Application of SDGM to Digital PID and Performance Comparison with Analog PID Controller

M. M. Israfil Shahin Seddiqe and Sobuj Kumar Ray

***Abstract*—Now-a-days PID (proportional-integral-derivative) controller is used in every type of small or large industries. For this the digital PID controller is so popular. But it is necessary to design and tune the PID controller in a proper method for applying the plant. The paper represents a method of tuning a digital proportional-integral-derivative (PID) controller. The name of the method is Steepest Gradient Descent Method (SDGM). It is an iterative method. The SDGM method is very simple, easy to use and each iteration is very fast. It is also a guaranteed method to find the minima through numerous times of iterations as long as it exists. By applying this optimization method PID controller is tuned both of digital and analog process. In the paper by applying the method of some plant the comparison result of analog and digital are shown. From the comparison it is found that the digital scheme is preferable than analog scheme.**

***Index Terms*—PID controller, SDGM.**

I. Introduction

The first controllers with proportional, integral, and derivative (PID) feedback control action became commercially available during the 1930s. The 1940s saw widespread acceptance in industry of pneumatic PID controllers, and their electronic counterparts entered the market in the 1950s. Digital hardware has been routinely used since the 1980s with significant impact on process control. Even several decades after three-mode controllers were introduced; the vast majority of controllers used in the chemical process industry are based on PI/PID models [1]. The popularity of these controllers has led to research on tuning methods, resulting in hundreds of publications on this topic. Ziegler-Nichols tuning relations and Cohen-Coon tuning rules are among the earliest published methods. Tuning relations based on error criteria are more recent model-based tuning rules, which offer improvements over earlier tuning methods. Tuning rules also exist for unstable processes as well as for tuning in the presence of plant model mismatch [2]. Despite the numerous approaches available for controller tuning, surveys indicate that poorly tuned control loops are abundant in industry. In general, the gain scheduling of the PID controllers is an optimization problem. The method of steepest descent (SDGM) is also known as The Gradient Descent, which is basically an optimization algorithm to find the local minimum of a function. By applying the SDGM to the PID controller, the controller is tuned and applied to some conventional plant then the digital and analog results are compared to the paper.

Manuscript received February 7, 2011; revised September 30, 2011.

M. M. Israfil Shahin Seddiqe is with Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bnagladesh .(e-mail: mmisrafil@gmail. com)

Sobuj Kumar Ray is with International University of Business Agriculture and Technology. (e-mail: sobuj_kumar_ray@yahoo.com)

II. PID Controller

PID controller consists of Proportional Action, Integral Action and Derivative Action. It is by far the most common control algorithm. PID controller's algorithm is mostly used in feedback loops [1]. PID controllers can be implemented in many forms. It can be implemented as a stand-alone controller or as part of Direct Digital Control (DDC) package or even Distributed Control System (DCS). The latter is a hierarchical distributed process control system which is widely used in process plants such as pharmaceutical or oil refining industries [3]. The schematic diagram of the PID controller is given below. Such set up is known as non-interacting form or parallel form.

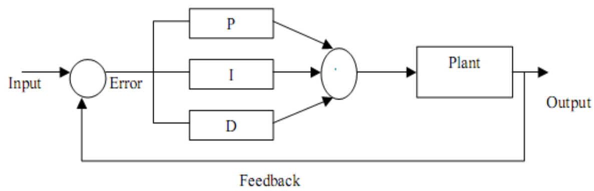

Fig. 1. Schematic of the PID controller - non-interacting form [3]

A. 2.1 Proportional Control

The proportional controller output uses a 'proportion' of the system error to control the system. However, this introduces an offset error into the system.

$$P_{term} = K_P \times e(t) \tag{1}$$

where, $e(t)$ is the error signal of the system and K_P is the gain of the P controller

B. Integral Control

The integral controller output is proportional to the amount of time there is an error present in the system. The integral action removes the offset introduced by the proportional control but introduces a phase lag into the system.

$$I_{term} = K_I \times \int e(t)\ dt \tag{2}$$

where, K_I is gain of the integral controller.

