ACTA TECHNICA CORVINIENSIS – Bulletin of Engineering
Tome V (Year 2012). FASCICULE 2 [April–June]. ISSN 2067-3809

[1.] Anayet U. PATWARI, [2.] Muammer D. ARIF, [3.] S. M. Tawfiq ULLAH

CONTROLLER DESIGN FOR BEHAVIOR PREDICTION OF SECOND ORDER CLOSED LOOP SYSTEM IN AUTOMATED INDUSTRIAL PROCESSES

[1.] DEPARTMENT OF MECHANICAL AND CHEMICAL ENGINEERING, ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT),
BOARD BAZAR, GAZIPUR, DHAKA: 1704, BANGLADESH

ABSTRACT: The process requirements, in a manufacturing environment, are subject to fluctuations dependent on both internal and external factors. Thus, most modern industries utilize automated control systems, which can make the necessary dynamic adjustments for achieving optimum productivity and product quality. The PID, PI, and PD controllers are widely relied upon for such automated control of processes. However, the adjustment and optimization of some of the functional aspects of these controllers, according to the process demands, are time consuming. Such fine tunings are usually done based upon expert knowledge and tech manuals. This study addresses this problem and presents a solution. The authors illustrate the simulation and development of a controller, using a GUI in MATLAB 2008, which is able to accept the process requirements as input and simulate necessary control actions based on the predictions made. The simulator, discussed, only handles Second Order Systems, which are prevalent in automated industrial control systems. The inputs required are the transfer function of the process and some relevant data from the corresponding frequency response domain. From these, the simulator can predict the closed loop behavior of the system. After the predictions, the GUI provides three viable controller choices to the user. The controller options are PI, PD, or PID types which offer greater flexibility to the user in optimizing the present condition of the process based on the fine tuned parameters for the chosen controller as determined by the simulator. Hence, the efficient operation and optimum output of the automated process is ensured.

KEYWORDS: Dynamic Control Simulator, Closed Loop System, Second Order System, PID Controller, Automated Industrial Processes

INTRODUCTION

Manufacturing processes, in today's competitive market, have to balance many almost mutually exclusive goals such as: higher productivity, stringent quality control, flexibility, short product lifetime, lower lead time, and reduced product price. Invariably, such adjustment and optimization of the process are too complex for manual oversight and control. Thus, a majority of modern industries have switched over to dynamic and automated process control systems. Such controllers perform online monitoring and fine tuning of the processes in order to meet all the manufacturing goals. Albaker et al. [1] utilized PID controllers to design the dynamic non-linear flight control of a propeller driven fixed wing Unmanned Ariel Vehicle (UAV). Their simulator could correctly predict the actions necessary for maintaining the commanded flight path of the UAV. Their predicted results were verified using actual experiments performed on the flight guidance of the UAV.

Upadhyay et al. [2] developed an adaptive non-linear feedback controller using PID controllers in order to maintain the temperature of a Continuously Stirred Tank Reactor (CSTR) in a chemical plant. They observed that the results of their simulation were in accord with experimental data. It was also mentioned that their particular controller design was very robust to modeling errors and random disturbances occurring in the tank; CSTRs by nature have strong non-linear behaviors.

Li et al. [3] discussed the application and importance of PID controllers in their review article. They also presented many cases of computer simulation of PID controllers' actions and their experimental validations. They concluded that there was a significant difference between academic research findings on PIDs and the problems encountered in using such controllers in the industry.

