

Implementation of PID Controlled SIMO Process on FPGA Using Bacterial Foraging for Optimal Performance

Tushar Jain, *Member, IACSIT*, Vishwanath Patel, M.J. Nigam

Abstract— Fast processing in recent microelectronics components and flexibility inherent for all programmable solutions give favor to digital control applications from software-hardware solutions using FPGA. In this paper, the design and implementation of a controlled Single Input Multi Output (SIMO) process is carried out on Field Programmable Gate Array (FPGA). The process is controlled by Proportional-Integral-Differentiator (PID) controller. Bacterial Foraging (BF) algorithm is used for tuning the parameters of PID controller for optimal performance. The work is successfully achieved in the direction to provide the optimal control signal to the physical SIMO process.

Index Terms— FPGA, Bacterial Foraging, PID controller, Optimal performance

I. INTRODUCTION

As the size of single FPGA has increased to several million gates, it has now been practical to consider adding a processor core to the FPGA chip for control applications. Hardware implementation of the controller can be achieved in number of ways; the most popular method of implementing a controller is using a general-purpose microprocessor or microcontroller. Generally, an 8-bit microprocessor can handle most of the necessary computations. Microprocessor based controllers are more economical and flexible, but often face difficulties in dealing with control systems [1] that require high processing and input/output handling speeds. Rapid advances in digital technologies have given designers the option of implementing a controller on a variety of Programmable Logic Device (PLD), FPGA etc. One of the hardware solutions is the ASIC, which is used in situations where high-speed operation is required, and is faster than general-purpose microprocessors. The disadvantage of ASIC is that no change can be made once the chip is created. Another hardware solution is the use of an FPGA, which is suitable for fast implementation and quick hardware verification. FPGA based systems are flexible and can be reprogrammed unlimited number of times.

FPGA is a digital integrated circuit that can be programmed to do any type of digital function. Ability to operate faster and hardware supportable for gates above one million in number are two of its advantage. Y.F. Chan et al [5]

discussed PID control algorithm implementation using a Distributed Arithmetic (DA)-based scheme. In his work, emphasis is given to the utilization of FPGA memory for implementing PID controllers.

II. BACTERIAL FORAGING OPTIMIZATION

The selection behavior of bacteria tends to eliminate poor foraging strategies and improve successful foraging strategies. After many generations a foraging animal takes actions to maximize the energy obtained per unit time spent foraging. This activity of foraging led the researchers to use it as optimization process. The *E coli* bacterium [3] has a control system that enables it to search for food and try to avoid noxious substances. The bacteria distributed motion can model as the following four stages:

A. *Swarming and Tumbling via flagella* (N_s)

The flagellum is a left-handed helix configured so that as the base of the flagellum (i.e. where it is connected to the cell) rotate counterclockwise, as shown in Fig.1-a, from the free end of the flagellum looking towards the cell, it produces a force against the bacterium pushing the cell. This mode of motion is called swimming. A bacterium swims either for maximum number of steps N_s , or less depending on the nutrition concentration and environment condition. During clockwise rotation each flagellum pulls on the cell shown in Fig.1-b, so that the net effect is that each flagellum operates relatively independently of the others and so the bacterium “tumbles”.

B. *Chemotaxis* (N_c)

A chemotaxis step is a set of consequence swim steps following by a tumble. A maximum of swim steps with a chemotactic step is predefined by N_s . The actual number of swim steps is determined by the environment. If the environment shows good nutrients concentration in the direction of the swim, the bacteria swim more steps. When the swim steps is stopped a tumble action takes place.

C. *Reproduction* (N_{re})

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. It is assumed that half of the population members have sufficient nutrients so that they will reproduce with no mutations. For reproduction, the population is sorted in order of ascending accumulated cost (higher accumulated cost represents that it did not get as many nutrients during its lifetime of foraging and hence, is not as “healthy” and thus unlikely to reproduce).

Least healthy group of bacteria dies out and the other healthiest splits into two.

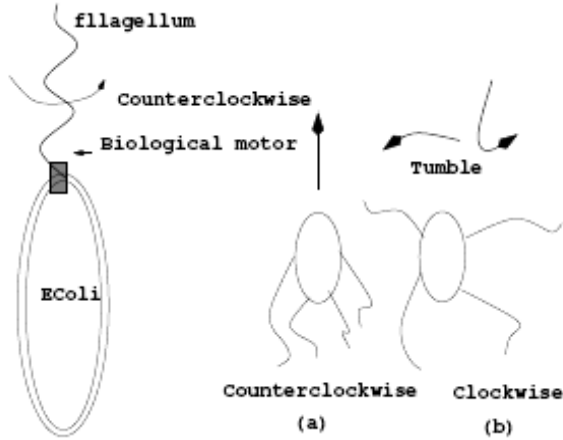


Figure 1. Swarming and Tumbling behavior

D. Elimination and Dispersal (N_{ed})

Elimination event may occur for example when local significant increases in heat kill a population of bacteria that are currently in a region with a high concentration of nutrients. A sudden flow of water can dispose bacteria from one place to another. The effect of elimination and dispersal event is possibly destroying chemotactic progress, but they also have the effect of assisting in Chemotaxis, since dispersal may place bacteria near good food sources.