C. Derivative Control

The derivative controller output is proportional to the rate of change of the error. Derivative control is used to reduce/eliminate overshoot and introduces a phase lead action that removes the phase lag introduced by the integral action.

$$D_{term} = K_D \times \frac{d\{e(t)\}}{dt} \tag{3}$$

where, K_D is the gain of the derivative controller.

D. Continuous PID Controller

The three controllers when combined together can be represented by the following transfer function.

$$G_c(s) = K(1 + \frac{1}{sT_i} + sT_d) \tag{4}$$

This can be illustrated below in the following block diagram.

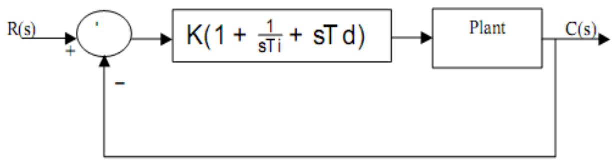

Fig. 2. Block diagram of Continuous PID Controller [3].

What the PID controller does is basically to act on the variable to be manipulated through a proper combination of the three control actions that is the P control action, I control action and D control action. The P action is the control action that is proportional to the actuating error signal, which is the difference between the input and the feedback signal. The I action is the control action which is proportional to the integral of the actuating error signal. Finally the D action is the control action which is proportional to the derivative of the actuating error signal. With the integration of all the three actions, the continuous PID can be realized. This type of controller is widely used in industries all over the world. In fact a lot of research, studies and application have been discovered in the recent years.

E. Digital PID

The digital PID controller i.e. the Z-transform of the PID controller transfer function.

$$\begin{aligned} G_C(Z) &= K_P + \frac{zK_I}{z-1} + \frac{(z-1)}{z}K_D \\ &= \frac{K_P(z^2-z)+z^2K_I+K_D(z^2-1)}{z(z-1)} \\ &= \frac{(K_P+K_I+K_d)z^2-z\,K_P-K_D)}{z(z-1)} \\ &= \frac{(K_P+K_I+K_d)-z^{-1}K_P-z^{-2}K_D)}{1-z^{-1}} \\ &= \frac{q_0+q_1z^{-1}+q_2z^{-2}}{1-z^{-1}} \end{aligned}$$

where,

$$q_0 = K_p + K_I + K_d \qquad q_1 = -K_p$$

$$q_2 = -K_D$$

III. Brief Description of SDGM

The method of steepest descent is the simplest of the gradient methods. Imagine that there's a function f(x), which can be defined and differentiable within a given boundary, so the direction it decreases the fastest would be the negative gradient of f(x). To find the local minimum of f(x), The Method of The Steepest Descent is employed, where it uses a zigzag like path from an arbitrary point x_0 and gradually slide down the gradient, until it converges to the actual point of minimum.

A quadratic form is simply a scalar, quadratic function of a vector with the form

$$f(x) = \tfrac{1}{2}s^TAx - b^Tx + c \tag{5}$$

where A is a matrix, x and b is vector and c is a scalar constant. I shall show shortly that if A is symmetric and positive-definite, f(x) is minimized by the solution to Ax=b. If f(x) is symmetric, then-

$$f'(x) = Ax - b \tag{6}$$

Setting the gradient to zero, we obtain Ax=b, the linear system we wish to solve. Therefore, the solution to Ax=b is a critical point of f(x). If A is positive-definite as well as symmetric, then this solution is a minimum of f(x), so Ax=b can be solved by finding an x that minimizes f(x).

In the method of Steepest Descent, we start at an arbitrary point $x_{(0)}$ and slide down to the bottom of the parabolic. We take a series of steps $x_{(1)}$, $x_{(2)}$,…. until we are satisfied that we are close enough to the solution x. When we take a step, we choose the direction in which f decreases most quickly, which is the direction opposite $f'(x_{(i)})$. According to Equation (6), this direction is $f'(x_{(i)}) = b - Ax_{(i)}$.

The error $e_{(i)}=x_{(i)}-x$ is a vector that indicates how far we are from the solution. The residual $r_{(i)}=b-Ax_{(I)}$ indicates how far we are from the correct value of b. It is easy to see that $r_{(i)}=-Ax_{(I)}$, and you should think of the residual as being the error transformed by A into the same space as b. More importantly, $r_{(i)}=-f'(x_{(i)})$, and you should also think of the residual as the direction of steepest descent.