However, the development of a simple and user friendly GUI based software, for the prediction of common controller actions and their fine tuning to meet process requirements, has rarely been researched. The authors of this research, therefore, have developed a simulator using MATLAB 2008 [4, 5] in order to predict the closed loop response of second order systems from the process requirements and corresponding frequency domain data as inputs. For guidelines the authors referred to the seminal work of Patwari et al. [6]. Patwari et al. had previously utilized MATLAB 2008 and Visual Basic 6 Enterprise Edition to develop a simulator for the prediction of chip serration frequency during end milling operation. The authors' simulator has a built-in Graphical User Interface (GUI) that illustrates the results of the prediction and provides the user, in vivid details, with *three types of basic controllers to select from. The three types of controllers are: (1) PI (Proportional and Integral), (2) PD (Proportional and Derivative), and (3) PID (Proportional, Integral, and Derivative). These three types of controllers are routinely used in automated control systems and the software provides the flexibility of choosing any one of them. Any controller, chosen, has already been adjusted by the software to meet the optimum output criteria of the particular process under consideration.*

The predictions of the software were validated in the case of a PID controller against experimental data and analysis. The PID controller was selected as it is the most common feedback controller used in dynamic closed loop control systems in automated industrial processes. The results of the validation along with the discrepancies observed are discussed in this paper along with suggestion to improve the prediction process.

CONTROLLER ARCHITECTURES

A. PID CONTROLLERS

PID controllers calculate an error 'e' value, for the process, as the difference between a measured process variable 'Y' and a desired setpoint 'R'. The controller attempts to minimize the error by adjusting the process control inputs 'u'. the PID controller algorithm involves three separate parameters: (1) The Proportional 'P,' denoted by the proportional gain 'K_p', (2) The Integral 'I,' represented by the Integral gain 'K_I,' and (3) The Derivative 'D,' usually represented by the derivative gain 'K_D.' The P depends on the present error, the I on the past error, and the D on the future error. The weighted sum of these three actions is utilized by the PID controller in order to generate the necessary process control signal u for the control of such parameters as the position of a control valve or in the case of the present research the power supply of a heating element. Due to its inherent flexibility, the PID controller is usually used, especially, where the underlying process in too complicated. Figure 1 is the block diagram of a PID controller and equation 1 is the typical transfer function.

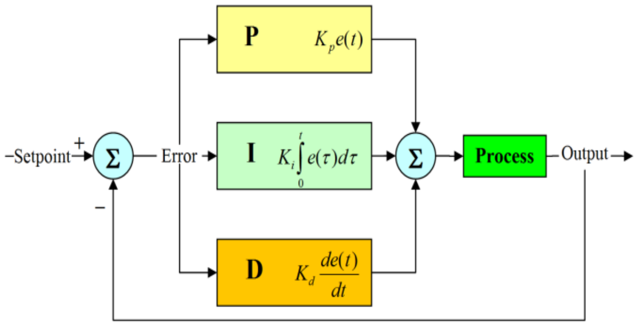

Figure 1 Block diagram of a PID controller

By tuning the three gain parameters, the PID algorithm can provide control instructions for necessary actions designed for a specific process requirement. The responsiveness of the controller is determined by the degree of overshoot and system oscillations. The simulator developed by the authors is capable of fine tuning the gain parameters as well as determining the specific controller response parameters. Consequently, the used is able to select the correct PID controller for the optimal results as well as know the limitations of the controller selected.

B. PI CONTROLLERS

Some applications do not require the use of all three actions and thus, the controller architecture may be simplified by setting the derivative gain to zero. This is done as derivative actions are sensitive to measurement noise. Such a controller is known as PI and has only two gain terms.

C. PD CONTROLLERS

The least common of the three controllers, the PD controller lacks the integral gain term. However, this can sometimes lead to major errors as the absence of an integral term can prevent the system for attaining the target value, due to the control action.

