

Development of Simulation Tool for Digital PID Controlled Systems

¹A.T. Ajiboye

Engineering and Scientific Services Department,
National Centre for Agricultural Mechanization,
Ilorin, Nigeria
atajiboye@yahoo.co.uk

²T.S. Ibiyemi

Department of Electrical and Electronics Engineering,
University of Ilorin, Ilorin, Nigeria
ibiyemits@yahoo.com

³J.A. Falade

Department of Electrical and Electronics Engineering,
University of Ilorin, Ilorin, Nigeria
falade.alaba@yahoo.com

¹Corresponding Author

ABSTRACT

Simulation tool for both the uncompensated and PID controlled second-order systems were developed. The system discrete-time response model was developed and coded using Visual Basic 6.0 version. Simulation of both the uncompensated and PID controller compensated standard second-order system was carried out using the developed tool. The resulting time-domain performance indices from the step responses showed no significant difference in the results obtained using the developed tool and MATLAB tool. This confirms the suitability of the developed tool in the area of simulation of any second-order system with or without PID controller.

Keywords- simulation tool; digital PID controller; second order system; uncompensated system; compensated system.

African Journal of Computing & ICT Reference Format:

A.T. Ajiboye, T.S. Ibiyemi and J.A. Falade (2013). Development of Simulation Tool for Digital PID Controlled Systems.
Afr J. of Comp & ICTs. Vol 6, No. 4. Pp 93-

1. INTRODUCTION

The role of controller in control system is to provide the excitation for the system under control and to control the overall system behaviour in order to meet desired performance specifications. The controller and system of interest in this work is Proportional-Integral-Derivative, PID controller and second-order system respectively. This is because PID controller is the most commonly used and best-known controller in linear and nonlinear industrial control systems [1], [2], [3], [4], [5], [6] and higher order systems can often be approximated by second-order systems [7]. PID controller is optimal for second order linear process without time delay and higher order systems with dominant second order behaviour [8].

Simulating any particular system involves the use of software to enable one predict system performance. Many simulation packages have been developed for the purpose of analysis of system performance using system mathematical models. Some of these include Jitterbug, TrueTime and MATLAB/Simulink [8].

For the purpose of this work, a separate computer programme was developed rather than adapt the existing software packages for use. This was done to enable us have in-depth knowledge of the steps involved in the development of control system simulation tools and also to expose us to simulation tool development for a particular system because the existing simulation software are generic in nature.

The plant and the controller transfer functions were transformed from s-domain to z- domain. Then, the inverse z-transform was taken to obtain the closed-loop discrete-time step response in the form of difference equation. Algorithms for solving the resulting difference equations were developed. Based on the developed algorithm, computer programme was written in Microsoft "Visual BASIC version 6.0" and then run on a microcomputer. Visual BASIC is an event-driven programming language for the Microsoft Windows environment. The choice of this language was based on its Graphical User Interface (GUI) capabilities.

This allows for a user-friendly environment in the entry of system parameters and the ease of plotting system response graphs. The step responses for uncompensated and compensated system were also plotted using MATLAB, this serves as standard for the performance evaluation of the developed tool. The resulting time-domain performance analysis from the step responses shows no significant difference in the results obtained using the developed tool and MATLAB tool. This confirms the applicability of the developed tool in the area of simulation of any second-order system with or without PID controller.

2. SYSTEM MODELLING

Figure 1 is the block diagram representation of a unity feedback control system, $R(s)$, $D(s)$, $G(s)$ and $Y(s)$ is the reference input, controller transfer function, plant transfer function and output.

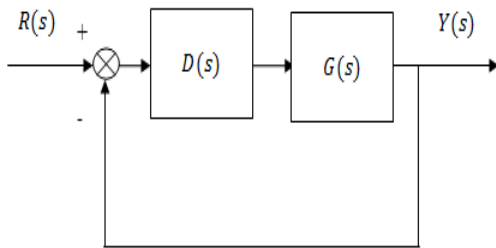


Figure 1: Block Diagram representation of a Unit Feedback Control System

The PID controller open loop transfer function, $D(s)$ is given by

$$D(s) = K_p + \frac{K_i}{s} + K_d s \quad \dots\dots\dots (1)$$

The discrete version of the transfer function of equation (1) was obtained using Euler Backward method which required just substitution of $(z - 1)/Tz$ for s in the controller continuous-time transfer function in order to obtain the transfer function in z-domain and the resulting transfer function is as expressed in equation (2).

$$D(z) = \frac{[(K_p T_i T + K_p T^2 + K_p T_d T_i) z^2 - (K_p T_i T + 2K_p T_d T_i) z + K_p T_d T_i]}{T_i T z^2 - T_i T z} \quad \dots\dots\dots (2)$$

where K_p , T_i , T_d and T are proportional gain, integral time, derivative time and sampling period respectively.