Suppose we start at $x_{(0)}=[-2,-2]^T$. Our first step, along the direction of steepest descent, will fall somewhere on the solid line in Figure 4(a). In other words, we will choose a point

$$x_{(1)} = x_{(0)} + \alpha r_{(0)} \tag{7}$$

The question is how big a step should we take?

A line search is a procedure that chooses y to minimize f along a line. Figure 3(b) illustrates this task: we are restricted to choosing a point on the intersection of the vertical plane and the parabolic. Figure 3(c) is the parabola defined by the intersection of these surfaces. What is the value of y at the base of the parabola?

From basic calculus, α minimizes f when the directional derivative $\frac{d}{d\alpha}f(x_{(1)})$ is equal to zero. By the chain rule $\frac{d}{d\alpha}f(x_{(1)}) = f'(x_{(1)})^T\frac{d}{d\alpha}x_{(1)} = f'(x_{(1)})^Tr_{(0)}$ Setting this expression to zero, we find that α, should be chosen so that $r_{(0)}$ and $f'(x_{(1)})$ are orthogonal. There is an intuitive reason why we should expect these vectors to be orthogonal at the minimum. Figure 4 shows the gradient vectors at various points along the search line. The slope of the parabola (Figure 3(c)) at any point is equal to the magnitude of the projection of the gradient onto the line (Figure 4, dotted arrows). These projections represent the rate of increase of f as one traverse the search line. f is minimized where the projection is zero—where the gradient is orthogonal to the search line.

To determine α, note that $f'(x_{(1)}) = -r_{(1)}$ and we have

$$r_{(1)}^Tr_{(0)} = 0 \qquad (b - Ax_{(1)})^Tr_{(0)} = 0$$

$$\left(b - A(x_{(0)} + \alpha r_{(0)})\right)^T r_{(0)} = 0$$

$$(b - Ax_{(0)})^T r_{(0)} - \alpha (Ar_{(0)})^T r_{(0)} = 0$$
$$(b - Ax_{(0)})^T r_{(0)} = \alpha (Ar_{(0)})^T r_{(0)}$$
$$r_{(0)}{}^T r_{(0)} = \alpha r_{(0)}{}^T (Ar_{(0)}) \qquad \alpha = \frac{r_{(0)}{}^T r_{(0)}}{r_{(0)}{}^T A r_{(0)}}$$

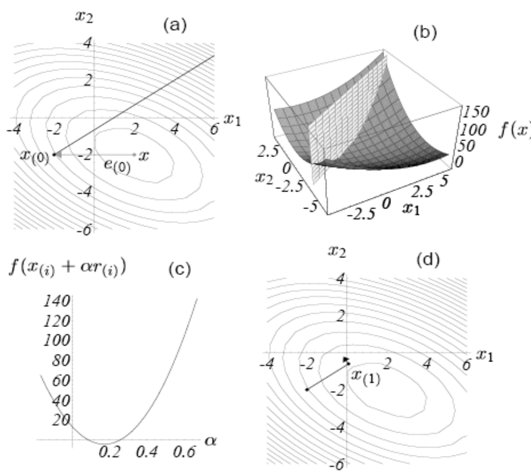

Fig. 3. The method of steepest descent. (a) Starting at $[-2,-2]^T$, take a step in the direction of steepest descent of *f*. (b) Find the point on the intersection of these two surfaces that minimizes *f*. (c) This parabola is the intersection of surfaces. The bottommost point is our target. (d) The gradient at the bottommost point is orthogonal to the gradient of the previous step.

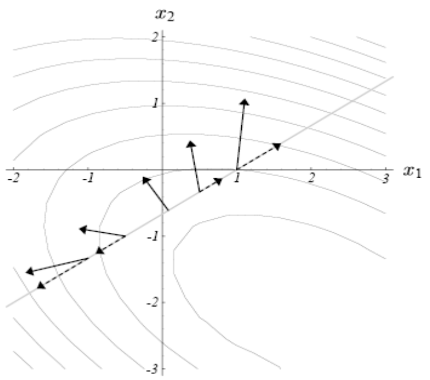

Fig. 4. The gradient *f'* is shown at several locations along the search line (solid arrows).