EQUIPMENTS

MATLAB 2008 was the only software used in the development of the simulator and the GUI. For the experimental validation of the simulator predicted results, a standard setup was used which included the following hardware:

1. *Model M3 PID Controller*
2. *Rupert & Co. Ltd. (England) electric heater with electric thermometer.*
3. *A standard analog stopwatch*

METHOD

The GUI (figure 2 and 3) enables the user to perform the whole analysis with ease and precision. In the first stage, of the analysis, the user is asked to enter the coefficients of the PID controllers transfer functions. The frequency response is then selected to generate Bode plots for impulse magnitude and phase, as shown in figure 2. The plots are used to determine the necessary frequency responses such as: Phase Margin, Band Width Frequency, and Constant value. Once these values are inserted into the software prompt by observing from the bode plot, the software displays its predictions for closed loop behavior as in figure 2. The predicted parameters are: Damping Ratio, Setting Time (in seconds), Rise Time (in seconds), and Steady State Error. These are then used to in the next stage of analysis as in figure 3.

The four controller response parameters are then taken into account to select the type of controller (in this case PID) to be employed with specified values for gain coefficients by the software. The result is numerical and graphical representation of the controller action, as shown in figure 3.

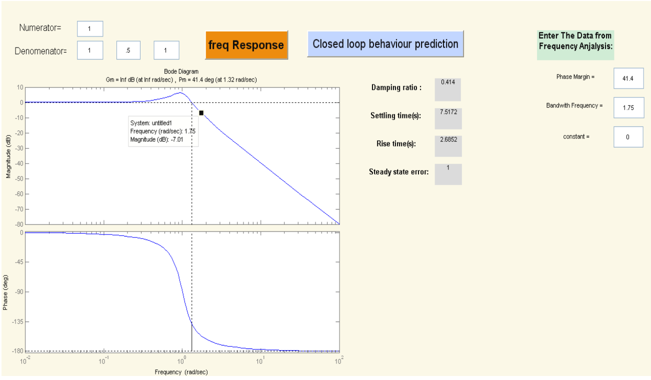
Numerator= 1
Denomenator= 1 .5 1
Damping ratio : 0.414
Settling time(s): 7.5172
Rise time(s): 2.6852
Steady state error: 1
Phase Margin = 41.4
Bandwith Frequency = 1.75
constant = 0
System: untitled1
Frequency (rad/sec): 1.75
Magnitude (dB): -7.01

Figure 2 GUI displaying Bode plots of frequency domain analysis for PID controller

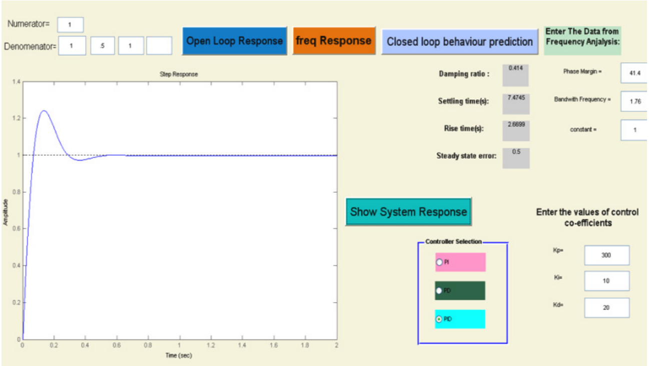
Numerator= 1
Denomenator= 1 .5 1
Damping ratio : 0.414
Settling time(s): 7.4745
Rise time(s): 2.6699
Steady state error: 0.5
Phase Margin = 41.4
Bandwith Frequency = 1.76
constant = 1
Kp= 300
Ki= 10
Kd= 20

Figure 3. GUI showing PID closed loop controller behavior

The experimental validation of the PID controller's predicted response by the software was performed in the laboratory. The PID controller was used to control the temperature and power input to an electric heater. Measured variable was the temperature reached by the heater whose set point was set at T_o = 32°C. The time taken to reach the set point was recorded along with the power input to the heater in increments of 30 seconds.

In the same way PD Controller was also employed to see if the GUI can successfully predict its response also.

RESULTS AND DISCUSSIONS

The results of the simulator's prediction for a specific PID controller and its validation from experimental data are shown in figures 4 to 7.

