A PID Controller for Real-Time DC Motor Speed Control using the C505C Microcontroller

Sukumar Kamalasadan
Division of Engineering and Computer Technology
University of West Florida, Pensacola, FL, 32513
Phone: (850) 857-6451, Fax: (850) 474-2804
Email: skamalasadan@uwf.edu

Abhiman Hande
Electrical and Computer Engineering Department
Lake Superior State University, S.S. Marie, MI 49783
Phone: (906) 635-2598, Fax: (906) 635-6663
Email: ahande@lssu.edu

Abstract

This paper presents a real-time DC Motor speed controller design using a microcontroller-based network system. The design architecture was developed using two Phytec evaluation boards each having an Infineon eight-bit C505C-L microcontroller. The system detects the real-time speed of the motor using the sensor device and then transfers data to the first Phytec board's microcontroller using serial communication. This data is processed and transferred to the second Phytec board's microcontroller using a Controller Area Network (CAN) communication scheme. The second microcontroller uses the received data to calculate the real-time control value to monitor and keep the motor speed constant based on a command signal. Data is then transferred back to the first Phytec board's microcontroller using CAN and is utilized to change the motor voltage such that its speed is constant. The importance of the proposed design architecture is its ability in controlling precisely the motor speed and/or direction especially when used in modern automobiles where the CAN protocol is quite popular. Moreover, the proposed real-time controller approach is based on the closed loop feedback error principle unlike the existing open loop designs. The paper details the system design and the experimental results that were obtained.

Key Words: Real-time DC motor speed control, C505C microcontroller, Controller Area Network

1. Introduction

Rapid progress in microelectronics and microcontrollers in recent years has made it possible to apply modern control technology to automobiles that need real-time control. Automotive technology uses several electronic control units (ECU) to control efficient and reliable operation of key components namely, the engine, transmission, anti-lock braking system (ABS), cruise, steering, vehicle traction, and entertainment [1]. DC motors control, many of these operations and therefore there is a need for implementing effective control strategies for digital control of these motors. Currently, auto manufacturers use multiplex buses for integrating all their vehicle electronics and instead of using the proprietary buses for the transfer of data within the vehicle, the Controller Area Network (CAN) proposed by Robert Bosch [6] is becoming an industry standard. Therefore, it is quite important to develop real-time DC motor control strategies such that these devices are effectively integrated with their control electronics using the CAN network protocol. Realizing the fact that large number of motors are utilized in modern vehicles and there is especially a need to control these small motors using a common bus with interrupt priorities, such real-time control is extremely essential.

Several research efforts have shown the ability of using digital signal processing (DSP) kits for real-time motor control such as a proportional-integral-derivative (PID) controller using the TMS320c31 DSK kit [2]. Subsequently it also addressed some steps into adaptive control using the same DSP kit [3]. However, many such processing kits do not have the ability of high-speed networking and use open loop control strategies. As opposed to those research directions, the proposed system makes use of the high speed CAN protocol that has built-in error management and therefore reduces the effect of electromagnetic interference (EMI) [6]. The system uses the Infineon eight-bit C505C-L microcontroller that has several peripherals including three timer/counter units, an analog-to-digital (A/D) converter, a serial communication interface (USART), and supports the CAN protocol [4,5].

This paper proposes a real-time DC motor control design approach using the CAN protocol to communicate between a local microcontroller that senses motor speed and controls a motor driver circuit, and a remote microcontroller that contains an embedded closed loop digital controller for the motor speed correction. At first, a PID Controller is designed using MATLAB® and Simulink® with a practical motor model. The transfer function model of the motor is then used to design the digital controller. Subsequently, this developed design is used to generate the sensor architecture and software program, which is loaded to the first microcontroller. This microcontroller acts as the integrator between the motor and the digital controller. The motor speed is then transferred to the second microcontroller (using a CAN bus) that is loaded with the controller architecture software program. Based on changes in the motor speed, which is recorded using the first microcontroller, the control value is generated which in turn is used to correct any speed deviation. The whole process including the network bus monitoring is carried out in real-time.