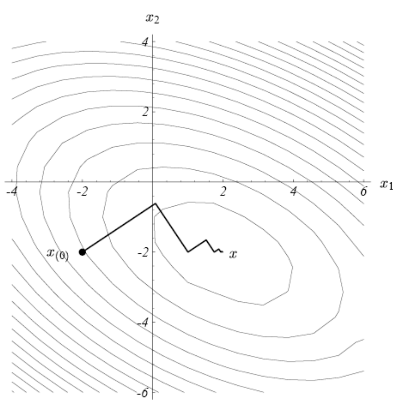

Fig. 5. Here, the method of Steepest Descent starts at $[2,-2]^T$ and converges at $[-2,-2]^T$

Each gradient's projection onto the line is also shown (dotted arrows). The gradient vectors represent the direction of steepest increase of N, and the projections represent the rate of increase as one traverse the search line. On the search line, N is minimized where the gradient is orthogonal to the search line.

Putting it all together, the method of Steepest Descent is:

$$r_{(i)} = b - Ax_{(i)} \tag{8}$$

$$\alpha_{(i)} = \frac{r_{(i)}{}^T r_{(i)}}{r_{(i)}{}^T A r_{(i)}} \tag{9}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} r_{(i)} \tag{10}$$

The example is run until it converges in Figure 5. Note the zigzag path, which appears because each gradient is orthogonal to the previous gradient.

The algorithm, as written above, requires two matrix-vector multiplications per iteration. The computational cost of Steepest Descent is dominated by matrix-vector products; fortunately, one can be eliminated. By premultiplying both sides of equation (3.6) by –A and adding b, we have

$$r_{(i+1)} = r_{(i)} - \alpha_i A r_{(i)} \tag{11}$$

Although Equation (8) is still needed to compute $r_{(0)}$, Equation (11) can be used for every iteration thereafter. The product A*r*, which occurs in both Equations (9) and (11), need only be computed once. The disadvantage of using this recurrence is that the sequence of Equation (11) is generated without any feedback from the value of $x_{(i)}$, so that accumulation of floating point round off error may cause $x_{(i)}$ to converge to some point near *x*. This effect can be avoided by periodically using Equation (8) to recompute the correct residual.

A. 3.1 Steepest Descent Algorithm

The SDGM algorithm will be as follows [5]

Step 1. Estimate a starting design $x^{(0)}$ and set the iteration counter k=0. Select a convergence parameter ε >0.

Step 2. Calculate the gradient of $f(x)$ at the point $x^{(k)}$ as $c^{(k)} = \nabla f(x^{(k)})$. Calculate $||c|| = \sqrt{c^T c}$. If $||c|| < \varepsilon$, then stop the iteration process as $x' = x^{(k)}$ is a minimum point. Otherwise, go to Step 3.

Step 3. Let the search direction at the current point $x^{(k)}$ as $d^{(k)} = -c^{(k)}$.

Step 4. Calculate a step size $\alpha^{(k)}$ to minimize $f(x^{(k)} + \alpha^{(k)} d^{(k)})$. A one-dimensional search is used to determine $\alpha^{(k)}$.

Step 5.Update the design as $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$.

Set $k = k + 1$ and go to Step 2.

IV. Optimization of the PID Controller

The optimizing method used for the designed PID controller is the steepest gradient descent method. In this method, we will derived the transfer function of the controller as

$$G_c(Z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \tag{11}$$

The minimizing of the error function of the chosen problem can be achieved if the suitable values of can be determined. These three combinations of potential values form a three dimensional space. The error function will form some contour within the space. This contour has maxima, minima and gradients which result in a continuous surface. The idea of this optimization method is reach the minima by the shortest path. In order to achieve this shortest path, moving down the steepest gradient will lead to reaching the minima the soonest. When the gradient changes from point to point, to ensure that the steepest path is still being used, it is significant to choose a new direction and make changes accordingly.

V. Simulation Results and Comparison

Some standard plants are chosen for simulating the performance of the controller by tuning the PID controller. The simulation is performed in the Matlab. The initial values of the SDGM are chosen by trial and error method.

