Design of PID Controller Based on PSO Algorithm and Its FPGA Synthesization

Burhan Aslam Arain*, Muhammad Farrukh Shaikh, Bharat Lal Harijan, Tayab Din Memon, Imtiaz Hussain Kalwar

Abstract: ***A Proportional-Integral-Derivative (PID) controller make its appearance in various control mechanism due to its adaptively, applicability and simple structure. The tuning for parameters K_P, K_D and K_I selection for PID is a tedious task. A Particle-Swarm-Optimization (PSO) algorithm is an evolutionary method that simulates the particles to provide best solutions in a given search-space based on fitness value. It provides another design of optimization for PID controller that provides better gain parameters, fast convergence and quick computation, in this paper, an efficient designed PSO based PID controller is then synthesized with the help of Xilinx SYSGEN. To evaluate the effectiveness and usefulness of PSO the DC motor based system response is figured and compared it with conventional method.***

Keywords: PSO algorithm, PID controller, FPGA synthetization, PID optimization, PSO-PID controller

I. INTRODUCTION

In industrial applications the control of motor's speed, robot's position, and conveyor drive traction system is done using voltage control techniques. The control of these machines can be achieved by Mechanical, Electrical and Electronic techniques. In earlier the control is achieved using mechanical techniques only [1]. Later the electrical and electronic methods are introduced to overcome the control problems. The PID is one of the best techniques used widely in all control application in the industries for machines. Today numbers of controllers still used PID for control action [2]. In PID type controller the parameters tuning is a very difficult task, different approaches of tuning have been proposed over the past years. Some are based on system response of the plant and some techniques require a model with the response of plant. The Zeigler-Nichols and Cohen-Coon methods for tuning are very popular among other traditional methods.

The Zeigler-Nichols is a good method for finding the gain of PID but it is not performed well in large overshoots and is a constraint with the type of system [3]. Where the modern soft computing optimization techniques for tuning of PID controller suggested to overcome the problems faced by traditional methods. These tuning methods do not require any system model information, these methods enhance the performance of PID parameters. A Neural Networks (NN), Ant-Bee colony method (ABC), Genetic algorithm (GE), Bacterial foraging optimization (BFO) and Particle swarm optimization (PSO) are popular optimization techniques. A PSO is simple and easiest optimization than other similar techniques, it is a method taken from nature of bio-inspired techniques. It provides an optimized solution in given search-space by assigning various particles in swarm which allotted a population-motion. In a swarm, the motion of each particle is updated with reference to its own motion practice and experience of another particle. It provides a flexible mechanism to improve the local and global explorations [4]. The implementation of PID controller by digital methods has complex task such as speed and power consumption but the FPGA based realization provides low power consumption, less cost, flexible design and less area utilization and fastest performance than other digital approaches [5].

In this work, we have tried to represent the PID optimization by PSO method in Matlab. For optimization the PSO coefficients are initialized that improves the system response through PID, the DC motor model is used to test the output response of PSO tuned PID. The FPGA based controller realization is done in Xilinx System Generator and the synthetization is performed by Xilinx ISE. The comparative outcomes provide the analysis between the proposed method and conventional method.

II. PSO PID SYSTEM DESIGN

A. Plant Model

For the simulation of PID optimization, it is necessary to describe plant model, hence the separately excited DC motor in speed control mode is used. In this category of the motor the armature and field windings are independent of each other means that the the field current is not affected by the armature. Therefore the mathematics of such type motor is defined by given transfer function [6].

$$\frac{w(s)}{v(s)} = \frac{K_T}{(R_A + L_A S)(JS + B) + K_B K_T} \quad (1)$$

Where the R_A = Armature resistance, L_A=Armature inductance, J=Motor inertia, B=Friction constant, K_T=Torque constant, K_B=Back emf constant, *v(s)*=Input voltage and *w(s)*=Motor speed output. The value for this parameter of DC motor is given in Table 1.

Revised Manuscript Received on 28 December 2018.

