

# Implementation of PID Controller PWM Module on FPGA

C. Sinduja<sup>1</sup>, Mrs. P. Thenmozhi<sup>2</sup>

<sup>1</sup>PG Student, Department of ECE, Gnanamani College of Technology, Tamil Nadu, India

<sup>2</sup>Assistant Professor, Department of ECE, Gnanamani College of Technology, Tamil Nadu, India

## ABSTRACT

A technique for designing an Intelligent PID controller using Very Large Scale Integrated circuits (VLSI) is presented. The PID controller parameters are optimized using the Particle Swarm Optimization (PSO) algorithm, which identifies errors and controls the system with multiple iterations of different parameters. The optimization process focuses on parameters such as gain, delay, lag, and various times to solve problems. To evaluate the system's performance, a new function that includes system adjusting time, rise time, overshoot, and system error is defined. The circuits are tested and implemented as intelligent PID controllers on VLSI devices in a laboratory plant. The algorithm applied to the PID controller reduces iterations and enables rapid control action, resulting in improved system time delay and performance. The optimization technique is applied to improve the system's response. Finally, the Intelligent PID controller is implemented in FPGA.

**KEYWORDS:** *Intelligent PID controller, VLSI, Particle Swarm Optimization (PSO), gain, delay, lag, system adjusting time, rise time, overshoot, system error, laboratory plant, FPGA*

## I. INTRODUCTION

There are two main approaches to implementing digital control systems: software-based and hardware-based.

Software-based systems involve the interaction between memory and processor, with tools such as PLCs, microcontrollers, microprocessors, DSPs, and computers. Hardware-based systems use VLSI devices, which are highly configurable ICs used to implement logic functions. They offer advantages such as ease of design, lower development costs, high speed capabilities, and power efficiency. VLSI devices can be programmed using software provided by manufacturers, with Hardware Description Languages like VHDL and Verilog used to implement complex algorithmic logic.

To address the challenge of identifying and changing parameters for PID controllers, an intelligent PID control method is proposed. This approach uses PSO to optimize controller parameters based on system error, addressing issues like lag, time-variety, and nonlinearity. An evaluation function is defined that considers system adjusting time, rise time, overshoot, and system error. The intelligent PID controller can

be applied to first-order and second-order systems with time delay and outperforms classic PID optimized by PSO. Proper computation of thresholds with low latency is crucial for the canny edge detection algorithm's performance. A robust threshold computation method with higher accuracy has been introduced, but it requires more resources and increases latency.

## II. DESIGN PID CONTROL PWM MODULE

The objective of this study is to utilize FPGA technology to develop a PWM module for PID control. FPGA-based implementation offers benefits such as parallel processing, low power consumption, and complex functionality at high speeds. The design of the PID control PWM module was carried out using Quartus II programmable logic design software and tested on the DE0 Nano Board from Altera's Cyclone IV FPGA family. Signal Tap II analyzer and RTL viewer were utilized for debugging and analysis, while Time Quest analyzer was used for proper timing constraint and clock arrangement. The simulation and hardware testing results demonstrate

**How to cite this paper:** C. Sinduja | Mrs. P. Thenmozhi "Implementation of PID Controller PWM Module on FPGA" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-5, October 2023, pp.499-503, URL: www.ijtsrd.com/papers/ijtsrd59966.pdf



Copyright © 2023 by author (s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



that utilizing FPGA has several advantages, including high speed, flexibility in design, and high reliability.

### III. PID CONTROLLER IN A FPGA

The aim of this study is to develop a microclimate monitoring system for precision agriculture using Zigbee technology. This system consists of sensor nodes that measure air temperature and humidity in a large area. The nodes are connected in a cluster tree topology, allowing for a long-distance communication range of up to 1 mile. Each node contains a microcontroller unit, air temperature and humidity sensors, and a solar cell for energy supply. An energy management scheme is implemented to optimize power use, and data is sent to a receiver every 8-30 minutes. The data is stored on a data-logging PC and displayed on a website for real-time monitoring. The use of Zigbee technology provides low cost, low power consumption, and flexible architecture for wireless networking of field sensors in precision agriculture.