The paper is organized as follows. In section 2, a generic control schematic using two microcontrollers communicating through a CAN bus is discussed. This section also details the DC motor speed control strategy using pulse width modulation (PWM). In section 3, the digital PID controller design concepts and the simulation test results of the PID controller on the DC motor model using the MATLAB environment is discussed. Section 4 details the flowchart for each microcontroller and their communication using the CAN bus, while section 5 shows the test results observed. Lastly, section 6 points out the conclusions inferred from the project.

2. Motor speed control using CAN

Figure 1 shows the scheme used for controlling the motor speed using two microcontrollers communicating with each other through a CAN bus. As can be seen, the motor speed is sensed and transmitted to the CAN bus by M_1. M_2 which generates the control value in real-time, aided by the digital controller utilizes this information. This is then transferred to M_1 via the CAN bus, and the driver circuitry which normally changes the voltage to stabilize the speed uses this information. The CAN bus monitor maintains the priority of different motor control operations and schedules the interrupt level for the specific operation domain.

Control of DC motors by PWM is very well known [7]. In the process of varying the pulse by controlling the switching of the input voltage for the off and on duration, a time dependent varying output voltage can be achieved. For example, Figure 2 shows a square wave PWM pulse with 50% duty cycle.

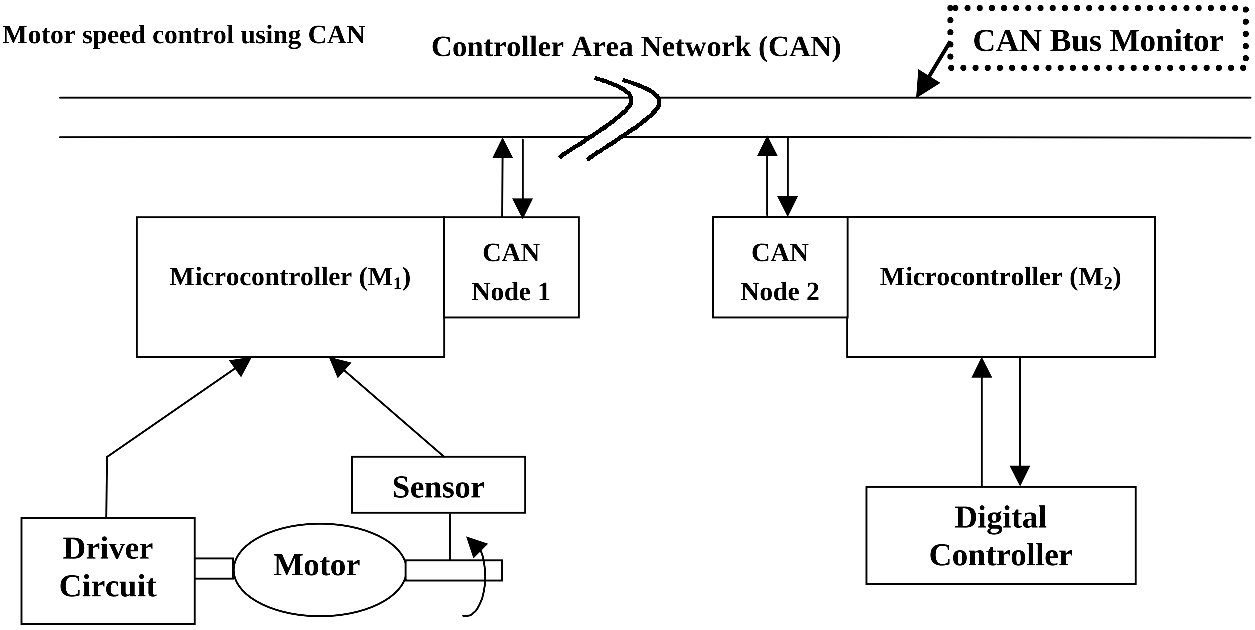

Figure 1: Generic schematic diagram of the real-time DC motor control

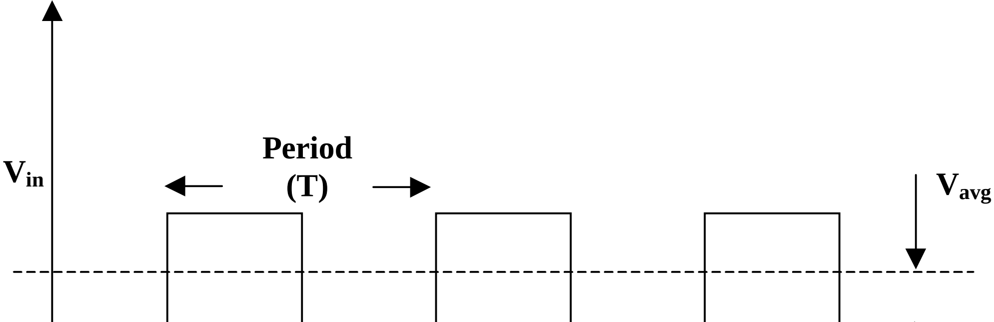

Figure 2: Square wave with 50% duty cycle

As shown, if the input voltage ‘V_{in}’ can be switched on and off frequently at the uniform rate then the total period ‘T’ will be:

$$T = T_{on} + T_{off} \rightarrow (1)$$

where: T_{on}= ON time and T_{off} = OFF time.

The average output voltage in this case is 0.5 * V_{in}. In general, the output voltage will be:

$$V_{avg} = \left(\frac{T_{on}}{T_{on} + T_{off}} \right) V_{in} \Rightarrow V_{avg} = (D) V_{in} \rightarrow (2)$$