The open-loop transfer function for a standard second-order system 'G(s)' is given by

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s} \quad \dots\dots\dots (3)$$

where ω_n is the system natural frequency in rad/s and ζ is the system damping ratio.

The discrete version of equation (3) was obtained by applying Euler Backward method as given in equation (4).

$$G(z) = \frac{Az + B}{Fz^2 - Jz + L} \quad \dots\dots\dots (4)$$

where

$$\begin{aligned} A &= 2\zeta^2\omega_n T + e^{-2\zeta\omega_n T} - 1 \\ B &= 1 - 2\zeta^2\omega_n T e^{-2\zeta\omega_n T} - e^{-2\zeta\omega_n T} \\ F &= 4\zeta^2 \\ J &= 4\zeta^2(1 + e^{-2\zeta\omega_n T}) \\ L &= 4\zeta^2 e^{-2\zeta\omega_n T} \end{aligned}$$

For better system performance, the sampling rate, f_s should be between $20f_n$ and $40f_n$ [9], [10]. Therefore, for the purpose of this work, sampling rate of $30f_n$ was used. Since sampling period is the reciprocal of sampling rate, we then have

$$T = \frac{2\pi}{30\omega_n} \quad \dots\dots\dots (5)$$

Using equation (4) the closed loop discrete-time response at any sampling instant k for uncompensated second order system, $C(k)$ was determined to be

$$C(k) = \frac{1}{F} [AR(k-1) + BR(k-2) - (A-J)C(k-1) - (B+L)C(k-2)] \quad \dots\dots\dots (6)$$

where $R(k)$ is the input at any sampling instance k .

The closed-loop discrete-time response at any sampling instance k for the compensated system, $C_1(k)$ was also determined by using equations (2) and (4) and is as expressed in equation (7)

$$C_1(k) = \frac{1}{v_3} [M_2 R(k-1) + N_2 R(k-2) + P_2 R(k-3) + Q_2 R(k-4) - (M_2 - W_2)C_1(k-1) - (N_2 + X_2)C_1(k-2) - (P_2 - Y_2)C_1(k-3) - Q_2 C_1(k-1)] \quad \dots\dots\dots (7)$$

where $R(k)$ is the input at any sampling instance k and

$$\begin{aligned} M_2 &= A(K_p T_i T + K_p T^2 + K_p T_d T_i) \\ N_2 &= B(K_p T_i T + K_p T^2 + K_p T_d T_i) - A(K_p T_i T + 2K_p T_d T_i) \\ P_2 &= AK_p T_i T_d - B(K_p T_i T + 2K_p T_d T_i) \\ Q_2 &= BK_p T_i T_d \end{aligned}$$

$$\begin{aligned}V_2 &= FT_iT \\W_2 &= JT_iT + FT_iT \\X_2 &= LT_iT + JT_iT \\Y_2 &= LT_iT\end{aligned}$$

3. DEVELOPMENT OF ALGORITHM

To plot the step response for either the uncompensated or compensated system there is the need for solving equation (6) and equation (7) respectfully. Therefore, the algorithms for solving these equations were developed and are listed under algorithm 1 and algorithm 2 respectfully.

Algorithm 1:

S1: Input system parameters, ω_n , T , and ζ
 S2: Input total response time t
 S3: Compute number of samples $n = t/T$
 S4: $C(0) = 0$; $C(1) = A/F$
 S5: $k = 2$

$$C(k) = \frac{1}{F} [A(k-1) + B(k-2) - (A-J)C(k-1) - (B+L)C(k-2)]$$

$$k = k + 1$$

 S6: If $k < 1$ Go To S6
 S8: If $k < 1$ Go To S6
 S9: STOP

Algorithm 2:

S1: Input system parameters, ω_n , T , and ζ , K_p , T_i and T_d
 S2: Input total response time t
 S3: Compute number of samples $n = t/T$
 S4: $C_1(0) = 0$; $C_1(1) = \frac{M_2}{v_3}$;

$$C_1(2) = \frac{1}{v_3} [M_2 + N_2 - (M_2 - W_2)C_1(1)]$$
 ;

$$C_1(3) = \frac{1}{v_3} [M_2 + N_2 + P_2 - (M_2 - W_2)C_1(2) - (N_2 + X_2)C_1(1)]$$

 S5: $k = 4$

$$C_1(k) = \frac{1}{v_3} [M_2 + N_2 + P_2 + Q_2 - (M_2 - W_2)C_1(k-1) - (N_2 + X_2)C_1(k-2) - (P_2 - Y_2)C_1(k-3) - Q_2C_1(k-1)]$$

$$k = k + 1$$

 S6: If $k < n$ Go To S6
 S8: If $k < n$ Go To S6
 S9: STOP

4. SOFTWARE DESIGN

Modular approach was employed for the software design stage. This method involves the breaking down of the overall system into modules and each module developed independently. The developed modules were then concatenated to form a complete system. The overall system with its modules is as shown in Figure 2.

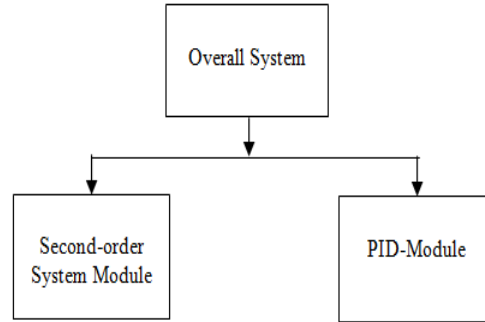


Figure 2: Overall System and its Modules

Each module of Figure 2 is divided into three segments as show in Figure 3

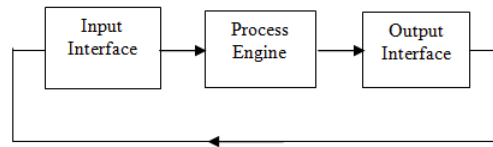


Figure 3: Module Components

The input interfaces are designed to accept system parameters as designed for in the developed algorithm. This interface was designed to take care of any errors in data entry. The processing engine is used for performing specific operation(s) on the input parameters as dictated by the developed algorithm for a particular module. The output interface accepts the results from the processing engine and displays it in a graphical form.

The developed programme was modularized as shown in Figure 4:

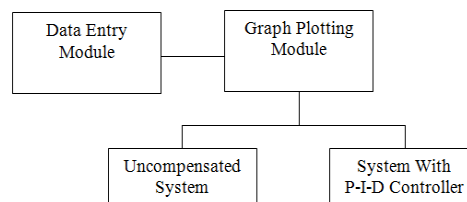


Figure 4: Modular arrangement of the Simulation Programme

The Data Entry module is a set of input fields designed to receive the system and controller parameters like the natural frequency, damping ratio, controller gains as well as the response conditions like the response time and the maximum graph peak. The user enters these parameters while computer, based on predefined formulae generates some other input parameters like the sampling period and the number of samples.

The Graph plotting module is responsible for the actual plotting of the response graph on a picture field and has two sub-modules under it: The sub-module for the uncompensated system, which only plots the graph for a system without the introduction of any form of controller parameter. Also under the graph-plotting module is the sub-module for compensated system, which plots graphs for system with proportional plus integral plus derivative, PID controller. The plots are viewed on a picture field.

For ease of usage, all the modules were integrated into the same parent form as shown in Figure 5.

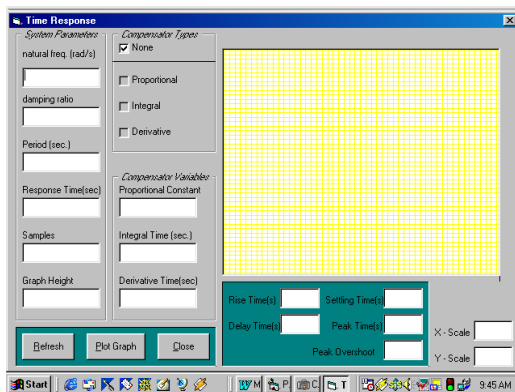


Figure 5: Interface for the Simulation Programme

5. SIMULATION AND DISCUSSION OF RESULTS

The first step in system simulation is to enter the system natural frequency, damping ratio, response time, graph height and controller parameters in the case of compensated system. The value of response time depends on sampling period and the number of samples required. It should also be born in mind that the higher the number of samples the better. The only factor limiting the number of samples is system over-flow which normally occur when the number of samples goes beyond certain level for a given computer system.