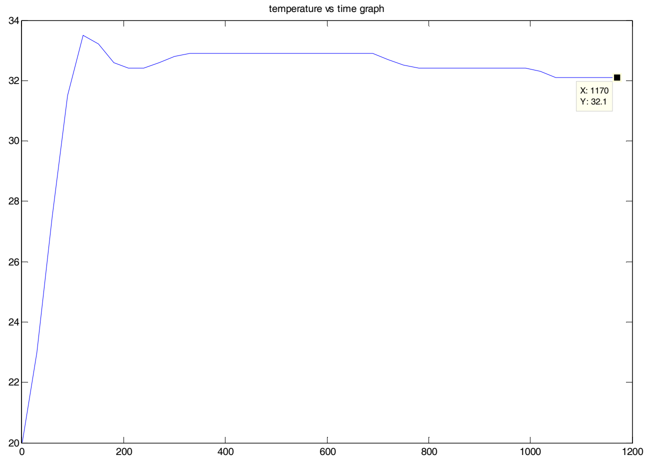

Figure 4. Temperature vs. time plot of predicted response of PID controller as determined by the software

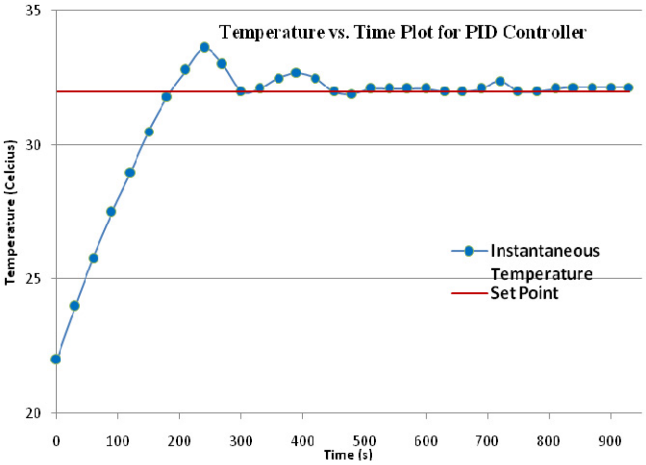

Figure 5. Temperature vs. time plot of PID controller performance as determined by experiment

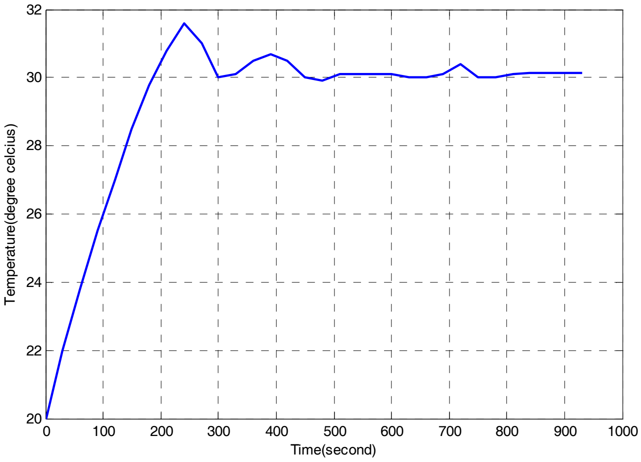

Figure 6. Temperature vs. time plot of predicted response of PD controller as determined by the software with set point 30

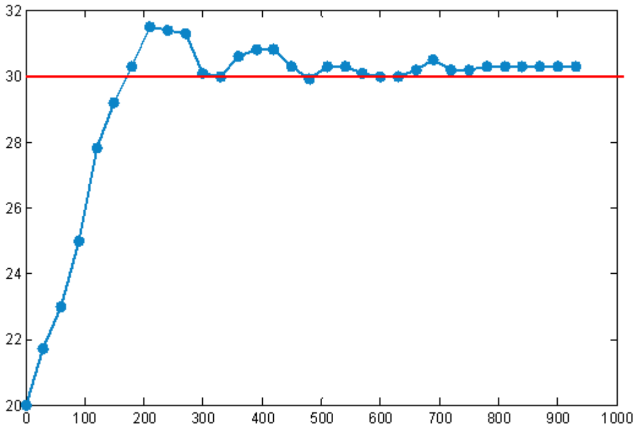

Figure 7. Temperature vs. time plot of PD controller performance as determined by experiment with set point 30

It was observed that the temperature vs. time graphs for both the predicted and experimental results of the PID and PD controller were almost similar. Both curves show the same general shape and fluctuations about the respective set points. Yet, there are certain discrepancies.