### IV. EXISTING SYSTEM

The purpose of this demonstration is to showcase the implementation of a PID controller in the Main.cpp using C++ code on the Altera NIOS II Processor. The code utilizes the proportional and derivative modes of the PID controller to generate a new output value based on the error input. The output value is then used to control the LeftSpeed and Right-Speed of the two motors on the A-Cute Car. The  $k_p$  and  $k_d$  values are set to 1.0 and 8.0 respectively, with  $k_i$  set to 0.0.

The LTC2308 IP is used to read eight digitized values from the LTC2308 ADC chip through a high-speed SPI bus. These values include one for the input power voltage and seven sensor values from the sensor board containing seven Photo Interrupters used to track dark lines on a white background. The PWM IP controls the rotation speed and direction of the DC motor, with each motor controlled by a PWM controller. The 1K waveform IP generates a 1M frequency to drive the buzzer, and the associated GPIO controls the beep sounds on or off switch. The left and right lamps are directly controlled by the GPIO IP, while the IR receiver decodes received IR signals transmitted from the Terasic remote controller.

Overall, the PID controller is a widely used control loop feedback mechanism in industrial control systems that combines proportional, integral, and derivative modes to correct errors between measured process variables and desired set points.

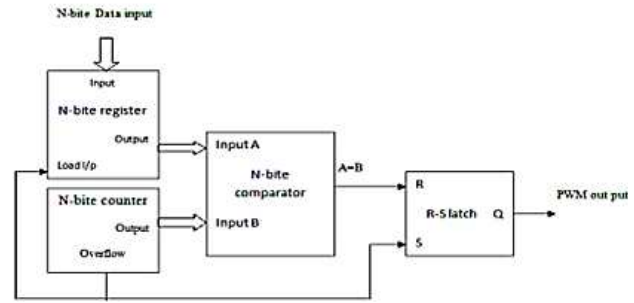


Figure I. Block Diagram Using Data Input And Output Pwm Module

### V. PROPOSED SYSTEM

To read eight digitized values from the LTC2308 ADC chip, the LTC2308 IP is used through a high-speed SPI bus. These values include one for the input power voltage and seven sensor values from the sensor board containing seven Photo Interrupters used to track dark lines on a white background. Control loops are used almost everywhere now a day. Anytime we adjust our current works according to the results obtained in previous results we form a control loop. For example, when we feel cold and turn our heater on we form a feedback loop, and when we press the accelerator of a car whenever we are getting late we again for a control loop. Whenever we change make any change in environment by sensing the previous results of that process we form a closed control loop in our mind. Changing the speed of the car is one of the best examples.

The PWM IP controls the rotation speed and direction of the DC motor, with each motor controlled by a PWM controller.

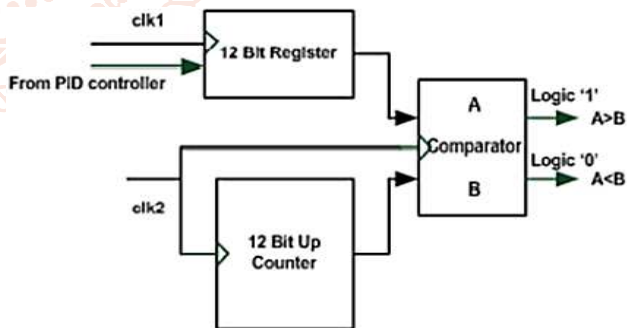


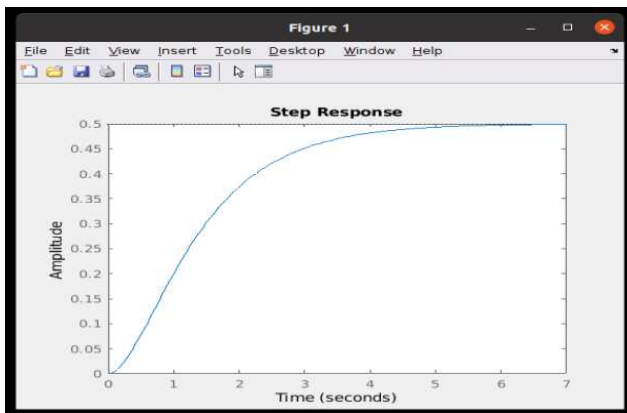
Figure II. Digital PWM Module

```

Editor - /home/pauljuma/PDsample.m
PDsample.m x +
12 - step(M)
13 - grid on
14
15
16 - Kp = 1;
17 - Ki = 0;
18 - Kd = 0;
19
20 - Gc = pid(Kp,Ki,Kd);
21
22 - Mc = feedback(Gc*Cp, H);
23 - step(Mc)
24
    
```