The bacteria foraging algorithm has been tested for control applications like harmonic estimation for a signal distorted with additive noise [2], and adaptive control [3]. In this paper (*E coli*) is used for tuning PID controller for the unstable system and the results are reported.

III. SIMO CASE STUDY

To investigate the impact of BF on the design of PID controller, a nonlinear SIMO process: Inverted Pendulum system is considered.

A. Inverted Pendulum

The inverted pendulum control problem [4] is usually presented as a pole balancing task. The system to be controlled consists of a cart and a rigid pole hinged to the top of the cart. The cart can move left or right on a one-dimensional bounded track, whereas the pole can swing in the vertical plane determined by the track. The linearized system equations around $q = p$ in the state space are:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (1)$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgk(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \quad (2)$$

where,

- M = mass of cart = 0.5 kg
- m = mass of pendulum = 0.2 kg
- b = friction of cart = 0.1 N/m/sec
- I = inertia of pendulum = 0.006 kgm²
- l = length of pendulum's center = 0.3 m
- u = force applied to cart

The state of the system is defined by values of four system variables: (x, \dot{x}, q, \dot{q}) the cart position, cart velocity, pendulum angle and angular velocity of the pendulum pole, respectively. Control force is applied to the system to prevent the pole from falling while keeping the cart within the specified limits.

B. PID Controller Implementation

The PID control algorithm is one of the most commonly used control algorithms in industry. The controller output is computed in continuous time as follows:

$$u(t) = k_p e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \quad (3)$$

where k_p is the proportional gain, T_i is the reset time and T_d is the derivative time.

An implementation of the PID controller on an FPGA is suggested in [1]. In this paper both parallel and serial PID designs are implemented and performed comparisons between them on the basis of resource utilization, speed and power consumption. The above equation (3.1) is discretised and the following equation is obtained:

$$u(n) = k_p \left[e(n) + \frac{T}{T_i} \sum_{j=0}^n e(j) + \frac{T_d}{T} (e(n) - e(n-1)) \right] \quad (4)$$

where $k_i = k_p T/T_i$ is the integral coefficient and $k_d = k_p T_d/T$ is the derivative coefficient.

The SIMO process considered is physical system which gives analog output but FPGA works on digital system. Therefore, the model used while implementing the design is shown in Fig.2.

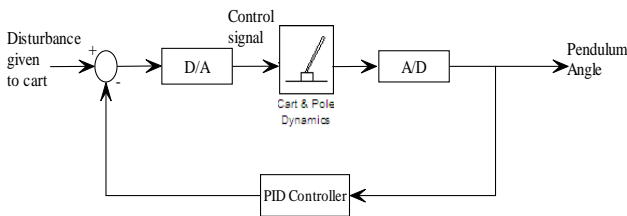


Figure 2. Block diagram of PID implementation

IV. RESULTS AND DISCUSSION

The open loop step response of the pendulum angle and the cart position is shown in Figure 3.

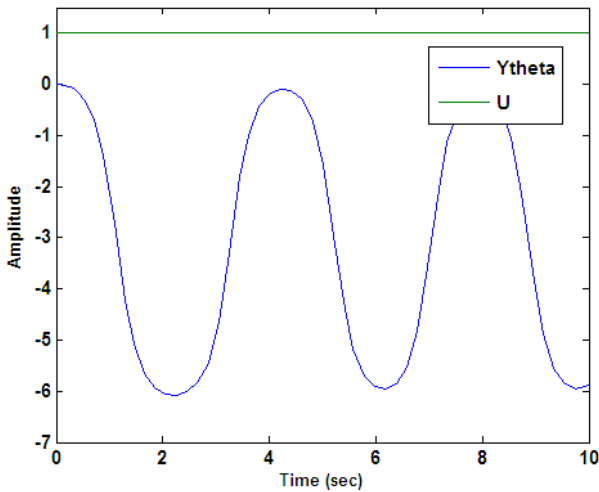


Figure 3. Open loop step response of IP

It can be seen that a unit force on cart sets the pendulum angle at 90^0 therefore, to improve the dynamics of the system; some controller has to be designed. The performance of the PID control can be judged by the values of its parameters. The BF algorithm discussed in Section 2. has been used to estimate the parameters.

