Review: Real Time Implementation of FPGA Base System - Speed Control of DC Motor using PID Controller

Mr. Satish B. Tarpe[1], Dr. Ram S. Dhekekar[2]
[1](Digital Electronics Engineering, Shri Sant Gajanan Maharaj College of Engineering, India)
[2](Digital Electronics Engineering, Shri Sant Gajanan Maharaj College of Engineering, India)

Abstract

This paper describes an implementation proportional-integral-derivative (PID) controller for speed control of DC motor, based on field programmable gate array (FPGA) circuit. The proposed scheme is aimed to improve the tracking performance and to eliminate the load disturbance in the speed control of DC motors. The proportional-integral-derivative (PID) controller is designed using the MATLAB tools. PID controller hardware is synthesized, functionally verified and implemented using Xilinx Integrated Software Environment (ISE) Version 9.1. The real time implementation of these controllers is made on Spartan-3E FPGA starter kit. Its primary goal is to implement a digital "Proportional-Integral-Derivative" (or PID) controller in an existing DC motor system with this unwritten economical specification that cost must be minimized.

Keywords - *DC motor, FPGA, PID controller, Spartan 3E kit, VHDL language*

I. INTRODUCTION

In all fields of engineering, many different solutions can be presented for the same problem. The goal of an engineer is twofold. The first and most distinct goal is to identify and design possible solutions to a complex problem. The second goal, which is perhaps less obvious, is to choose which solution is best suitable to most economically meet project specifications. Intelligent controllers as new technologies have recently been applied to electrical power control systems in general and motor control systems in particular [1]. Many digital techniques have been used to implement a digital controller. However, the conventional proportional–integral–derivative (PID) controller is still being a key component in industrial control systems, because it is simple and provides useful solutions to many industrial processes [2].

Higher density programmable logic devices, such as field programmable gate array (FPGA) can be used to integrate large amounts of fuzzy logic in a single integrated circuit (IC) [3,4]. FPGAs are one of the fastest growing parts of the digital integrated circuit market in recent times. Reconfigurable FPGA systems provide additional flexibility and can adapt to various computational tasks through hardware reuse. Therefore, FPGA becomes one of the most successful technologies for developing the systems which require a real time operation [5–7].

II. PID CONTROLLER

A PID controller, as its name suggests, provides proportional, integral, and derivative compensation to an existing system. These three forms of compensation increase system performance in a variety of ways. Proportional control can both increase gain margin and stabilize a potentially unstable system. Integral control can minimize steady state error. Derivative control can increase system speed by increasing system bandwidth. One drawback of PID control is overall complexity. This results in very expensive means of implementing a digital version of a PID controller.

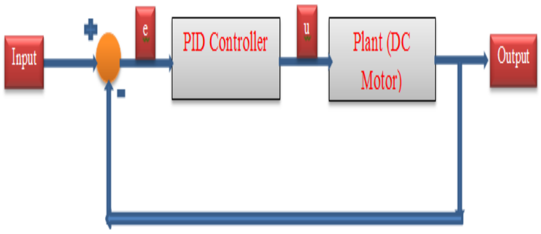

Fig. (1): System for Speed Control of DC Motor

A system with the PID as controller and the DC motor as plant is shown in Fig. (1). Of the many possibilities, digital signal processors (or DSP's) are the most widely used to solve this problem, however other possibilities exist which may be more cost-effective. Implementation of any complex digital controller must be done by means of some form of computer. Typical microcontrollers, while cheap, do not normally provide enough processing power to effectively perform all but the simplest calculations real-time. Digital signal processors, on the other hand, are designed to implement complex algorithms quickly. The major drawback of DSP's, however, is cost. This project will attempt to find a median between these two extremes of performance and cost. The proposed solution is to design a special-purpose computer whose only purpose is to quickly execute the complex PID algorithm. This computer will be designed using the IEEE 1076-1987 standard known as VHDL (or Very High Speed Integrated Circuit Hardware Description Language), and will be implemented on an FPGA (or Field Programmable Gate Array)

III. THE THREE-TERM CONTROLLER

The transfer function of the PID controller looks like the following:

$$K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{n} \quad \text{---(1)}$$