Burhan Aslam Arain*, Department of Electronic Engineering, Mehran UET, Jamshoro, Pakistan.
Muhammad Farrukh Shaikh, Department of Electrical Engineering, Isra University, Hyderabad, Pakistan.
Bharat Lal Harijan, China Huadian Power Plant Operation Co. Ltd., EPTL Islamkot Pakistan.
Tayab Din Memon, Department of Electronic Engineering, Mehran UET, Jamshoro Pakistan.
Imtiaz Hussain Kalwar, Department of Electrical Engineering, DHA Suffa University, Karachi, Pakistan.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 1. Parameters of Dc Motor

Parameters	Value
Armature Resistance, R_A	11.2ohm
Armature Inductance, L_A	121.5Mh
Motor Inertia, J	0.0221 Kgm^2
Friction Constant, B	0.002953Nm/rad/sec
Torque Constant, K_T	1.28Nm/A
Back EMF Constant, K_B	1.28V/rad/sec

B. PID Controller

PID (Proportional-Integral-Derivative) controller is the most famous type of control widely use in engineering applications. The others controller has never important like PID controller but it should be tuned properly. The continuous type PID controller as shown in Figure 1 and the transfer function for the model is given as:

$$U(S) = Kp.E(S) + Ki\,\frac{E(S)}{S} + Kd.S^2E(S) \tag{2}$$

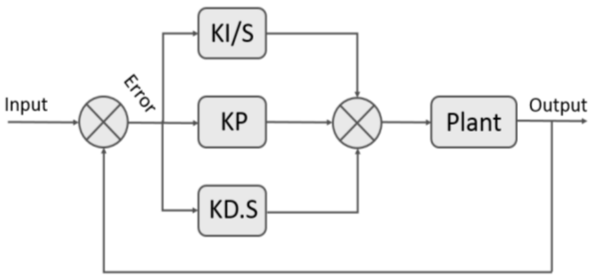

Figure 1. PID Controller Structure

Where U(S) is represented controller output, E(S) is the error signal in S domain, K_P represents proportional gain, K_I denote integral action that improves steady state and K_D shows derivative that enhance the performance of the system in the transient state. First Ziegler-Nichols method is used to tune the PID parameters but this method provide overshoots in transient response, give a satisfactory response in steady state, however the initial values for this conventional method is taken from [6]. To overcome this problem the PID controller is optimized by the PSO method that minimize error to improve plant output.

C. PSO Algorithm:

PSO Algorithm: the PSO algorithm is based on stochastic techniques such as ant colony method and genetic algorithm. The concept of PSO is similar to the social behavior of birds in swarm and flocking of fish is develop by Kennedy and Hebert in 1995. PSO is an iterative method uses particles to find optimum solution in specific search space. All the particles has certain value called positions in PSO that is compared to each other and evaluate against the value of fitness function. The particle which has best fitness value directed other particles to update it position. There are two fitness value are required in the update process of particles the personal best value called pbest value and global best value called gbest value. The pbest value is trace in every iterations of each particles and gbest is computed among the best solution in pbest value [7][14]. After acquiring the best two values the particles start to upgrade its velocity and positions by following equations.

$$v_{n\times d}^{m+1} = w.v_{nd}^{m} + C_1.rand().(pbest - x_{n\times d}) + C_2.rand().(gbest - x_{n\times d}) \tag{3}$$

$$x_{n\times d}^{m+1} = x_{n\times d}^{m} + v_{n\times d}^{m+1} \tag{4}$$

Where the $v_{n\times d}^{m+1}$ is updated velocity of n particle of m+1 iteration, v_{nd}^{m} the n particle velocity of m iteration, rand () is initially distributed random values (0-1), C1 and C2 represent the acceleration constant, w denoted the inertial weight factor. The inertial weight factor is used to maintain the particles last velocity while the accelerations factors control the flow of particles towards optimum solutions. The $x_{n\times d}^{m+1}$ is new update position of n particle in m+1 iteration, $x_{n\times d}^{m}$ last n particle position of m iteration. The particles position is mainly depend upon update velocity whereas the update velocity is totally control by acceleration factors and assigned inertial weight.

D. PSO-PID Controller:

The PSO algorithm is utilize to tune the gain parameters of PID. The adaption of new values of gain parameters is depend on fitness function value as shown in Figure 2.

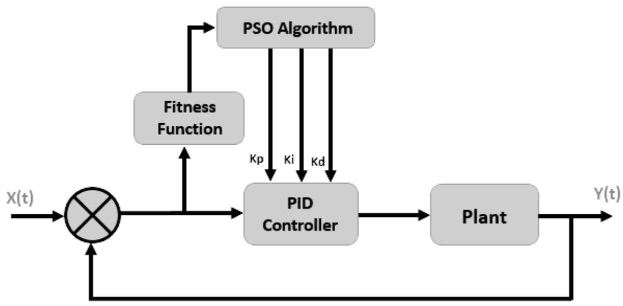

Figure 2. Design Structure of PSO PID Controller

For PID controller the 3-d (dimensional) search space is used that represents gains K_P, K_I and K_D, the cognitive and social factor empirically is equal to 4. The weight value is set between 0 and 1, the value between 0.4-0.6 is consider best value. The particle size and iterations is decided according to requirement. The fitness function to calculate the performance of plant in each iteration for all particles, the best fitness value is chosen among all the calculated values of fitness. The following listed steps are involve for optimization of PID controller.