The main objective for designing the controller is to keep the control signal with disturbance signal and the pendulum angle at zero level. The values of K_p , K_i and K_d for PID controller are computed using BF optimization technique with Integral Square Error (ISE) as Performance Index (PI) is

$$K_p = 69.391 K_i = 319.6278 K_d = 3.8249 \quad (5)$$

The corresponding transfer functions of control signal and pendulum angle w.r.t. disturbance input are:

$$\frac{U(s)}{R(s)} = \frac{s^3 + 0.1818 s^2 - 31.18 s - 4.454}{s^3 + 17.57 s^2 + 284.2 s - 1448} \quad (6)$$

$$\frac{Y_{theta}(s)}{R(s)} = \frac{4.545 s - 1.425 e - 012}{s^3 + 17.57 s^2 + 284.2 s - 1448} \quad (7)$$

The response of the PID controlled system with step disturbance is shown in Fig.4. Implementation on FPGA requires the discrete functions of control signal and pendulum angle. This invokes another parameter called

sampling time as the design factor. Therefore, the discrete form is given by:

$$\frac{U(z)}{R(z)} = \frac{z^3 - 2.986 z^2 + 2.97 z - 0.9838}{z^3 - 2.812 z^2 + 2.652 z - 0.8389} \quad (8)$$

$$\frac{Y_{theta}(z)}{R(z)} = \frac{0.000214 z^2 - 1.217 e - 005 z - 0.0002018}{z^3 - 2.812 z^2 + 2.652 z - 0.8389} \quad (9)$$

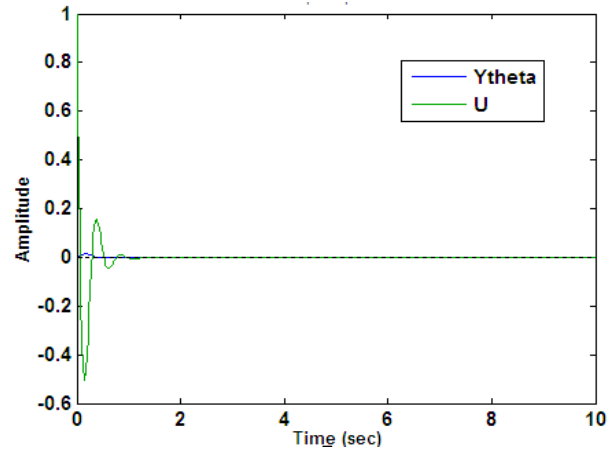


Figure 4. Controlled SIMO process for step disturbance

The system (Eqs. (8), (9)) is now simulated in VHDL and implemented onto the FPGA. The step response is shown in Fig.5, created in Modelsim.

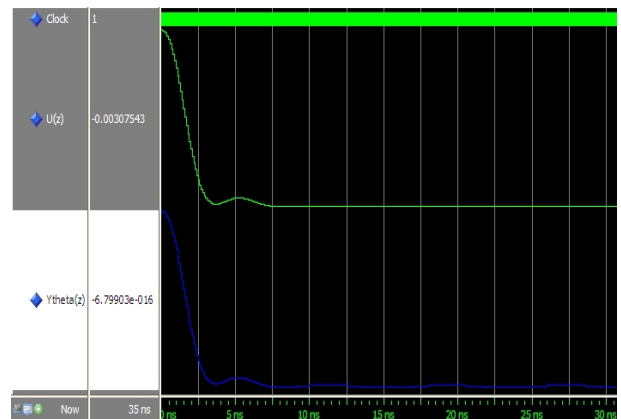


Figure 5. Step Response created in Modelsim

V. CONCLUSION

The FPGA implementation of PID controller shows that only in 10ns, the pendulum angle and the control signal both reached to zero. As from figure.4, control signal requires 1 sec with overshoot and undershoot. This shows the effectiveness of implementing the digital system on FPGA; it invokes the designer to have one extra design parameter i.e. sampling time for optimal performance.

The parallel design and serial design of PID structures will be taken as future work to reduce the power consumption and increased hardware utilization. BF algorithm combined with other intelligent techniques, such as neural networks, expert systems, and fuzzy logic control systems open a new way to design and construct intelligence control systems adapted to complex processes.

REFERENCES

- [1] Wei Zhao, Byung Hwa kim, Amy C. Larson and Richard M. Voyles "FPGA implementation of closed loop control system for small scale robot". International conference on advanced robotics-ICAR 05, pages 70–77, 2005.
- [2] S. Mishra, "A hybrid Least Square-Fuzzy Bacterial Foraging Strategy for Harmonic Estimation," IEEE Trans. Evolutionary Computation, vol. 9, no. 1, pp. 61–73, 2005.
- [3] K.M.Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," IEEE Control Systems Magazine, vol. 22, no. 3, pp. 52–67, June 2002.
- [4] Dadios, Elmer P.; Williams, David J., "A Fuzzy-Genetic Controller for the Flexible Pole-Cart Balancing Problem", Proceedings of IEEE International Conference on Evolutionary Computation, pp 223, ISBN: 0 7803 2902 3, Nagoya, Japan, 1996.
- [5] Y.F. Chan, M. Moallem, W. Wang, "Efficient Implementation of PID Control Algorithm using FPGA technology", 43rd IEEE Conference on Decision and Control, December 2004.