Implementation PID Controller in Time Domain and Z Domain on FPGA

Saman Kaedi\textsuperscript{1,2}
Yousef Seifi Kavian\textsuperscript{1,3}
\textsuperscript{1}Department of Electrical and Computer Engineering Chamran University Ahwaz, Iran
\textsuperscript{2}kaedi.saman@gmail.com
\textsuperscript{3}y.s.kavian@scu.ac.ir

Abstract--In this paper a PID controller is designed and it’s implemented on Field Programmable Gate Array (FPGA) by two method: in first method PID controller is designed in time domain and in second method PID is designed in frequency domain (Z domain) and final two method compare together. The filter that designed in time domain is more accurate than other method.

Keywords: PID controller, FPGA, Z domain, time domain.

I. Introduction

The proportional–integral–derivative (PID) controller is one of the most common types of feedback controllers that are used in dynamic systems. This controller has been widely used in many different areas, such as aerospace, process control, manufacturing, robotics, automation, and transportation systems \cite{1,2}.

Implementation of PID controllers has gone through several stages of evolution, from early mechanical and pneumatic designs to microprocessor based systems. Field programmable gate arrays (FPGA) have become an alternative solution for the realization of digital control systems, which were previously dominated by general purpose microprocessor systems \cite{1,4}.

PID controllers have been widely used over the past five decades due to their simplicity, robustness, effectiveness and applicability for a broad class of systems FPGA-based controllers offer advantages such as high speed, complex functionality, and low power consumption \cite{5,6}.

II. Implementation

We should make three operations (proportional, integral and differential) separately for Implementation PID controller and sum these operations with each other till output reach.

The FPGA get three coefficients of PID (K_p, K_i, K_d) as input that everyone has 12 bit.

A. First method: PID in Time Domain

A.1 Proportional controller:

This controller multiplies input in the factor of proportional control.

\[ \text{Out}_p = k \times Y \] (1)

A.2 Differential controller

For differentiate we consider two kind of input with different time period and differentiate at that time \cite{7}.

\[ \text{Out}_\text{diff} = T_d \times \frac{dy}{dt} \] (2)
\[ \text{Out}_\text{diff} = T_d \times \frac{y_2 - y_1}{dt} \] (3)

And \( dt \) is the time between two samples of input. It’s half of period of clock. For chronological system, clock input is used, that for sampling input occurs in
rising edge and the other sample in falling edge of clock.

Here because the frequency of clock is 2 KHz \( dt = \frac{1}{1024} \) so instead of divide by 1024 can be 10 times shift to left.

A.3 Integral controller

For the integrating in each period of time should calculate the area under the curve and sum with the previous value of the integral.

For calculating the area of each period of time we consider two sample of input with specify time distance and the average between two values considers to height of rectangle.

For calculating this area, height multiplies by period.

\[
\text{Out}_{\text{int}} = \frac{1}{T_i} \int_0^t y \, dt
\]  

\[\text{(4)}\]

\[
\text{Out}_{\text{int}} = \frac{1}{T_i} \int_0^t y \, dt
\]  

\[\text{(5)}\]

\[
delta = \frac{1}{T_i} \times \frac{2(b-a)}{2} \times dt
\]  

\[\text{(6)}\]

\[
\text{Out}_{\text{int}} = \frac{1}{T_i} \times (\text{Out}_{\text{int}} + \delta)
\]  

\[\text{(7)}\]

Instead of multiply by \( \frac{1}{1024} \) can be divided by 1024 or shift it 10 times to right. For division (divide by \( T_i \)) an algorithm of divide is used.

Divide operation is written as a function input the process.

\[
\frac{a}{b} = \text{Divid} (a, b)
\]  

\[\text{(8)}\]

The algorithm is as follows within a loop each time b subtraction value a till remain amount less than b. number of repeating loop is value of divide.