1st Step: The 1st step specifies the PID controller model and its parameters that are needed in tuning process, the proportional gain K_P, integral gain K_I and derivative gain K_D. After parameters definition the PID is connected with plant in a closed loop.

2nd Step: Initialize size of particles N = 10, Dimension of search-space D = 3 (for KP, KI and KD gains) and iteration length M=1-20 is considered. Here different iterations size is chosen to shows the effect of iterations on outcomes.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3rd Step: Assign the weight = o.4, both accelerations factor C1 and C2 = 2, positions and velocity to particles at initial stage by random numbers in a two vectors one for position and other for velocity. Then the initial fitness of each particle pbest is computed the values stored in fitness vector.
4th Step: From the pbest fitness vector the minimum fitness value gbest is determined and stored in variable.
5th Step: After calculating pbest and gbest, each particles update its velocity and positions based on fitness value, weight constant and acceleration co-efficient. The new updated values are stored in already defined positions and vectors for particles in 3rd step.
6th Step: Calculate the values of fitness pbest and gbest for new updated position if these fitness values are improved the replace last values with these new values.
7th Step: Repeat the steps 4th, 5th and 6th till the last iteration or if any conditions for convergence is satisfied.

III. FPGA SYNTHETIZATION OF PID:

The implemented design on FPGA as per requirement can be altered at any stage, from Xilinx System Generator XSG a DSP based systems can be captured and designed in Simulink environment easily. At first a continuous PID controller defined in equation 2 for implementation in FPGA need to transform in digital form as a DSP system structure with some manipulations [8]. The digital form of continuous PID is given by equation 5 and 6.

$$U(z) = \left(K_P + \frac{K_I Ts(1+z^{-1})}{2(1-z^{-1})} + \frac{K_D(1-z^{-1})}{Ts}\right)E(z) \quad (5)$$

$$U(z) = (K_P(1-z^{-1}) + \frac{K_I}{2}Ts(1+z^{-1}) + \frac{K_D(1-z^{-1})^2}{Ts})E(z) \quad (6)$$

Where U(z)=digitized output of PID, TS=Sampling time, Z-1=Unit delay, with Xilinx SYSGEN the synthesizable Verilog and VHDL file easily generated and then from generated VHDL file the PID controller is synthesize in Xilinx ISE using Spartan 3E FPGA board. The Figure 3-4 shows the Xilinx SYSGEN model of optimized PID.
Where U(z)=digitized output of PID, TS=Sampling time, Z-1=Unit delay, with Xilinx SYSGEN the synthesizable Verilog and VHDL file easily generated. The Figure 3-4 shows the Xilinx SYSGEN model of optimized PID. The gateway in and gateway out blocks must be connected that interface the Xilinx SYSGEN model with other models in Simulink and converting the data of other type into FPGA data type. The input and output signals is provided and taken from these blocks.

