sharan_r@usp.ac.fj

Abstract-Robots often tend to swerve in an arc when one wants it to move forward or backward in a straight line. This is because the wheels do not rotate at the same speed. This research describes the design of a PIC microcontroller-based speed feedback system for low speed DC motor control to be used for robotic applications. Such a system, integrated with an appropriate motor control circuitry, is expected to dramatically enhance the performance of a DC motor. A feature of this design is that it uses basic principles of speed measurement. The experimental results show that the speeds measured by the PIC-microcontroller are reasonably close to the values measured by the tachometer.

I. INTRODUCTION

The strategy usually used for direct current (DC) motor control in robotic motion and similar applications is pulse width modulation. This, however, uses an open loop control scheme. In open loop control, there is no feedback from the motors informing the robot's program how fast the wheels are turning. Rather, the motors are just given different commanded voltages. But depending on terrain, surface obstacles, slippage in wheel contacts, or load on the robot, the commanded voltages do not necessarily imply particular speeds [1].

In contrast to the open loop system, a closed loop system utilizes a feedback signal as a measure of the actual output response. A feedback control system tends to maintain a prescribed relationship of one system variable to another by comparing functions of these variables and using the difference as a means of control [2].

The feedback data to be obtained is the velocity of the DC motor. To implement a velocity control algorithm, the robot needs sensors on the wheels, such as shaft encoders. Such feedback enables what is known as closed loop control algorithm [3].

II. FEEDBACK SYSTEM DESIGN

An open loop control system of a DC motor normally consists of a pulse width modulator and motor control circuitry. Information about the speed of the motor is obtained by adding a feedback system, as shown in Fig 1, to form a closed loop system.

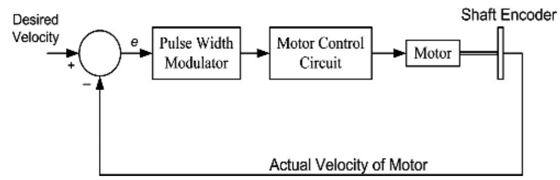


Fig. 1: A simple closed loop system for DC motor control

The basic idea of the control loop is to take in the desired velocity command, generate appropriate duty cycle (the on-time of the motor) based on the command to operate the motor, see how fast the motor actually spins, and then measure that speed and compare it to the commanded speed. The difference is referred to as the error signal and it is either positive or negative [1].

If the actual speed of the motor is less than the desired speed, the difference speed is positive, and so the duty cycle is increased to increase the actual speed of the motor to the desired speed. If the actual speed is greater than the desired speed, the error signal is negative and the duty cycle is reduced to slow the motor [4]. The amount by which the duty cycle is varied to match the actual and desired speed is purely based on the designed controller.

III. SHAFT ENCODER

The feedback transducer used for velocity is the shaft encoder. A shaft encoder is a sensor that measures the rotational rate of a shaft [5]. Typically, a shaft encoder is mounted on the output shaft of a drive motor or on an axle. The signal delivered by this sensor is a pulse train, which makes it appropriate for use in a digital system. Each time the shaft turns by a small amount, the state of its output changes from high to low or vice-versa. Thus, the rate at which pulses are produced corresponds to the rate at which the shaft turns.

Incremental shaft encoders contain a spinning disk that has slots cut in it. The disk attaches to the motor shaft and spins with it. An emitter is placed on one side of the disk's slots and a detector on the other. As the disk spins, the light passing through the disk is interrupted by the moving slots, and a signal in the form of a pulse train is produced at the

T. Sobh et al. (eds.), Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, 397–399.

output of the detector. By using a microcontroller to count these pulses, the speed of the drive wheel is found [1].

The emitter and detector sensors used for the research reported in this paper are the infrared emitting diodes and phototransistor respectively. The device number for the diode is OP140 and it emits infrared energy at 935 (nm) [6]. The phototransistor, of OP550 series, is spectrally and mechanically matched to the OP140 series of infrared emitting diodes [7].

The output from the phototransistor is a train of pulses with period, T seconds or $T/60$ minutes. Therefore, the speed of the motor in revolutions per minute is:

$$RPM = \frac{60}{NT} \tag{1}$$

where N is the number of slots on the shaft encoder.