Figure III. PID Control of unmanned aerial vehicles (UAVs)

PID Control System Design and Automatic Tuning using MATLAB/Simulink introduces PID control system structures, sensitivity analysis, PID control design, implementation with constraints, disturbance observer-based PID control, gain scheduled PID control systems, cascade PID control systems, PID control design for complex systems, automatic tuning and applications of PID control to unmanned aerial vehicles. It also presents resonant control systems relevant to many engineering applications. The implementation of PID control and resonant control highlights how to deal with operational constraints. PID Control System Design and Automatic Tuning using MATLAB/Simulink is intended for undergraduate electrical, chemical, mechanical, and aerospace engineering students, and will greatly benefit postgraduate students, researchers, and industrial personnel who work with control systems and their applications.



**Figure IV. PID Control of unmanned aerial vehicles (UAVs)**

## VI. PWM MODULE

The purpose of this demonstration is to showcase the use of PWM as a technique for delivering power to a circuit in a precise and controlled manner. PWM involves modulating the duty cycle of a signal source to provide logic "1" and "0" for a controlled period of time. The block diagram of the PWM generator is shown above, with an N-bit data word as input corresponding to the actual PWM duty cycle.

The register stores the input (set value) and outputs it when the load input is high. This output is then compared with the N-bit counter value using a comparator. When these two values become equal, the comparator output is used to reset the R/S latch output. When the counter overflows, the load signal goes high and the latch output is set to high.

PWM is a widely used technique in various applications, including motor control, lighting control, and power electronics. Its ability to provide precise and controlled power delivery makes it an essential tool for many industries.

## VII. SOFTWARE IMPLEMENTATION

MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly

Dealing with Matrices and Arrays

2-D and 3-D Plotting and graphics

Linear Algebra

Algebraic Equations

Non-linear Functions

Statistics

Data Analysis

Calculus and Differential Equations

Numerical Calculations

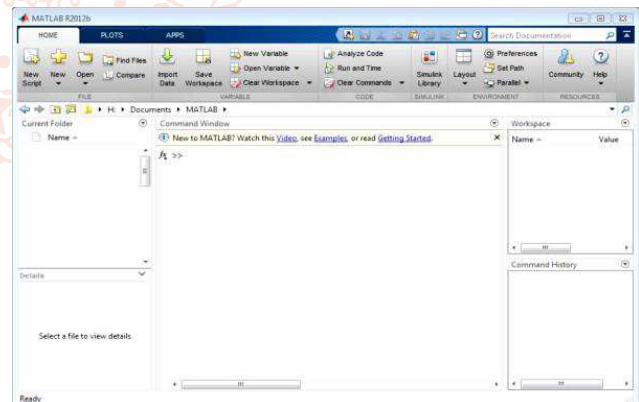
Integration

Transforms

Curve Fitting

Various other special functions

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.



**Figure V. Matlab Window**

## VIII. THE MATLAB ENVIRONMENT

The MATLAB environment (on most computer systems) consists of menus, buttons and a writing area similar to an ordinary word processor. There are plenty of help functions that you are encouraged to use. The writing area that you will see when you start MATLAB is called the command window. In this window you give the commands to MATLAB. For example, when you want to run a program you have written for MATLAB you start the program in the command window by typing its name at the prompt.



The command window is also useful if you just want to use MATLAB as a scientific calculator or as a graphing tool. If you write longer programs, you will find it more convenient to write the program code in a separate window, and then run it in the command window (discussed in Intro to programming).

manipulations, etc.). Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost.