where: Vavg = average output voltage and D= duty cycle.

There are two possible ways to control the speed of the motor viz. open loop control and closed loop control. In open loop control, the control value is not dependent on the output or the speed of the motor, whereas in closed loop control in which the control value is dependent on the speed of the motor. In the proposed approach a closed loop speed control with a digital PID controller is designed. A potentiometer was used as the reference command signal to set the input voltage at various levels as required. Further, the control value obtained from microcontroller M_2 is utilized to generate the average output voltage for adjusting the duty cycle (D) in order to maintain constant motor speed.

3. Digital controller design and simulation

Figure 3 shows the block diagram of the proposed digital PID controller, where R(s) is the reference input, y(s) is the system output, C(s) is the controller transfer function, and H(s) is the feedback loop (sensor) transfer function.

The digital PID Controller has the following form [8],

$$C(z) = K_P + K_I \left[\frac{T}{2} \frac{z+1}{z-1} \right] + K_D \left[\frac{z+1}{Tz} \right] \rightarrow (3)$$

where:

‘K_P’, ‘K_I’ and ‘K_D’ are the proportional, integral and derivative parameters of the controller and ‘T’ the sampling time.

The discrete form of the controller transfer function ‘C(z)’ can also be written as:

$$C(z) = \frac{a_0 + a_1 Z^{-1} + a_2 Z^{-2}}{1 - z^{-1}} \rightarrow (4)$$

where:

$$a_0 = K_p + \frac{K_I T}{2} + \frac{K_P}{T},\ a_1 = -K_p + \frac{K_I T}{2} - \frac{2K_P}{T} \text{ and } a_2 = \frac{K_P}{T}$$

In the ‘Z’ domain the above mentioned second order polynomial will be:

$$Y(z) = a_0 + a_1 Z^{-1} + a_2 Z^{-2} * C(z) \text{ and}$$
$$X(z) = (1 - Z^{-1}) * C(z) \rightarrow (5\text{-}6)$$

where: X(z) is the input to the controller and Y(z) the output of the controller.

Taking the inverse z-transformation, the following expressions are obtained:

$$Y(k) = a_0 g(k) + a_1 g(k-1) + a_2 g(k-2) \text{ and}$$
$$g(k)=X(k)+g(k-1) \rightarrow (7\text{-}8)$$

where: ‘k’ is the time constant. This gives a relation between the output of the plant and the controller.