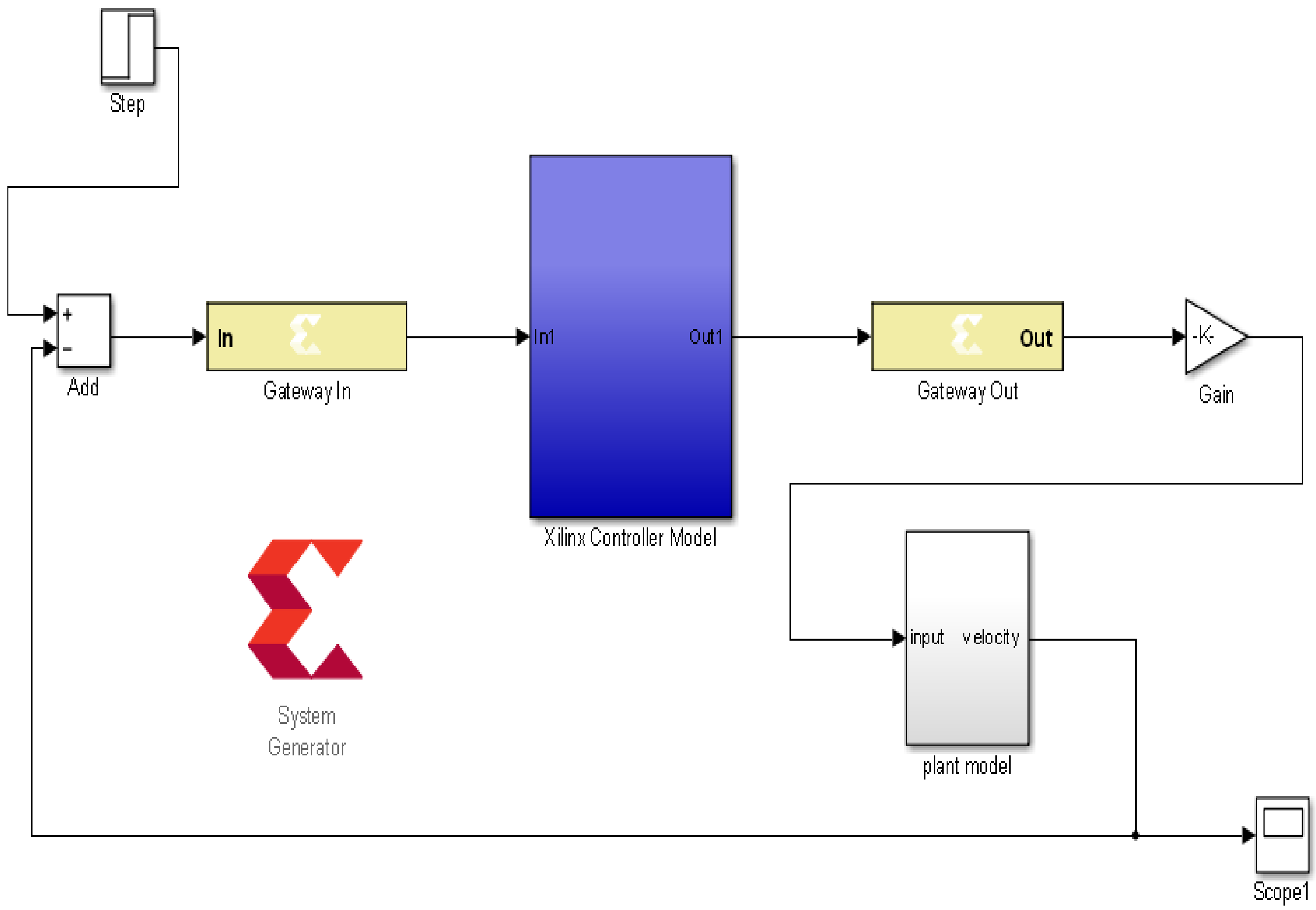

Figure 3. Simulink Model of XILINX SYSGEN PID with Plant

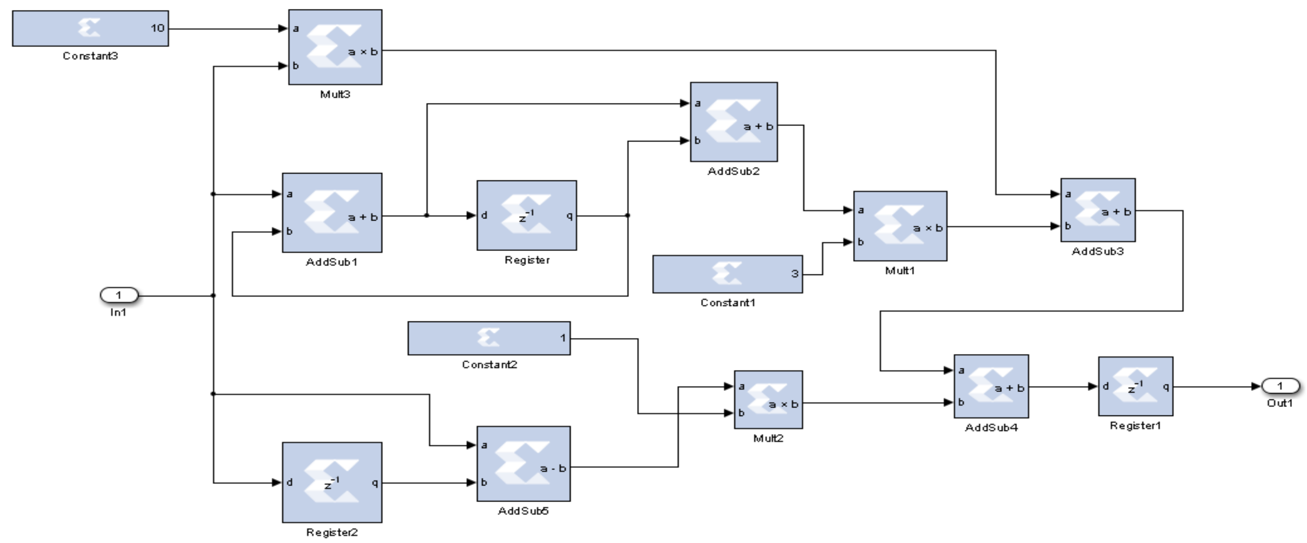

Figure 4. Internal Structure of XILINX SYSGEN PID Controller

IV. RESULTS AND DISCUSSIONS

The simulation results are performed in MATLAB with DC motor transfer function by taking parameters mention in Table 1 for optimization and FPGA synthetization is computed in Xilinx ISE. The step response for both methods are illustrated and results are compared in terms of overshoots, settling time and steady error. The Figure 5 shows the response of conventional method having big overshoots 78%, 7.18s settling time and provides satisfied steady state. The Figure 6-8 is shows PSO optimization with variable iterations size n=1-20 it is be noted from the response that