If the diameter of the wheel attached to the shaft of the motor is d (meters), the circumference of the wheel = πd (m) and since one revolution represents a distance of πd (m), the speed now becomes:

$$v = \frac{\pi d}{NT} \tag{2}$$

where v is the motor speed in meters per second.

The precision of the calculated speed depends on the resolution of the shaft encoder. Increasing the number of slots on the shaft encoder increases the resolution of the encoder and vice-versa. An encoder with higher resolution would however require very accurate machining and buying one would even cost more. So there is a tradeoff between resolution and costs of the encoder. The encoder designed for this research, however, has 16 slots. This implies that there will be 16 complete pulses produced in one revolution or a sum of 32 high and low signals. Therefore, the minimum angular movement that is detected by the microcontroller is $360^o/32 = 11.25^o$ and since the circumference of the wheel is πd (m), the robot has a travel resolution of:

$$r = \frac{\pi d}{2N}\ (m) \tag{3}$$

If used with a wheel of diameter 0.2 m, the maximum travel resolution will be:

$$r_{(d=0.2m, N=16)} = \frac{\pi(0.2)}{32} \approx 0.0196\,\text{m}\,.$$

Therefore, the microcontroller will have knowledge of every 0.0196 (m) of movement of the wheel of the robot.

IV. MICROCONTROLLER

Microcontrollers are devices that have found extensive use in electronic products globally. They provide a method to learn about digital interfacing and programming, and also provide the capability to easily create applications that control real world devices [8].

The microcontroller used for the work reported in this paper is the Peripheral Interface Controller (PIC) which is developed by Microchip [9]. PIC is essentially an input/output controller and is designed to be very fast. Its program memory is made from flash technology, that is, it can be reprogrammed. The device used is PIC 16F877 which has an internal 16-bit timer. The timer is used to time a single pulse for calculating the speed.

V. EXPERIMENTATION AND RESULTS

The first phase of experimentation involved writing codes to count inputs to the PIC 16F877. Simple push button switches were used as input and the number of manual switch presses was compared to the counter value from the microcontroller.

The next phase involved modification of these codes to time a single pulse or count. The internal *timer1* of PIC was used for this purpose. Since this is a 16 bit timer, it can time a pulse up to a period of 2^{16} = 65536 (μs) = 65.536 (ms). Firstly, various frequencies from the signal generator were measured. This is done to show the accuracy of the frequency that is measured from the motor.

The relationship between the measured frequency using PIC microcontroller and the percentage error is shown in Fig. 2. The percentage error is given as:

$$\varepsilon = \frac{F_m - F_a}{F_a} \times 100\% \tag{4}$$

where F_m and F_a denote the measured and actual frequencies. It is observed that as the frequency increases, the error in the frequency measured from PIC increases linearly, that is, there is a linear relationship between error and frequency.

Furthermore, the shaft encoder was mounted on the shaft of a DC motor and it was driven at different speeds. The speed of the motor was first measured using the PIC microcontroller, and then a tachometer was used to compare the results. The results with an average of 10 runs are shown in Table 1.

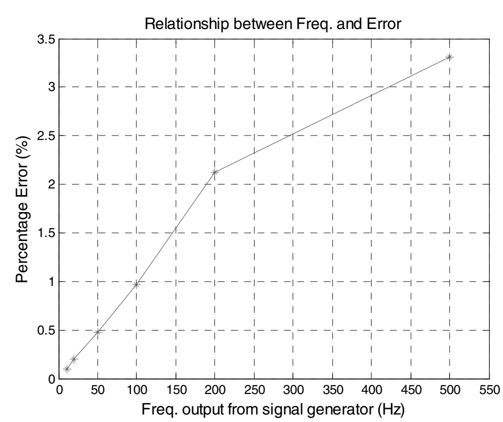


Fig. 2: Characteristics of the error associated with the frequency measured from PIC

Table 1: Measured speeds of the motor using the tachometer and PIC.