For the purpose of this work, an hypothetical second-order system with natural frequency of 200 rad /s and damping ratio of 0.4 together with PID controller with proportional gain, integral time and derivative time of 1, 1000s and 0.004s respectively, were used for the testing of the developed system. To simulate the step response of an uncompensated system, the "None" check box is checked and the "Plot Graph" button is clicked on after entering the system parameters. For the compensated system, after entering the system parameters, the "None" checkbox is checked off and the controller parameter units are checked, after which the parameters are entered before the "Plot Graph" button is clicked on. The step responses for uncompensated and compensated system were also plotted using MATLAB, this serve as standard for the performance evaluation of the developed tool.

Figures 6 and 7 are the responses obtained using the developed tool while Figures 8 and 9 were obtained when MATLAB was used.

Control systems are inherently time domain systems this make it necessary to express the system performance indices in time domain. These performance indices for standard second-order system response to a unit step function are rise time, t_r , peak time, t_p , system percentage overshoot, P.O and settling time, t_s . The resulting time-domain performance analysis determined from the step response of Figures 6 to 9 are as presented in Table 1.

From Table 1 it can be seen that there is no significant difference in the results obtained using the developed tool and MATLAB tool. Therefore, the developed tool can be used for simulation of any second-order system with or without PID controller.

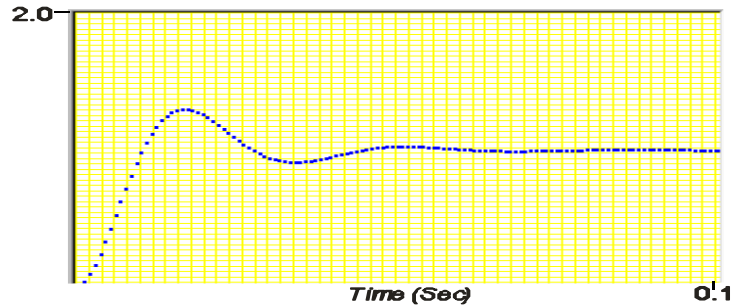


Figure 6: Step Response Graph of System without Compensation using the Developed Tool

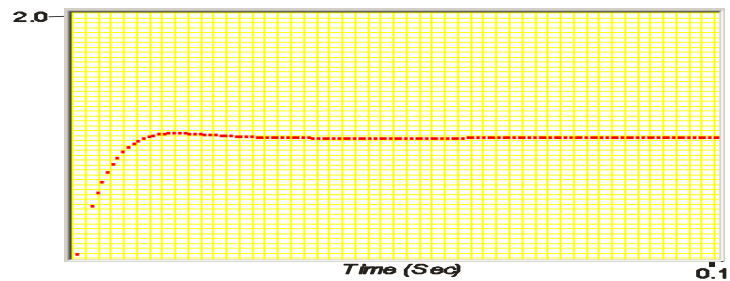


Figure 7: Step Response Graph of System with Compensation using the Developed Tool

Table1: Developed Tool and MATLAB Simulation Results

Figure Number	Controller Parameters			Time domain performance indices			
	K_p	T_i (s)	T_d (s)	t_r (s)	t_p (s)	t_s (s)	P.O
Figure 6	Non	Non	Non	0.0090	0.0180	0.0480	28%
Figure 7	1	1000	0.004	0.0088	0.0190	0.0120	1.2%
Figure 8	Non	Non	Non	0.0074	0.0172	0.0420	25.4%
Figure 9	1	1000	0.004	0.0087	0.0200	0.0130	1.1%