at first optimization the PID provide few overshoots in transient state and more settling time. As the iterations increases the optimization almost eliminate the overshoots, give good settling and rise time as given in Table 2.

Table 2. Optimized Values of PID Controller

Parameters	Conventional Method	PSO n=1	PSO, n=10	PSO n=20
Overshoot, OV	78%	19%	8.3%	1.5%
Rise Time, R_T	0.123s	1.36s	1.43s	0.45s
Settling Time, S_T	7.18s	7.84s	11.8s	0.675s
Proportional Gain, K_P	15	1.31	1.77	5.18
Integral Gain, K_I	105	0.55	0.31	0.19
Derivative, K_D	0.54	0.28	0.56	0.4

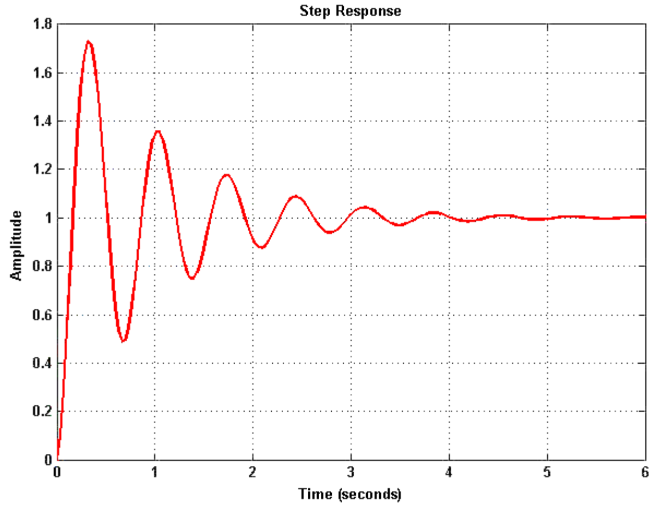

Figure 5. Conventional Method Tuned PID Controller Response

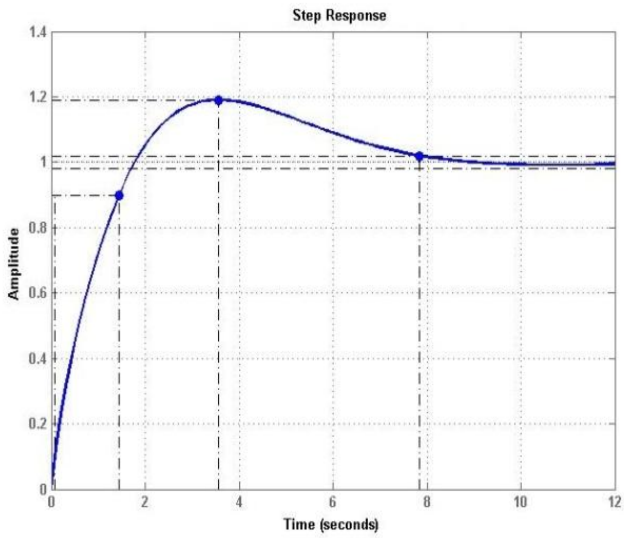

Figure 6. PSO Optimized PID Response when N=1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