PID CONTROLLER

1. For PID controller the time taken to reach a reasonably close neighborhood of the set point of 32 °C was shorter at 930 seconds in the

experimental case compared to 1170 seconds as predicted by the software. This is due to the effect of the surrounding environment during the experiment. Such random external disturbances such as air current, ambient temperature fluctuations, leakage of heat from the setup, and inherent sensor and human error are responsible for this difference.

2. *Also, the time taken to attain the maximum temperature of about 33.5 °C was much shorter in the simulation result. This can again be attributed to external environmental effects.*
3. *Due to the application of PID Controller the system overshoot and steady state error are considerably low and for attaining more reduced rising time and settling time the proportional control action (kp value) should be increased more.*

PD CONTROLLER

4. *For PD Controller the predicted result shows that the rising time is 240 sec whereas the actual experimental result shows that the rising time is about 210 sec, which is very near to the predicted result.*
5. *Both the predicted and experimental results show almost the same response. But the experimental curve shows a bit more steady state error. And the system overshoot is almost negligible in both the cases as here we applied the derivative control here.*

The time taken for the controller to reach a constant temperature which is near to the set point (30 degree) is about 800 second from the experiment. But from the simulated result the time was 830 second. The difference can be considered within the experimental limitations and the random effects of the environment.

CONCLUSIONS

The comparison of the results in the two cases, simulation and experimental setup, only highlight the importance of fine tuning when closed loop controllers are utilized in the real cases. The authors are currently investigating the use of more adaptive and robust controller setups to address these issues. Nonetheless, PID controllers are very common and the simulator is very simple, user friendly, and cheap to implement and use. The authors are confident that their simulator will greatly benefit the work of engineers employed in the industry or involved in state-of-the-art research involving PID, PI, and PD controllers.

REFERENCES

[1] B. M. Albaker and N. A. Rahim, "Flight path PID controller for propeller-driven fixed-wing unmanned aerial vehicles," Int. J. of the Physical Sciences, Vol. 6(8), pp. 1947-1964, 2011.

[2] R. Upadhyay and R. Singla, "Analysis of CSTR temperature control with adaptive and PID controller (a comparative study)," Int. J. of Engineering and Technol., Vol. 2, No. 5, pp. 453-458, 2010.

[3] Y. Li and K. H. Ang., "PID control system analysis and design-problems, remedies, and future directions," IEEE Control Systems Magazine, Vol. 26, No. 1, pp. 32-41, 2007.

[4] Math Works Incorporation, MATLAB User Manual, version 7.0 R14, Natick, MA, 2006.

[5] R. K. Bansal, A. K. Goel, and M. K. Sharma, MATLAB and Its Applications in Engineering, 2nd ed., India: Dorling Kindersley Pvt. Ltd. (under license from Pearson Education in South Asia), pp. 424-454, 2010.

[6] Anayet U Patwari, A.K.M. Nurul Amin, Waleed Faris, "Identification of Instabilities of chip formation by Imag- processing technique" Proceedings of ICME 2009, BUET, Dhaka

ACTA TECHNICA CORVINIENSIS – BULLETIN of ENGINEERING

ISSN: 2067-3809 [CD-Rom, online]

FACULTY OF ENGINEERING HUNEDOARA,
5, REVOLUTIEI, 331128, HUNEDOARA, ROMANIA
http://acta.fih.upt.ro