The DC motor is modeled using the following transfer function:

$$G_{DC}(s) = \frac{22s}{(s+4)} \rightarrow (9)$$

In the digital domain this is translated with respect to a sampling time (T) of 250 msec as,

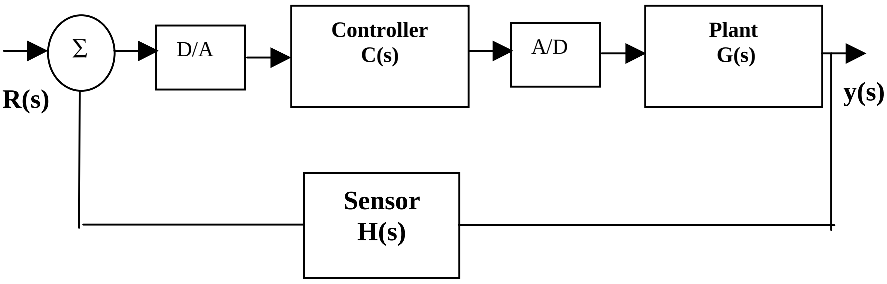

Figure 3: Discrete PID controller structure

$$G_{DC}(Z) = [1 - Z^{-1}]\left[\frac{22}{1-0.368Z^{-1}}\right] \rightarrow (10)$$

Combining the motor model and the controller equation the closed loop system will be,

$$\frac{y(Z)}{R(Z)} = \frac{C(Z)*G(z)}{1+C(Z)G(Z)} = S(z) \rightarrow (11)$$

with the characteristic equation as $1 + C(z)G(z)$. The controller parameters are designed such that the steady state error $e(Z) = R(z)[1 - S(z)]$ asymptotically tends to zero i.e. $\frac{1}{1+C(z)G(z)} = 0$ for a unit step input. Combining equations (3) and (10),

$$1 + C(z)G(z) = 1 + 22a_0 + (22a_1 - 0.368)Z^{-1} + 22a_2Z^{-2}$$

or

$$1 + 22a_0 + (22a_1 - 0.368)Z^{-1} + 22a_2Z^{-2} \rightarrow (12)$$

The developed DC motor model and the closed loop system are then used to generate the coefficients of the digital PID controller using MATLAB®. Figure 4 shows the settling time for the model for a motor input set point of 15 revolutions per second (rps). Considering this settling time for the practical DC motor model, the coefficients of the discrete PID constants are formulated for the closed loop system. Further, using these coefficients from (8), the following coefficient matrices are also obtained for the DC motor equation.

PID_Controller = {1.001, -0.996, 0, 0, 0} / {1, -1, 0, 0, 0}; (which shows the PID Controller value for the time span desired)

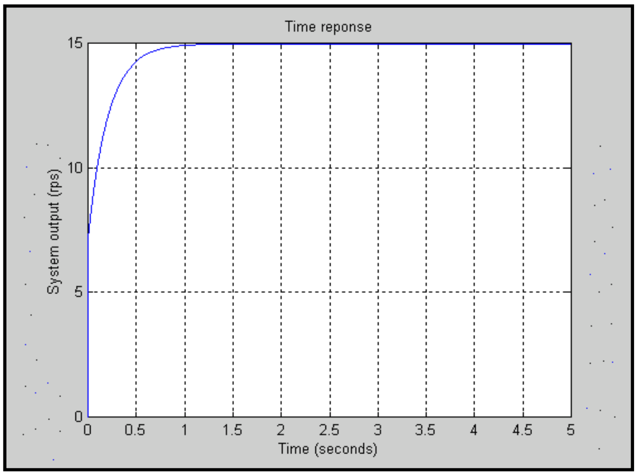

Figure 4: Step response of the DC motor model

DC Motor Equation = {0, 1.8151, -1.8060, 0, 0}/{1, -1.6703, 0.6703, 0, 0}; (which shows the motor characteristic that needs to be controlled)

4. Software development and implementation

The actual implementation of the PID controller was done using the Digital Application Engineer (DAvE) software from Infineon Technologies [9] and the µVision2 software package from Keil software Inc. [10].