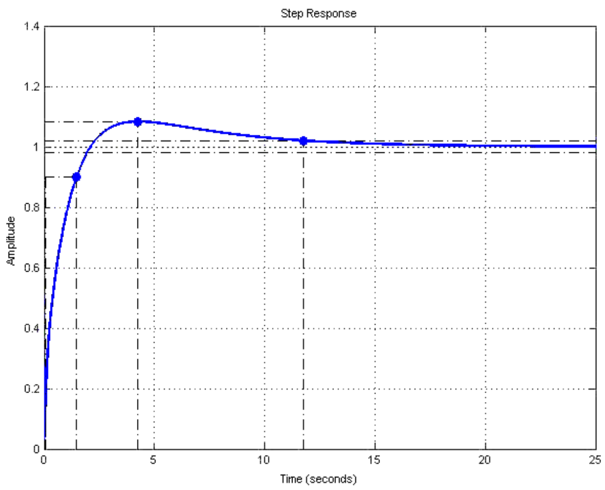

Figure 7. PSO Optimized PID Response when N=10

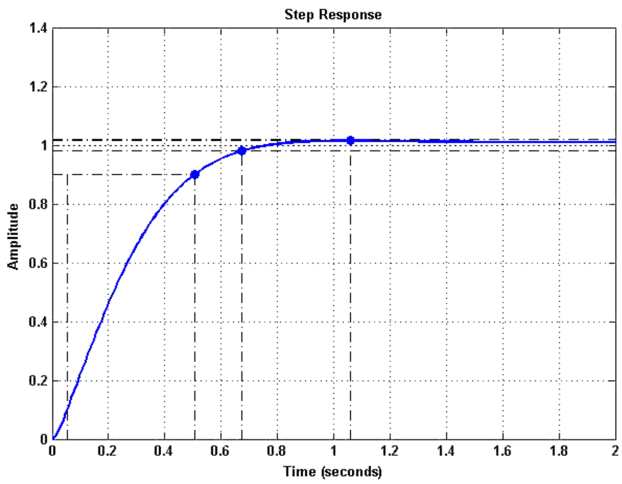

Figure 8. PSO Optimized PID Response when N=20

After this optimization the Xilinx SYSGEN model is simulate for synthetization in Xilinx ISE using VHDL generated file, the synthesized model is shown in Figure 9-10. There is no additional resources is required for realization of Xilinx SYSGEN model through Xilinx ISE for FPGA but it need some design manipulation for final hardware implementation.