For i in 0 to 4096 loop

If a<b then exit;

Else

E=e-b;

End if;

Return I;

End loop;

B. Second method: PID IN Z DOMAIn

In this method PID controller transfer function is written in form discrete (Z domain) by bilinear transform and calculated output function \[\text{[13]}.\]

\[
D(s) = K_p + \frac{1}{T_i} + T_d s
\]  

\[\text{(9)}\]

\[
D(z) = K_p + \frac{T}{2T_i} \times \frac{Z+1}{Z-1} + \frac{T_d}{T} \times \frac{Z^{-1}}{Z}
\]  

\[\text{(10)}\]

\[
D(z) = \frac{\text{out}(z)}{\text{e}(z)} = \frac{a_0 + a_1 Z^{-1} + a_2 Z^{-2}}{1-Z^{-1}}
\]  

\[\text{(11)}\]

\[
a_0 = K_p + \frac{T}{2T_i} + \frac{T_d}{T}
\]  

\[\text{(12)}\]

\[
a_1 = -K_p + \frac{T}{2T_i} - \frac{T_d}{T}
\]  

\[\text{(13)}\]

\[
\text{out}(z) - Z^{-1} \text{out}(z) = a_0 e(z) + a_1 Z^{-1} e(z) + a_2 Z^{-2} e(z)
\]  

\[\text{(14)}\]

\[
\text{out}(n) = \text{out}(n-1) + a_0 e(n) + a_1 e(n-1) + a_2 e(n-2)
\]  

\[\text{(15)}\]

Result of simulation for both methods is brought here.

III. Hardware

First and second method has been implemented on FPGA Spartan3. In first stage, three coefficients of PID \((K_p, K_i, K_d)\) has been gotten to FPGA as 12-bit digital input signals. On other hand, PID output has been defined as 24-bit digital output.

The circuit has 12-bit input which is the controller input. It has three 12-bit inputs which distinguish factors of \(k, T_i, T_d\).

Also an clock input consider for circuit .This input should be connect to a wave with 2 KHz frequency to control the chronological of circuit.

The inputs and factor of controller which is analog in operative systems come in controller from 12 bits A/D converter that provides input data in form of digital for FPGA \[\text{[19]}\].
Also in output for convert digital data to analog ones use a D/A converter which provides output for circuit.

This controller has four 24-bit outputs:

- Proportional output
- Differential output
- Integral output
- Proportional differential output

IV. Simulation

For simulating the PID controller inputs which is simulated, consider to input of function and enter 40 point as input in test bench. After this compare calculated outputs with output from Matlab controller which equal with good are accuracy.

Adjustment the factors of PID controller occurs in Ziegler-Nichols. And for \( \frac{1}{s^2 + s + 5} \) factors are calculated:

\[
T_d = 15, T_i = 2, K = 1
\] (16)

In first stage this coefficients has been put in Matlab PID controller and controlled system in fig.1. In fig.2 error signal of system (error=-input-output) that is PID input signal has been shown. In fig.3 PID output signal in Matlab, has been shown. In fig.6.a FPGA PID input has been shown. Fig.4,5 shows input and output signal of PID in FPGA implementation in ISIM simulator in first and second method. Finally in fig.6.b PID output of Matlab simulator and FPGA implementation in first and second method has been compared.

![Figure1. Schematic of test PID controller in Matlab](image-url)
5. Conclusion

In this paper presented a PID controller designed by two methods and implemented on FPGA. Outputs of every method compare with MATLAB output of PID controller that controls a sample function in close loop.

The PID that designed in time domain is more accurate than other method. Because in frequency method we use an approximate for transfer from continues domain to discrete domain (Z domain).
Figure 4. Result designed PID controller by first method in xilinx.

Figure 5. Result designed PID controller by second method in xilinx.
Figure 6. (a) input PID controller and (b) output PID controller by method 1, 2 and matlab output.

References


[4] Suying Yang, Miaomiao Gao, Jianying Lin, and Zhuohan Li “The IP Core Design of PID Controller Based on SOPC”