### System 1

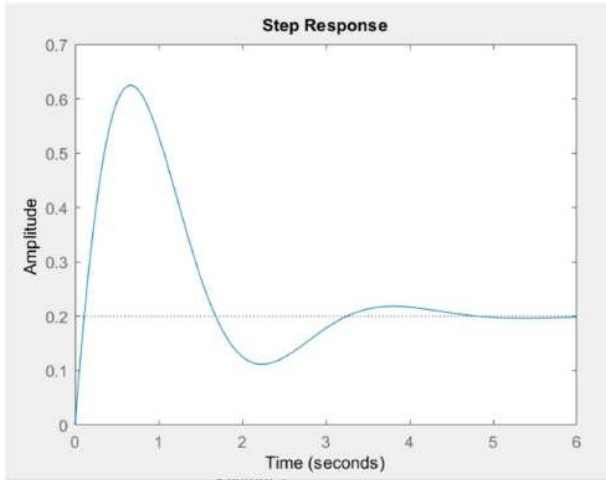


Figure VI. Phase Crossover Frequency

#### A. Variables

Variables are defined using the assignment operator, =. MATLAB is a weakly typed programming language. It is a weakly typed language because types are implicitly converted. It is a dynamically typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function.

#### B. Vectors/matrices

As suggested by its name (a contraction of "Matrix Laboratory"), MATLAB can create and manipulate arrays of 1 (vectors), 2 (matrices), or more dimensions. In the MATLAB vernacular, a vector refers to a one dimensional (1×N or N×1) matrix, commonly referred to as an array in other programming languages. A matrix generally refers to a 2- dimensional array, i.e. an m×n array where m and n are greater than 1. Arrays with more than two dimensions are referred to as multidimensional arrays. Arrays are a fundamental type and many standard functions natively support array operations allowing work on arrays without explicit loops.

#### IX. FPGA SIMULATION

MATLAB has structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field

Root Locus Output Plots :

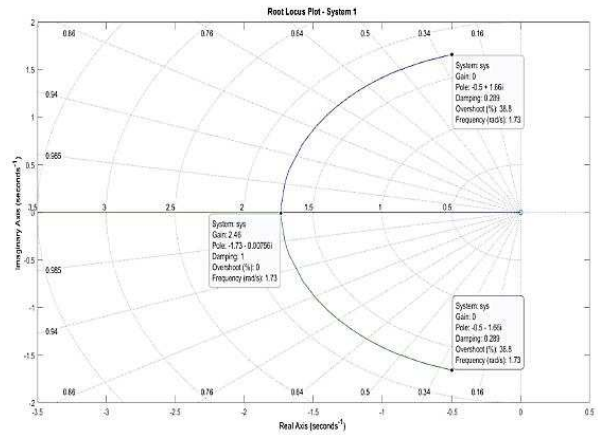


Figure VI. Root Locus Output

#### X. OUTPUT

%% Experiment: To plot the Bode plots of given Systems

clc; close all; clearvars;

%% Q7: Find the Gain 'K' for which Gain Cross over Frequency is '5 rad/sec'

%% Ans: Gain K = 0.056, as seen from Bode plot (Solution in Assignment)

% 0.056 s<sup>2</sup>

% ----- <== System 1

% 0.004 s<sup>2</sup> + 0.22 s + 1

Num1 = [0.056 0 0]; % Numerator Coefficients

Den1 = [0.2 1]; %Denominator factor 1

Coefficients

Den2 = [0.02 1]; %Denominator factor 2

Coefficients

Tofs1 = tf(Num1,conv(Den1,Den2));

%Creation of System

figure(1)

margin(Tofs1) %Plots the Bode Plot and also gives the "Gain Crossover Frequency"

%% Q8: Find the Phase Crossover Frequency

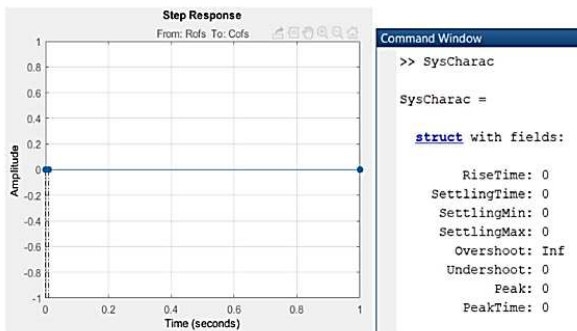
% Ans = 9.976 (Theoretically), Verified from bode plot