Plant 1

A simple fourth order system with G(S) shown in equation is initially chosen as a primary test plant for the proposed method [6].

$$G(s) = \frac{1}{s(s+1)(s+2)(s+3)} \tag{13}$$

From the plot, the gradients of the K_p, K_I and K_D are given below. It is seen that the gradient values are decreasing and in descent direction.

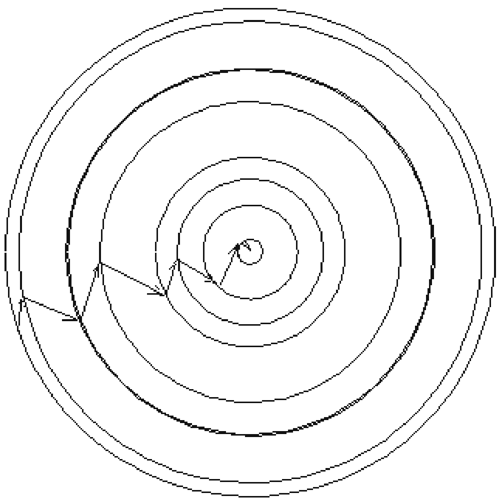

Fig. 6. The convergence of the K_P for Plant 1

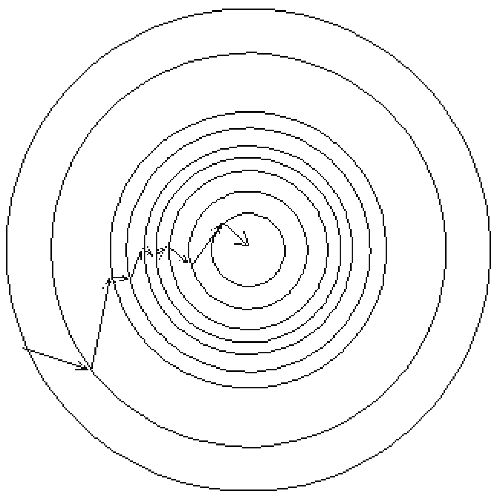

Fig. 7. The convergence of the KI for Plant 1

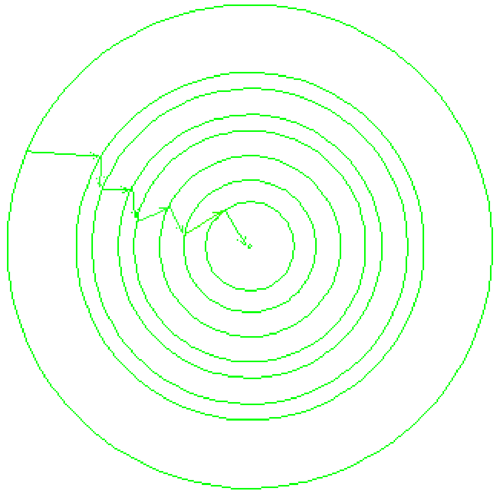

Fig. 8. The convergence of the Kd for Plant 1

The step response output signal is given below. In this plot we can see how the optimized controller behaves. It can be seen that the curve behaves as if it gradually increase to the steady state position and over the steady state level and finally decrease to the steady state level. It will improve its performance until there is little error exists and finally it will reach the final value.

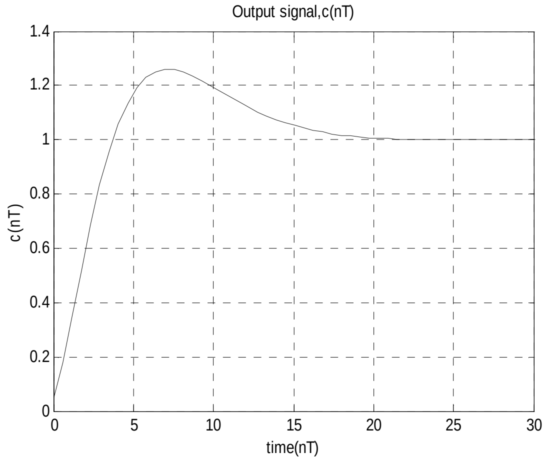

Fig. 9. Optimization output curve with SDGM for Plant 1

Below is the plot of the error signal of the optimized controller. In the figure below it is shown that the error was minimized and this correlate with the response shown in output figure.