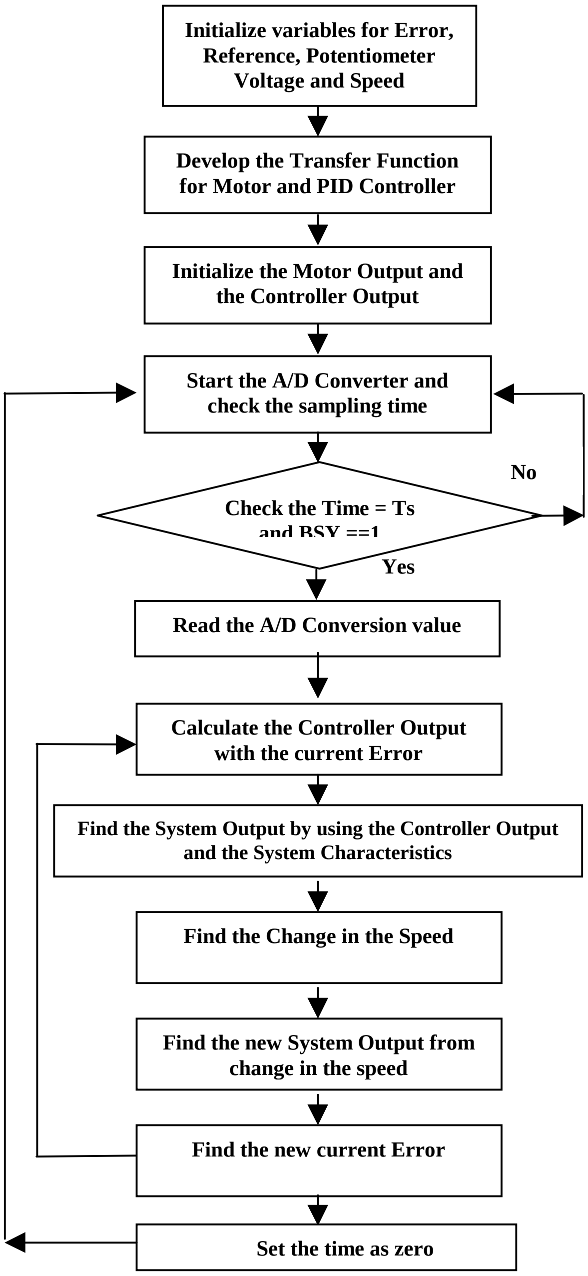

Figure 5: Algorithmic flowchart

While DAvE was used to set the peripheral registers of the microcontroller, µVision2 was used for writing and compiling the 'C' code. Figure 5 shows the flowchart of the algorithm that was developed. The controller equations were loaded into the flash memory of the microcontroller M_2. For simulation purposes, the transfer function equations of the motor were loaded into the flash memory of the microcontroller M_1. At first, the system output from the transfer function was utilized to generate the controller value from the microcontroller M_2. The motor speed was read at 250 msec intervals using the A/D converter of M_1, and the digital inputs were converted into a CAN data frame for two bytes. The object consists of these two bytes and an 11-bit identifier 0x111. This data is then transferred to the CAN bus which is identified by the CAN bus monitor as the system output. Microcontroller M_2 waits for a CAN message with identifier 0x111. Therefore, as soon as the above object with identifier 0x111 appears on the CAN bus, it is captured by M_2. M_2 uses this received data to generate the controller value. The controller value is calculated using the difference between the reference speed (based on the potentiometer setting) and the actual speed measured by M_1. This error is then used to generate the PID constants, which in turn develops the duty cycle (D) equivalent for the discrete closed loop control.

The required duty cycle is then transferred back to the CAN bus using another 11-bit identifier (0x222) and two bytes of data from CAN Node 2 (refer Figure 1). Every second, the CAN nodes (both Node 1 and 2) identifies their object and subsequently the data, and interact with each other to develop the required motor driver voltage and in turn, the required motor speed. This overall process of feedback control is done in real-time such that the motor speed stays at the same set value.