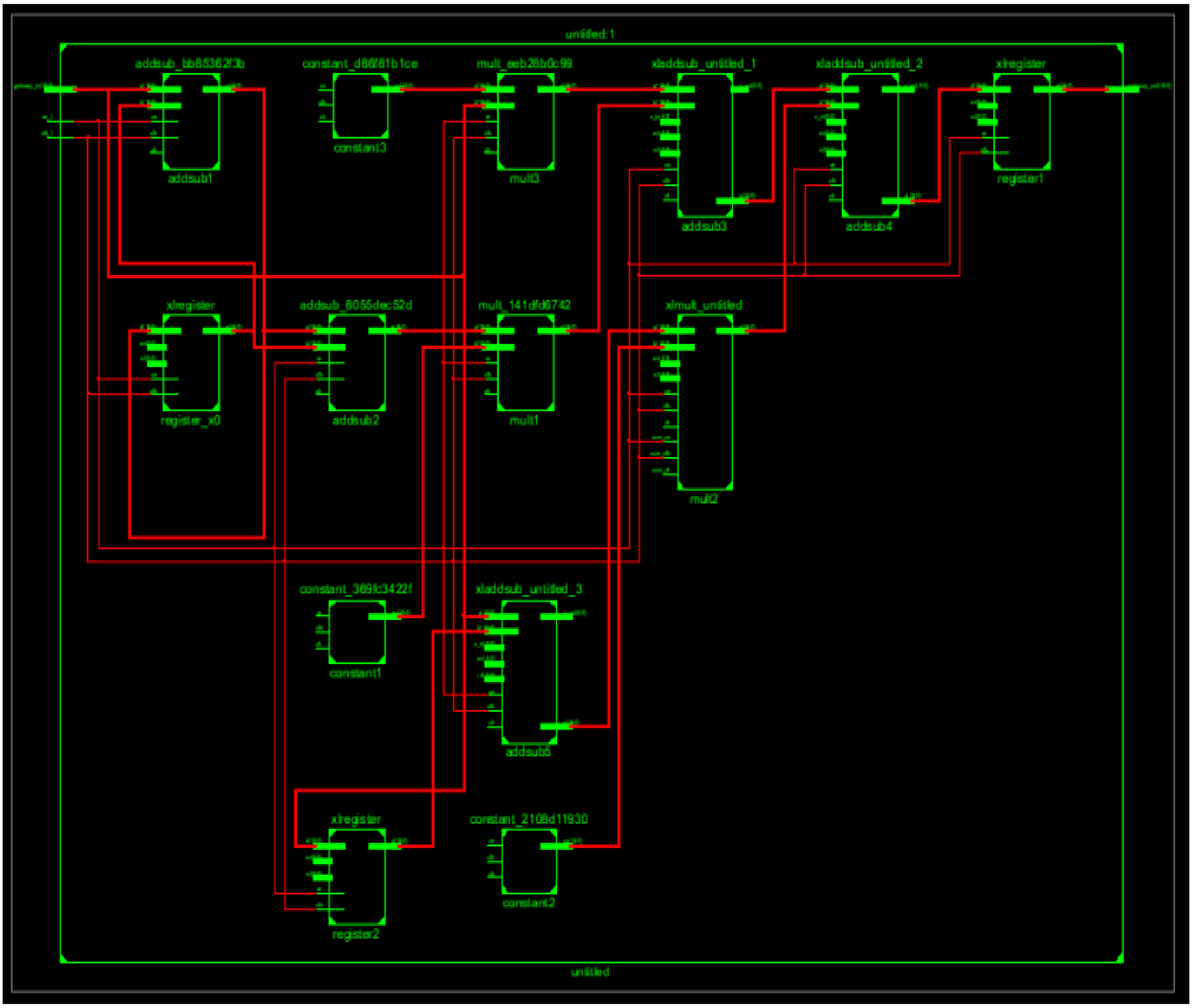

Figure 9. RTL View of Synthesized PID Controller.