Where,

Kp = Proportional gain

KI = Integral gain

Kd = Derivative gain

First, let's take a look at how the PID controller works in a closed-loop system using the schematic shown above. The variable (e) represents the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal (u) just past the controller is now equal to the proportional gain (Kp) times the magnitude of the error plus the integral gain (Ki) times the integral of the error plus the derivative gain (Kd) times the derivative of the error.

$$u = K_P e + K_I \int edt + K_D \frac{de}{dt} \quad \text{--- (2)}$$

Where, e= reference speed – actual speed----(3)

This signal (u) will be sent to the plant, and the new output (Y) will be obtained. This new output (Y) will be sent back to the sensor again to find the new error signal (e). The controller takes this new error signal and computes its derivative and it's integral again. This process goes on and on.

IV. THE CHARACTERISTICS OF P, I, AND D CONTROLLERS:

A proportional controller (Kp) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (Ki) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (Kd) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of controllers Kp, Kd, and Ki on a closed-loop system are summarized in the table (1) shown.

CL Response	Rise Time	Overshoot	Settling Time	Steady State Error
K_P	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Small Change	Decrease	Decrease	Small Change

Table (1): Effects of Kp, Kd, and Ki on a closed- loop system

Note that these correlations may not be exactly accurate, because Kp, Ki, and Kd are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you are determining the values for Ki, Kp and Kd.

The goal of this paper is to show you how each of Kp, Ki and Kd contributes to obtain

- Fast rise time
- Minimum overshoot
- No steady-state error

V. MODELING OF PI CONTROLLER USING VHDL

The standard representation of a PI controller is given by Eq. (1); where, there are two main components that can be identified; they are the proportional and integral terms. The error signal e(t) is defined by Eq. (3), and it is the difference between the process output and the desired speed. The integral term was obtained using the method of rectangular integration. In this equation, Kp and Ki, are parameters related to the gain of each term.

The algorithm given by Eq. (2) is seldom used in practice. In this work, for speed control of a DC motor, a cascaded control structure is usually preferred. This controller is known as parallelism controller which is being used [7]. By differentiating Eqs. (6) and (8) can be obtained, where UPI is the control signal.

$$\frac{du_{PI}(t)}{dt} = K_P \frac{de(t)}{dt} + K_I e(t) \quad \text{---(4)}$$

In discrete-time systems, Eq. (4) can be written as follows:

$$u_{PI}(kT) - u_{PI}(kT - T) = K_P(e(kT) - e(kT - T)) + K_I e(kT)$$

---(5)

VI. CONTROLLER DESIGN FOR FPGA IMPLEMENTATION

The Spartan-3E FPGA is embedded with the 90 nm technology at the heart of its architecture. This reduces the die size and cost, increases manufacturing efficiency, and addresses a wider range of applications. The Spartan-3E diagram, shown in Fig. (3), allows users to easily migrate to different densities across multiple packages and supports 18 different single ended and differential I/O standards [8].

The speed of the motor is measured by means of optical encoder and is given as an input to the expansion connectors on board (J1 6-pin Accessory Header) [9]. As shown in Fig. (2). The reference speed is assigned to motor internal configuration memory of the FPGA.