5. Practical implementation requirements

In order to implement the proposed scheme with a practical DC motor instead of reading the speed from the output of the modeled equation, the recorded speed was taken from the motor sensor. Further, the output of microcontroller M_2 is utilized to generate the average voltage with the help of a driver circuit. The following subsections discuss the process.

Feedback sensor:

The feedback sensor senses the speed from the practical motor. There are various types of sensors that can be used for sensing the speed accurately during the desired time. One such sensor is an optical infrared interrupter switch as explained in [11]. It is an open collector buffered digital output that is compatible with transistor-transistor logic (TTL). A key advantage of this type of sensor is that it is not connected physically to the motor shaft and therefore does not bear any additional load to the motor. However, an optical sensor will be an ideal choice if it needs to be used in an oil-laden environment.

Input driver circuit:

The input driver can be directly developed from the microcontroller port. One important factor is the load current requirement of the motor. It is quite obvious that the requirements of any practical motor are higher than any microcontroller can supply. Thus, the current amplification procedure is essential even though the digital value of the motor voltage can be derived from the output port. Transistors or integrated circuit (IC) chips can carry out the amplification procedure. One such implementation is done in [11] using an L293D driver. This device can supply up to 600 mA current per channel. The output port of the microcontroller can be directly connected to this device for driving the motor.

6. Simulation and test results

To test the system, the reference speed was set to about 4.5V (about 50% D). The motor speed was then displayed on a desktop computer using the HyperTerminal software, and transferred to microcontroller M_2. The controller software embedded within M_2 calculates the necessary duty cycle required for maintaining the motor speed at the required constant value. Figure 6 shows the actual output voltage waveform and the input set point. The codes were generated using the DAvE and µVision2 software programs. The software was written in ‘C’ language.

The proposed design uses a Dearborn Protocol Adapter II (DPA-II) to monitor the CAN system bus and uses a practical DC motor model. Figures 7 and 8 show the simulation results for a change in the speed value and the motor speed corresponding to the above change. As can be seen from Figure 7, a speed change of around 1.5 – 1.7 rps is continuously fed using a potentiometer. The remote controller has been applied in the forward loop based on the DC motor model and the updated value of the controller is used each time as the motor input using the CAN bus. The speed variation at each instant along with the controller action is observed. It was noted that the motor output is within the range of set point speed, which is 14.5 - 15.5 rps. Thus, the controller action is effective as the actual speed without the controller should drop to 13.3 – 13.5 rps during the snap-shot time span of 1 second. Figure 8 also shows the controller response during this period.