RPM (Tachometer) T_{RPM}	RPM (PIC) P_{RPM}	% Error
43**	43.5	1.16
46.6*	47.05	0.97
60.84**	61.26	0.69
73.55*	74.7	1.56

*clockwise **anticlockwise

The percentage error is defined as:

$$\varepsilon = \frac{M_{RPM} - T_{RPM}}{T_{RPM}} x100\% \qquad (5)$$

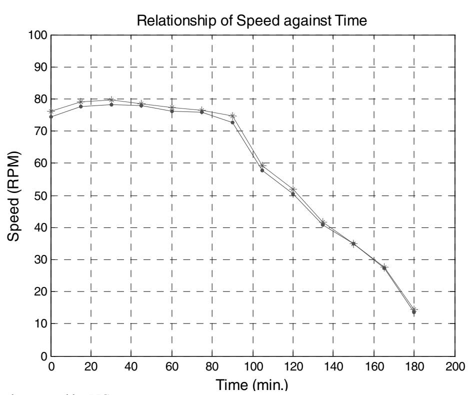
where M_{RPM} and T_{RPM} denote the measured speed using the microcontroller and the tachometer respectively.

The speed data obtained from the two sources have some differences but can be considered constant and thus, suitable for measuring speed of DC motor in the lower speed range. Also, a tachometer is basically used to record speed data whereas the speed readings from the PIC microcontroller can be used for speed control or position control of a motor or wheel of a robot.

Some further tests were carried out to justify the need for a closed loop system. Firstly, the battery that is used to power the DC motor under test was fully charged and then the motor was run at its maximum speed. The speed of the motor was recorded at intervals of 15 minutes using the tachometer and the PIC microcontroller.

A graph of time against speed was drawn, as shown in Fig. 3, to determine the relationship between the two variables as the battery voltage drops.

For the first 90 minutes, the speed of the motor is nearly constant. But after 90 minutes, there is a proportional decrease in speed with respect to time. This experimentation shows that the speed of the motor decreases with time as the battery voltage drops. Thus, there is a need for a closed loop system to be incorporated to maintain the speed of the DC motor.



* Speed measured by PIC
• Speed measured by tachometer

Fig. 3: Plot of speed against time

VI. Conclusion

A simple feedback system using a shaft encoder is incorporated in an open loop system to monitor the speed of a DC motor. This information is vital for the control of a DC motor. It can be used to vary the duty cycle of the pulse signal so that the motor speed can be controlled. To maintain desired speed, regardless of terrain, means that the robot needs to calculate the speed of the DC motor to see how fast the wheel is turning and then update the pulse width accordingly. The maximum speed that the tested DC motor had was approximately 75 revolutions per minute at a frequency of about 20 Hertz. Since the PIC microcontroller is quite accurate in measuring frequencies up to about 500 Hz, the results obtained are very close to the tachometer readings.

This is a basic speed feedback system developed for low speed DC motors. More work can be done on this to increase the speed measurement range of this feedback system. PIC microcontrollers also offer interrupt features which can be explored further as a way forward in modifying this system.

References

[1] J. L. Jones, and A. Flynn, *Mobile Robots-Inspiration to Implementation*, A. K. Peters, U.S.A., 1993.
[2] C. Phillips, and R. Bishop, *Feedback Control Systems*, 4th edition, Prentice Hall, U.S.A., 2000.
[3] J. Borenstein, and Y. Koren, ‘A mobile platform for nursing robots’, *IEEE Transactions on Industrial Electronics*, 32(2), 158-165, 1985.
[4] P. Scott, *The Robotics Revolution: Complete Guide for Managers and Engineers*, Basil Blackwell, U.K., 1984.
[5] P. Smith, *Active Sensors for Local Planning*, World Scientific Printers, Singapore, 2001.
[6] Optek Technology, *GaAs Plastic Infrared Emitting Diodes*, Product Bulletin OP140A, May 1996. http://www.optekinc.com/pdf/Op140.pdf
[7] Optek Technology, *NPN Silicon Phototransistors*, Product Bulletin OP550A, June 1996. http://www.optekinc.com/pdf/OP550A.pdf
[8] M. Predko, *Programming and Customizing PIC micro MCU Microcontrollers*, McGraw Hill, U.S.A., 2002.
[9] Microchip Techonology Inc. http://www.microchip.com