% 1000

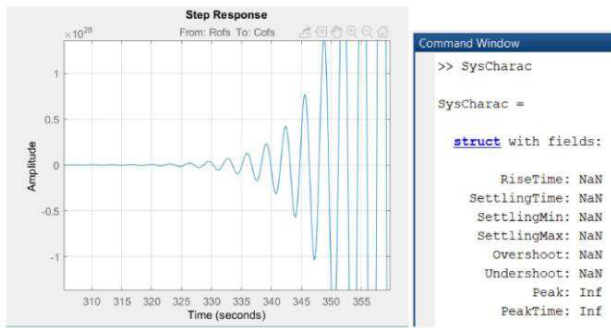
% ----- <== System 2

```
% s^3 + 24.95 s^2 + 99.53 s + 999
Num2 = 1000; %Numerator Coefficients
Den3 = [1 22.5]; %Denominator factor 1 Coefficients
Den4 = [1 2.45 44.4]; %Denominator factor 2
Coefficients
Tofs2 = tf(Num2,conv(Den3,Den4));
%Creation of System
figure(2)
margin(Tofs2) %Plots the Bode Plot and also gives
the "Phase Crossover Frequency"
```

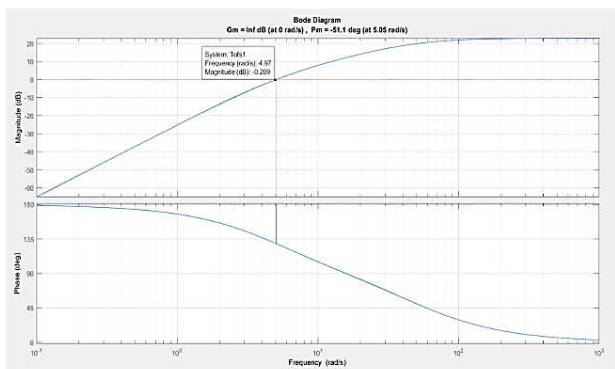
**System O.1**



**Figure VII. PID Controller Response**



**Figure VIII. PID Controller Response**



**Figure IX. Bode Diagram**

## XI. CONCLUSION

The proposed system utilizes a FPGA-based design approach to create a temperature control system. This approach streamlines the design process for embedded control designers, reducing development time for efficient embedded software. The system allows for software modeling and simulation in an environment such as Matlab/Simulink, hardware implementation, and post synthesis simulation of the entire system. Future work includes implementing a multiplierless digital PID controller using distributed arithmetic architecture, offering high processing speed, reduced power consumption, and hardware compatibility for FPGA implementation.

## XII. REFERENCE

- [1] Wei Zhao, ByungHwa Kim, Amy C. Larson, and Richard M. Voyles, "FPGA Implementation of Closed loop control System for Small Scale Robot", In Proceedings., 12th International conference on advanced robotics-ICAR05, 2005, pp.70-77.
- [2] Daijin Kim. "An implementation of Fuzzy Logic Controller on the Reconfigurable FPGA system", IEEE transactions on Industrial Electronics, Vol.47, No.3, 2000.
- [3] Y. F. Chang, M. Moallm and W. Way, "Efficient PID Controller Using FPGA", IEEE transactions on Industrial Electronics, Vol.35, pp. 119-125.
- [4] L. Samet, N. Masmoudi, M. W. Kharrat and L. Kamoun, "A Digital PID Controller for Real time and Multi Loop Control: a comparative study", in proceedings of 1998 IEEE international Conference on Electronics Circuits and Systems, Vol.1, 1998, pp. 291-296.
- [5] Paul Leisher, Christopher Meyers. Dr. T. Stewart and Dr. G. Dempsey, "FPGA Implementation of a PID controller with DC Motor Application", Bradley University, a project report, 2002.
- [6] Stephen Brown, "Digital logic design", Tata Mechgraw Hill publications, India, 2002.
- [7] Rajesh Nema, Rajeev Thakur, and Ruchi Gupta, — Design & Implementation of FPGA Based on PID Controller, International Journal of Inventive Engineering and Sciences (IJIES), Volume-1, Issue-2, pp 14-16, January 2013.