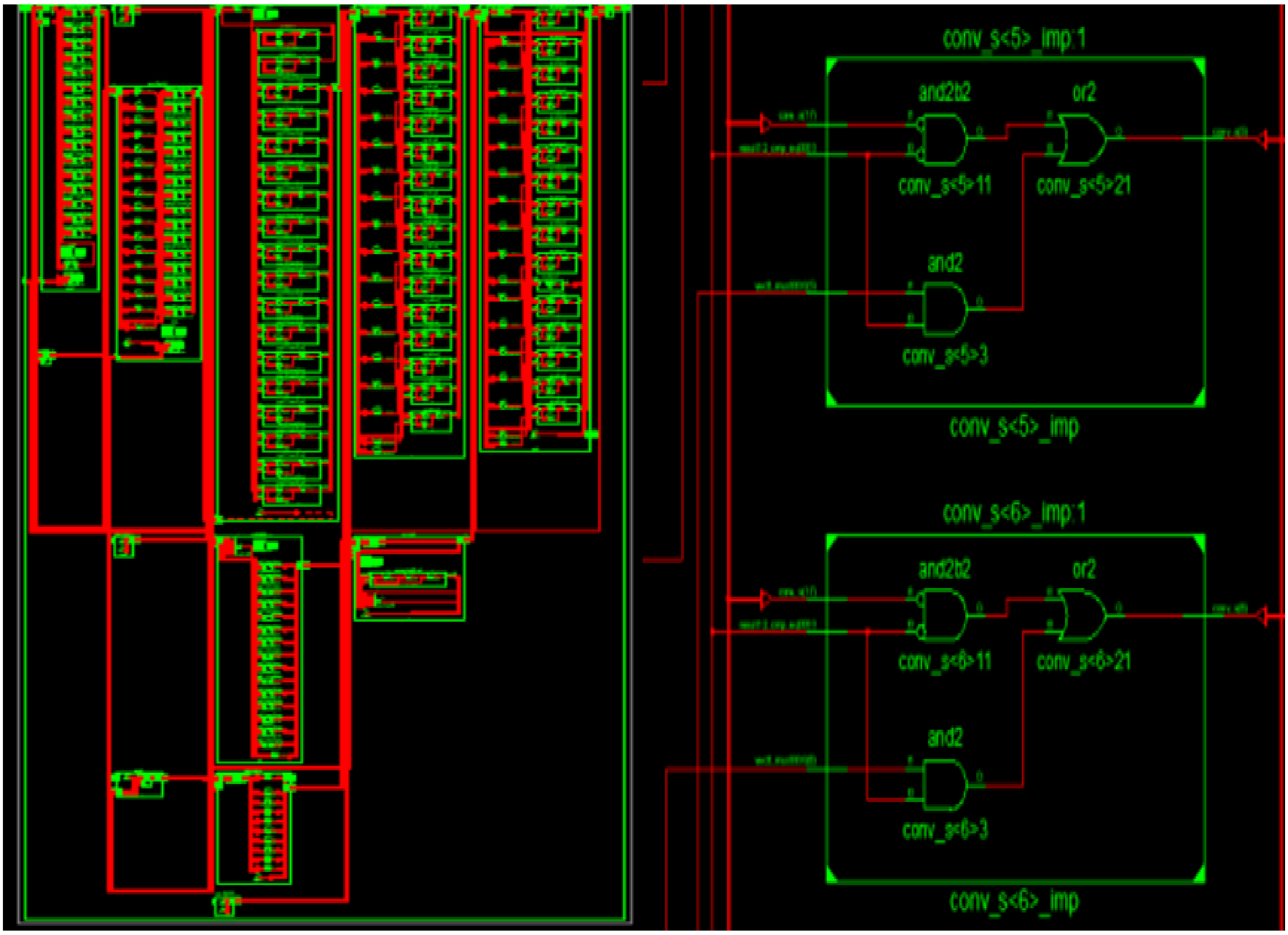

Figure 10. Internal Structure of RTL View

V. CONCLUSION

In this paper the digital PID controller is designed using Xilinx System Generator tuned via PSO algorithm is presented. In addition, optimization is compared with conventional method. The results are evident about the conventional method tuned PID controller is not performed good having large setling time and big overshoots. The PSO tuned PID controller provides satisfactory output by optimizing the gain parameters K_P, K_I and K_D, to minimize the error difference between desire and actual response. It is also noted that PSO optimization depends upon the number of iteration, particles size and other constant in algorithm. Gretaer number of iterations provide more accurate results however, it slows the process. Optimization of the tuned PID controller is realized with the help of Xilinx System Generator XSG using DSP design techniques in FPGA. The SYSGEN directly transforms the Simlulink model to VHDL file and is deployed on FPGA chip. It can be conluded that the XSG is an efficient tool for FPGA design realization from DSP techniques.

REFERENCES

1. G. Rajeshkanna, "Modern Speed Control of Separately Excited Dc Motor by Boost Converter Fed Field Control Method," In International Conference on Computer Communication and Informatics (ICCCI -2013), Coimbatore, India, 2013.
2. S. Vijay and V.K. Garge, "A Comparative Study On Speed Control Of D.C. Motor Using Intelligence Techniques," International Journal Of Electronic And Electrical Engineering, Vol. Volume 7, No. Number 4, Pp. 431-436, 2014.
3. T. Yucelen, O. Kaymakci, S. Kurtulan, "Self-Tuning PID Controller Using Ziegler-Nichols Method For Programmable Logic Controllers," In IFAC Proceeding Volumes, 2006.
4. N. Kundariya And J. Ohri, "Design Of Intelligent PID Controller Using Particle Swarm," International Journal Of Scientific & Engineering Research, Vol. 4, No. 7, July 2013.
5. S. Banerjee and A. Sinha, "Performance Analysis of Different DSP Algorithms on Advanced Microcontroller and FPGA," In IEEE International Conference, ACTEA, 2009.
6. S. K. Guman, V. K. Giri, "Speed Control Of Dc Motor Using Optimization Techniques Based PID Controller," In 2nd IEEE International Conference On Engineering And Technology (ICETECH), Coimbatore India, March 2016.
7. A. H. Al-Mter and S. Lu, "A Particle Swarm Optimization Algorithm Based On Uniform Design," International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol. 6, No. 2, March 2016.
8. Geramipour, Arezou, M. Khazaei, A. Marjaninejad, and M. Khazaei., "Design Of FPGA-Based Digital PID Controller Using Xilinx SYSGEN For Regulating Blood Glucose Level Of Type-I Diabetic Patients," International Journal Of Mechatronics, Electrical And Computer Technology, Vol. 3, No. 7, Pp. 56-69, April 2013.
9. B. L. Harijan, M. F. Shaikh, B. A. Arain, T. D. Memon And I. H. Kalwar, "FPGA Based Synthesize Of PSO Algorithm And Its Area-Performance Analysis," International Journal Of Advanced Computer Science And Applications (IJACSA), Vol. 9, No. 6, 2018.
10. J. Ciganek, M. Kocur And S. Kozak, "Hardware Realization Of Advanced Controller Design Methods Using FPGA," IFAC-Papers Online, Volume, Vol. 49, No. 5, Pp. 163-168, 2016.

11. R. Eberhart and J. Kennedy, "Particle Swarm Optimization," Proceedings of the IEEE Int. Conf. on Neural Networks, Vol. 4, P. 1942–1948, 1995.
12. H. Tim, "Fitness Estimation and the Particle Swarm Optimization Algorithm," In IEEE Congress on Evolutionary Computation, 2007.
13. V. Gupta, K. Khare And R. P. Sigh, "Efficient FPGA Implementation Of 2nd Order Digital Controllers Using Matlab/Simulink", ARPN Journal of Engineering And Applied Sciences, Vol. 6, No. 8, August 2011
14. X. Hu, "Http://Www.Swarmintelligence.Org.
15. B. Jena and S. Kumar, "Tuning Of PID Controller by Bioinspired Techniques", Thesis, Department of ECE NIT, Rourkela, 2011