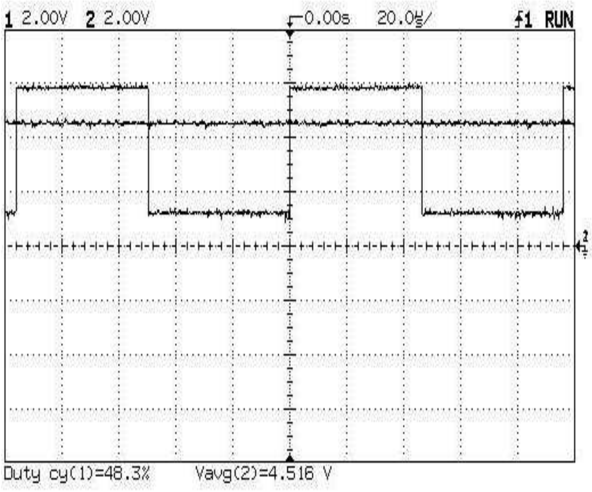

Figure 6: Output and input voltage changes

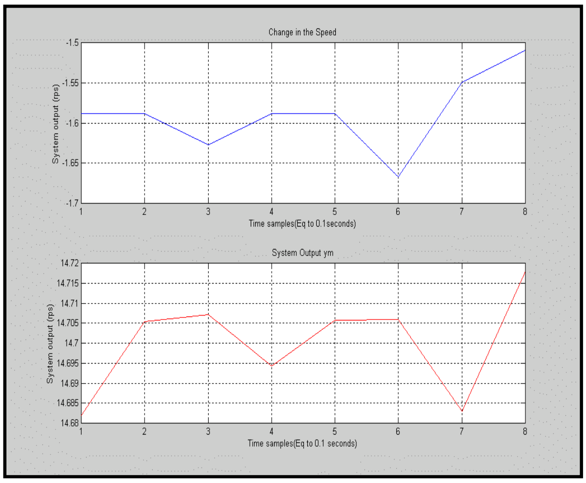

Figure 7: Motor speed changes

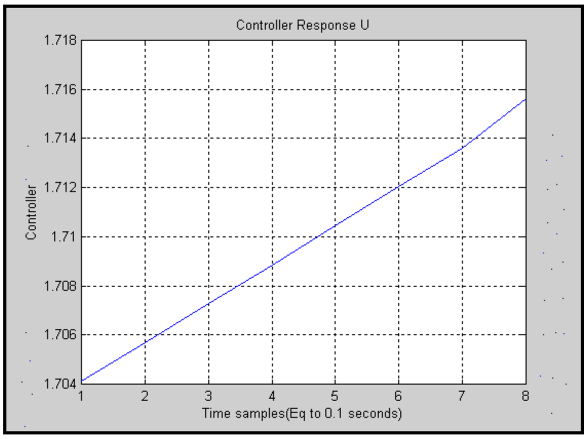

Figure 8: Controller response

7. Conclusions

A real-time PID digital controller has been developed that controls the speed of a DC motor in a closed loop manner. A CAN differential bus is used for transferring information between a local microcontroller that senses the motor speed and a remote microcontroller that contains the logic of the digital controller. The proposed scheme works as a remote controller, which communicates with a local DC motor or its model using the CAN protocol. The main investigation focused on developing a system that corrects changes in motor speed due to extraneous disturbances in an extremely rapid fashion. Therefore, microcontrollers that had full duplex CAN communication capability were chosen. The output results have shown that a 5% change in speed is brought back to zero, and the motor speed settles to its constant value of 15 rps within 1 second using the proposed controller scheme. These observations clearly show the ability of the real-time controller to constantly keep the output variable within limits when there is a change or disturbance in the output, and in turn maintain the speed of the motor at a constant value. The most critical aspect that requires close consideration is the sampling time.

References

[1] W.B.Ribbens, "Understanding Automotive Electronics", SAE Publications, 1998.

[2] J.Tang and R.Chassaing, "PID Controller Using the TMS320C31 DSK for Real-Time DC Motor Control" Proceedings of the 1999 Texas Instruments DSPS Fest, Houston, Texas, August 1999.

[3] J.Tang et. al., "Real-Time Adaptive PID Controller Using the TMS320C31 DSK", http://www.integrated-dsp.com/software/Publications/FEST2k.PDF, November 2002.

[4] "C505C 8-Bit CMOS Microcontroller User's Manual 08.97", Siemens, August 1997.

[5] "KitCON Hardware-Manual Version 3.0", PHYTEC Inc., June 1997.

[6] "CAN Specification, Version 2.0", Robert Bosch GmbH, Postfach 300240, D-7000, Stuttgart, September 1991.

[7] P.C.Sen, "Principles of Electric Machines and Power Electronics", Second Edition. John Wiley & Sons Inc., 1997.

[8] G.F.Franklin, J.D.Powell, and M.L.Workman, "Digital Control of Dynamic Systems", Addison-Wesley, 1998.

[9] "Digital Application Engineer – DAvE", Infineon Technologies,http://www.infineon.com/cgi/ecrm.dll/ecrm/scripts/search/prom_page_dispatcher.jsp?oid=29504&search_string=DavE, 2004.

[10] "µVision2 IDE and Debugger", Keil Software Inc., www.keil.com/uvision2, 2004.

[11] M.M.Jamali et. al., "A CAN based Real-Time Embedded System for DC Motor Control", SAE 2002 World Congress, Detroit, Michigan, March 4-7, 2002.