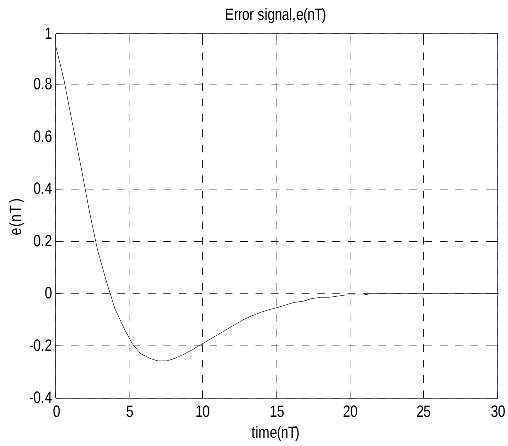

Fig. 10. Error Signal of the Optimized System for Plant 1

As the error was minimized, the system is reaching its stability. From the above figure, the initial error of 1 is finally reduced to zero.

In the curve the parameter values are

Table I: Response of analog PID

Performance	Analog PID
Peak overshoot (%)	16.3
Settling time(s)	15.4
Rise time(s)	0.924

The comparison of the analog and digital PID controller are given below-

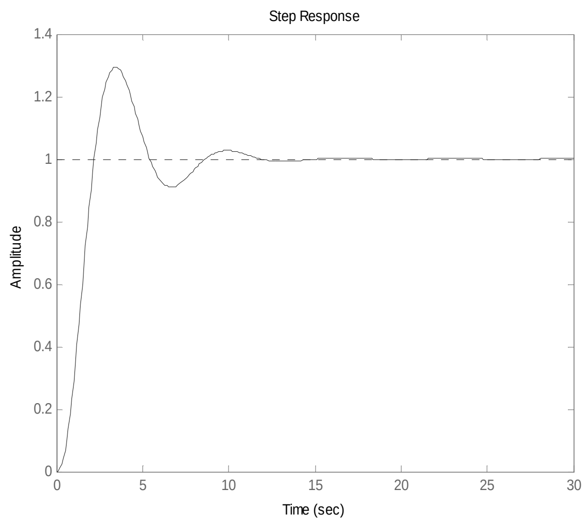

Fig. 11. Analog PID output

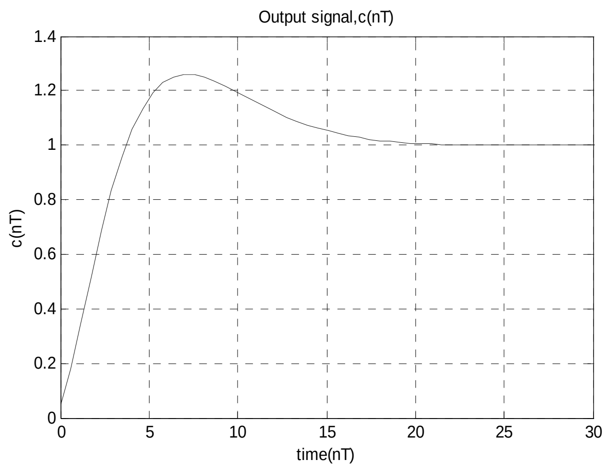

Fig. 12. Digital PID output

Table II: Comparison of analog and digital PID plant 1.

Performance	Analog PID	Digital PID
Peak overshoot (%)	16.3	5
Settling time(s)	15.4	8.99
Rise time(s)	.924	2.88

Plant 2

Another plant is a third order plant, which transfer function is given below [7].

$$G(s) = \frac{1}{s(s+1)(s+5)} \tag{14}$$

The gradients will be as follows.