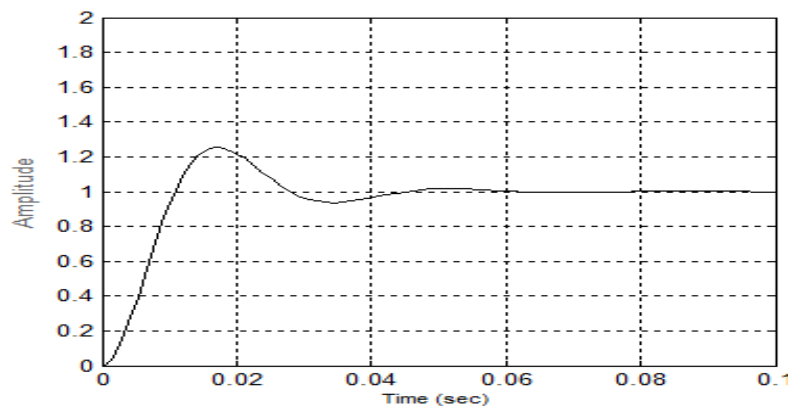


Figure 8: Step Response Graph of System without Compensation using MATLAB Tool

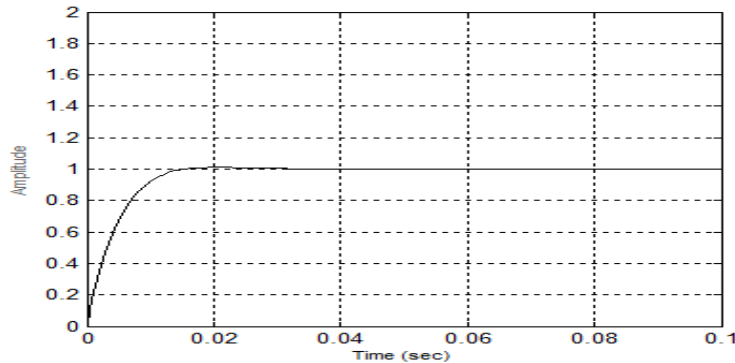


Figure 9: Step Response Graph of System with Compensation using MATLAB Tool

6. CONCLUSION

A simulation tool for uncompensated and digital PID controller compensated second-order control systems was developed. The tool was used for the simulation of both uncompensated and compensated second-order systems. When the time domain performance indices obtained using the developed tool were compared with those obtained using MATLAB tool the difference is not significant. Hence, confirming the applicability of the developed tool in the area of simulation of second-order system with or without PID controller.

REFERENCES

- [1] K. J. Åström and T. Hägglund, "The Future of PID Control," *Contr. Eng. Pract.*, Vol. 9, pp. 1163-1175, 2001.
- [2] R. Kelly and J. Moreno. "Learning PID Structure in an Introductory Course of Automatic Control." *IEEE Transactions on Education*, vol.44, no.4, 2001, pp.373.
- [3] S. A. Al-Assadi and L. A. M. Al-Chalabi. "Optimal Gain for Proportional – Integral – Derivative Feedback." *IEEE Control Systems Magazine*, 1987, p.16.
- [4] N. Hohenbichler and J. Ackermann. "Synthesis of Robust PID Controllers for Time Delay Systems." available at: www.ntnu.no/users/skoge/prost/proceedings/ecc03/pdfs/198.pdf
- [5] M. Ali, H. Selamat, M. F. Rahmat and H. Abdulahim. "Optimized Proportional Integral Derivative (PID) Controller for the exhaust Temperature Control of a Gas Turbine System using Particle Swarm Optimization" *International Journal of the Physical Science*, vol.7, no.5, p.721 30th Jan, 2012. Available at <http://www.academicjournals.org/IJPS>. [3/6/2012]
- [6] J. F. Opadiji and A. T. Ajiboye. "Genetic Algorithm-Based Precision Tuning of Digital P-I-D Controller for Second-Order Systems." *Nigerian Journal of Technological Development*. vol.8, no.1, pp.107, 2011.
- [7] J. J. Distefano, A. R. Stubberud, I. V. Williams and K. A. Rao. *Feedback and Control Systems*. Second Edition. Tata McGraw – Hill Publishing Company Limited, New Delhi, India, 2007, Pp. 3.45.
- [8] Pohjola, M. "PID Controller Design in Networked Control Systems." M.Sc in Technology Thesis, Department of Automation and Systems Technology, Helsinki University of Technology. pp.15and 17.
- [9] F. L. Lian,, J. Moyne and D. Tibury. "Network Design Consideration for Distributed Control Systems." *IEEE Transactions on Control Systems Technology*, vol.10, no.2, pp.297 and 302, 2002.
- [10] F. L. Lian, J. Moyne and D. Tibury. "Time Delay Modeling and Sample Time Selection for Networked Control Systems." *Proceeding of ASME – DSC*, vol. xx, 2001, pp.1 – 3.