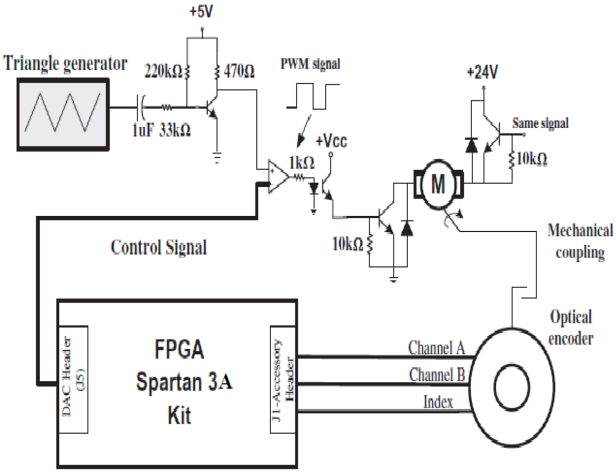

Fig. (2): Block Diagram of the DC Motor Speed Control using FPGA

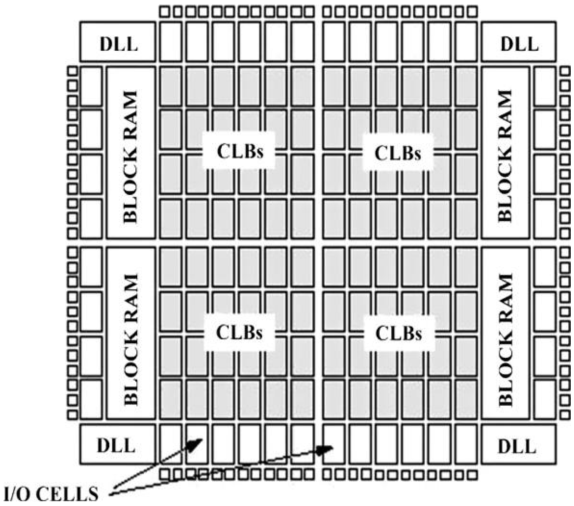

Fig. (3): Layout for FPGA

To verify the performance of the controller design on the hardware, the VHDL code (Bit file) is downloaded into the target FPGA device (Spartan-3E family XC3S500) using Xilinx ISE pack 9.1i [10-11]. The complete experimental system consists of a DC motor, FPGA kit, and the circuits of (PWM driver and H-bridge). The overall block diagram of DC motor speed control is shown in Fig. (2).

VII. CONCLUSION

The main contribution of this work is to present an approach of real time PID controller for DC motor using VHDL description and FPGA technology. PID based on FPGA has some advantages such as flexible design, feature accuracy, high reliability, cost improvement, and high speed. The performance of the PID is compared with the PI controller for reference speed and load changes. The satisfied ability of the system control with PID is better than PI controller. In a future work it is planned to investigate implementation of adaptive fuzzy controllers on FPGA.

REFERENCES

Journal Papers:

[1] Hamed B, Al-Mobaied M., "Fuzzy PID controllers using FPGA technique for real time DC motor speed control", *Intell Control Autom 2011;2(August):233–40*.

[2] Millan I, Montiel O, Sepulveda R, Castillo O. "Design and implementation of a hybrid fuzzy controller using VHDL", *Soft computing for hybrid Intel. systems*. Berlin, Heidelberg: Springer-Verlag; 2008, SCI 154.

[3] Kung YSh, Fung RF, Tai TY. Realization of a motion control IC for X_Y table based on novel FPGA technology. *IEEE Trans Industr Electron 2009*;56(1):43–53.

[4] Hamed BM, El-Moghany MS. Fuzzy controller design using FPGA for sun tracking in solar array system. *I.J. Intell Syst Appl 2012*;1(February):46–52.

[5] Monmasson Eric, Cirstea Marcian N. FPGA design methodology for industrial control systems – a review. *IEEE Trans Industr Electron 2007*;54(4).

[6] Sakthivel G, Anandhi TS, Natarajan SP. Real time implementation of a fuzzy logic controller on FPGA using VHDL for DC motor speed control, *G. Sakthivel et al.. Int J Eng Sci Technol 2010*;2(9):4511–9.

[7] Qu L, Huang Y, Ling L. Design of intelligent PID controller based on adaptive genetic algorithm and implementation of FPGA. *Berlin, Heidelberg: Springer-Verlag; 2008*, Part II, LNCS 5264.

[8] Shrikanth G, Subramanian K. Implementation of FPGA based object tracking algorithm, Postgraduate Thesis, (SVCE, India); 2008.

[9] Spartan-3E FPGA Starter Kit Board User Guide, UG230, vol. 1, No. 1, 20 June, 2008. <http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf> [Last modified in January 2013]

[10] Ebrahim Abd El-Hamid Mohamed Ramadan, "Embedded system based on a real time fuzzy motor speed controller"*Ain Shams Engineering Journal*,15 November 2013

[11] Miss. Vaishali Munde, Fuzzy Logic For Controlling Speed Of DC Motor, The *International Journal Of Engineering And Science (IJES)* ||Volume||2 ||Issue|| 5 ||Pages|| 33-39||2013|| ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805