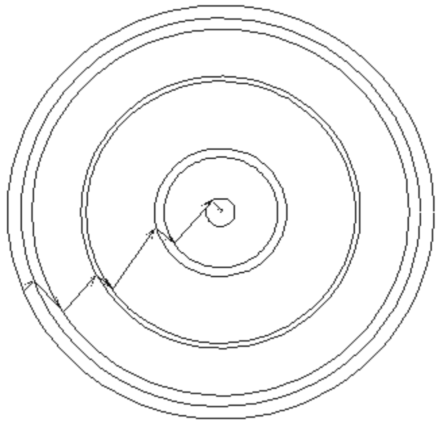

Fig. 13. The convergence of KP for plant 2.

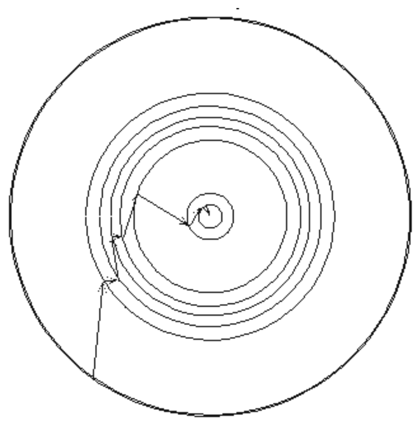

Fig. 14. The convergence of the KI for Plant 2

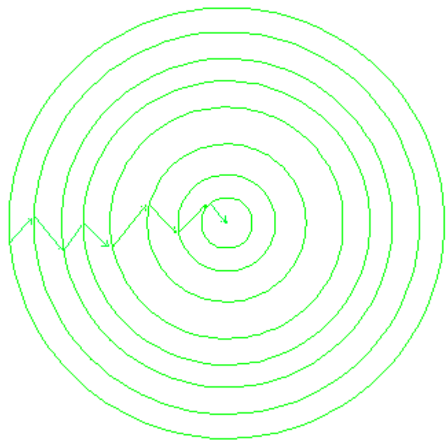

Fig. 15. The convergence of the Kd for Plant 2

The step response output signal for SDGM will be as follows.

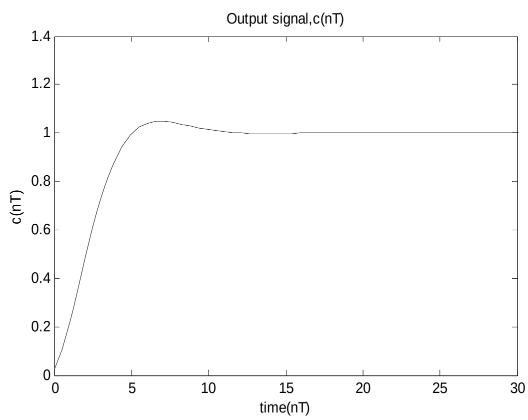

Fig. 16. Optimization output curve with SDGM for Plant 2

The error signal then will be-

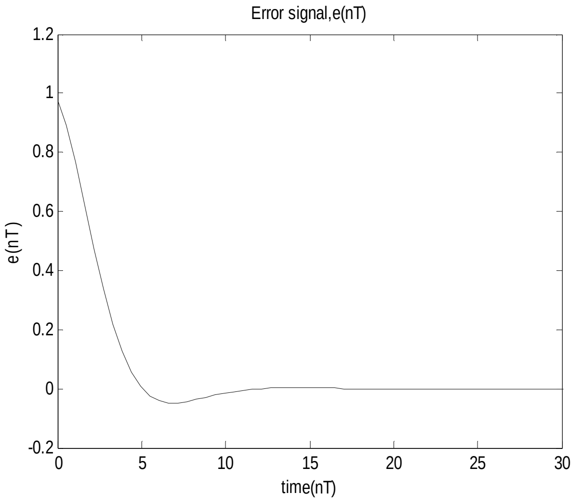

Fig. 17. Error Signal of the Optimized System for Plant 2

The comparison of the analog and digital PID controller are-

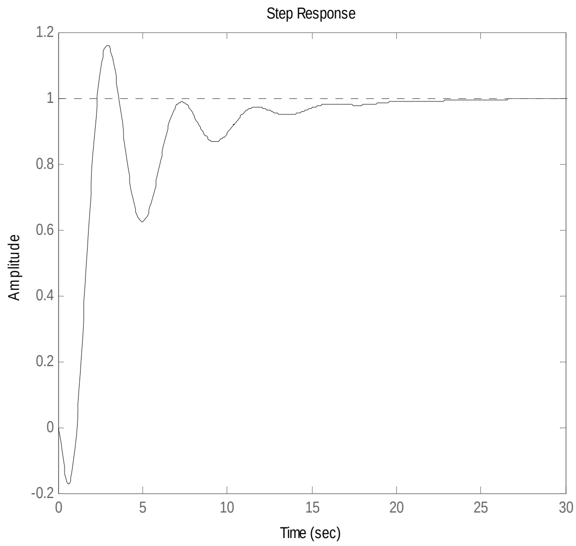

Fig.18. Analog PID

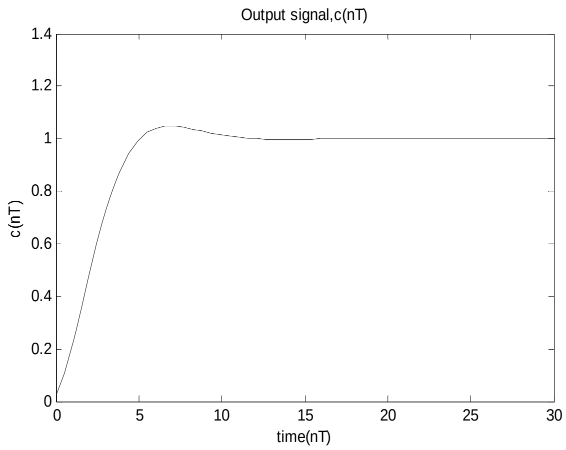

Fig. 19. Digital PID output

The comparison table for the third order system is

Table III: Comparison of analog and digital PID plant 2.

Performance	Analog	Digital
Maximum Overshoot (%)	29.5	26.0
Settling Time (s)	10.7	17.2
Rise Time (s)	1.4	3.0

VI. Conclusion

In this work, the performance of SDGM based gain tuned digital and analog PID controllers applied to some standard plants has been simulated. In order to compare the performance of the system responses, the digital and analog PID output step responses were represented in Fig. 9 and Fig. 16. The parameters of the transient response such as rise time, settling time, overshoot have been calculated and compared. The simulation results reveal that the performance of the SDGM based analog PID controller has been improved with respect to the SDGM based digital PID controller. The comparison result presented in Fig. 12 and Fig. 19 clearly supports the superiority of the digital PID controller over the analog PID controller.

Acknowledgement

I would like to acknowledge Md. Rakibul Islam, Assistant professor, Department of Electrical and Electronic engineering, Rajshahi University of Engineering and Technology, Bangladesh during that period for supervising the B. Sc. Engineering degree.

References

[1] Astrom, K., T. Hagglund, “PID Controllers; Theory, Design and Tuning”, Instrument Society of America, Research Triangle Park, 1995.

[2] Aidan O’Dwyer, “Handbook of PI and PID Controller Tuning Rules” Imperial College Press, London, 2003.

[3] Saifudin Bin Mohamed Ibrahim, “The PID Controller Design Using Genetic Algorithm,” University of Southern Queensland Faculty of Engineering and Surveying, 2005.

[4] Jonathan Richard Shewchuk. “An Introduction to the Conjugate Gradient Method without the Agonizing Pain” August 4, 1994, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[5] Kefu Liu, “Steepest Descent Method”

[6] Ian Griffin, “On-line PID Controller Tuning using Genetic Algorithms,” *Dublin City University*, 2003.

[7] K. Ogata, “Engenharia de Controle Moderno”, Prentice-Hall do Brasil, 1997.

Mr. M. M. Israfil Shahin Seddiqe was born in 1988. Mr. Israfil received his Bachelor degree in Electrical and Electronic Engineering from Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bnagladeshin April 2010. The major fields of study is control system.

Mr. Sobuj Kumar Ray was born in 1987, Bogra, Bangladesh. Mr. Ray received his Bachelor degree in Electrical and Electronic Engineering from the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh in April 2010. Now he is a faculty in the department of Electrical and Electronic Engineering, International University of Business Agriculture and Technology, Dhaka, Bangladesh (www.iubat.edu). The major field of study is control system.