Journal of Engineering and Applied Sciences 14 (8): 2538-2545, 2019
ISSN: 1816-949X

Tuning the Parameters of the PID Controller Using MATLAB

Beza Negash Getu
Department of Electrical, Electronics and Communications Engineering (EECE),
American University of Ras Al Khaimah (AURAK), Ras Al Khaimah, United Arab Emirates (UAE)

Abstract: In some applications such as controlling temperature of food processing and storage units, controlling speed of mechanical systems in production lines, position controlling in robots, we need to maintain the temperature, speed, position or other parameters of the process at a certain desired set point. Some of these processes work well only within a narrow range of the desired set point. A Proportional Integral Derivative (PID) controller is one of the well-known controllers used in many applications effectively monitoring and controlling the process or the system. In this study, we present how we can tune the parameters of the PID controller optimally using MATLAB Simulation Software.

Key words: Temperature, process (plant), reference value, process output, feedback control, PID, MATLAB

INTRODUCTION

In some industrial processes or home applications, it is vital to control the temperature, pressure, position speed or other parameters of interest to gain proper operation of the process and achieve productivity. For example, an air conditioning system used in houses or in industrial production lines, an oven system for homes, a heat exchanger system used in chemical plants, a furnace and other applications need accurate control of temperature at certain desired point or within a certain range of the desired point (Jie, 2017; Srivastava *et al*., 2014; Suguna *et al*., 2014). A properly designed feedback control system monitors and regulates the system or process to achieve the desired target or output level. Whatever the process or the parameter to control, be it temperature, water level, flow rate inclination of spaceship and airship, speed, pressure, position, direction of vehicle or aircraft and etcetera, the principle of the feedback control system is to set a certain desired level for the input and monitor the level of the corresponding output in a closed loop circuit. The primary purposes of using feedback in control systems is to reduce the sensitivity of the system to output parameter variations and achieve the desired target.

Proportional-Integral and Derivative (PID) controllers are used in most practical feedback control systems such as consumer electronics industrial and chemical processes (Bequette, 2003; Bennett, 1993; Knospe, 2006; Getu, 2016). The PID controller helps to achieve the desired level of output in a short possible time with minimal overshoot and with little steady state error. The performance of the PID controller are also determined by overshoots, rising time, settling time and steady state error parameters.

In this study, we assume a certain process, for example, a temperature dependent process such as a heater or a furnace system and show the capability of the PID controller in achieving the required set temperature. The aim of the controlling is to achieve a perfect process control and as a result increase efficiency and productivity.

The PID controllers are very efficient compared to the traditional contactor-based process ON/OFF type controlling that has several disadvantages such as frequent system failure, high energy consumption, production losses and damage to equipment used for the process (Suguna *et al*., 2014). The PID controller can be implemented using discrete electronic components or a Pulse Width Modulation (PWM) system that can also be generated from an Arduino or other microcontroller system (Allam *et al*., 2016). In the research studies it has been observed that a first order or a second order system can be used to approximately describe the dynamics of most processes (Elssadig and Hussien, 2016; Kealy and O'Dwyer, 2002). In this research, we assume such transfer functions for modelling the process (plant) and illustrate the simulation results of the PID controlled closed loop system. It is shown that the parameters of the PID controller (proportional gain, the integral gain and the derivative gain) can be optimally tuned using MATLAB to yield the desired set point.

System block diagram: Consider a feedback control system as shown in the block diagram given in Fig. 1. The input to the system is a target reference signal, r (t), (could be temperature, water level, position, speed etcetera) that depends on the application process. In the forward path,

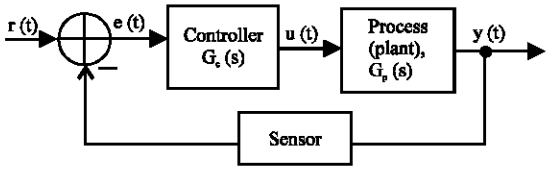

Fig. 1: Block diagram of feedback control system

we have the controller with transfer function G_c (s) and the process with transfer function G_p (s). The controller could be Proportional (P), Proportional Integral (PI), Proportional Derivative (PD), PID or any other type that controls the process to achieve the desired reference target. The error signal, e(t) which is the difference between the process output, y(t) and the reference signal, r (t) will be an input signal to the controller and the process will be adjusted (output increased or decreased) until the required target reference value is achieved. The value of the process output is obtained from a measurement device (a sensor), for example, the sensor can be a thermocouple or a resistance thermometer for temperature sensing, a tachometer for speed measurement and other sensors for other parameters of interest (Fig. 1). If the controller is a PID controller, the error e (t) is related to the output u (t) as (Richard and Robert, 2011) (Eq. 1):

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt} \quad (1)$$

where, K_p, K_i and K_d are called the proportional gain, the integral gain and the derivative gain, respectively. The proportional gain provides an overall control action proportional to the error signal, the integral gain action is to reduce steady-state errors using the integrator and the derivative gain improves transient response through the differentiator. By tuning the 3 parameters of the model, a PID controller can deal with specific process requirements. Using the Laplace domain (Oppenheim *et al*., 1997), the input-output transfer function for the controller is given as (Eq. 2 and 3):

$$G_c(s) = \frac{U(s)}{e(s)} = K_p + \frac{K_i}{s} + sK_d \quad (2)$$

$$G_c(s) = \frac{K_D\left(s^2 + s\frac{K_p}{K_D} + \frac{K_i}{K_D}\right)}{s} \quad (3)$$

MATERIALS AND METHODS

Process model: Consider temperature controlling in a furnace, industrial chemical process or a heating element used at home or commercial buildings. A first order approximation of the heating process is given as (Elssadig and Hussien, 2016; Kealy and Dwyer, 2002) (Eq. 4):

$$G_p(s) = \frac{K_0}{\tau s + 1} \quad (4)$$

where, K_0 process gain, the ultimate value of the response (the steady state value) for a unit step change in the input. τ time constant, the measure of the time needed for the process to adjust to a change in the input. For a step input, the time constant is the defined as the time it takes for the system response to achieve 63% of its total output change (the steady state value). In practice, the model parameters for the process are derived or extracted from the thermal dynamics of the heating element (plant) based on the recorded step response data and appropriate curve fitting technique to the data. Similarly, if the plant or process is approximated by a second order system, the corresponding mathematical model is given as:

$$G_p(s) = \frac{K_0}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (5)$$

where, again τ_1, τ_2 are time constants and this system can be considered as a cascade or series connection of 2 first order systems. The PID controller as given in Eq. 2 and 3 is a compensator system when it is applied for the controlling function in the closed loop system. As a result of the controller, 2 zeros and one pole (at s = 0) are added to the loop transfer function, L(s). For the first order system, L (s) is given by Eq. 6:

$$L(s) = G_p(s)G_c(s) = \frac{K_0 K_D}{\tau} \frac{\left(s^2 + s\frac{K_p}{K_D} + \frac{K_i}{K_D}\right)}{s\left(s + \frac{1}{\tau}\right)} \quad (6)$$

RESULTS AND DISCUSSION

System simulations: The performance of the closed loop system shown in Fig. 1 was studied using MATLAB simulations. Using the PID controller integrated in MATLAB it is easy to simulate the system and observe the output response for different values of K_p, K_i and K_D parameters of the controller. For a unit step input reference for the overall system, our design criteria is to achieve.

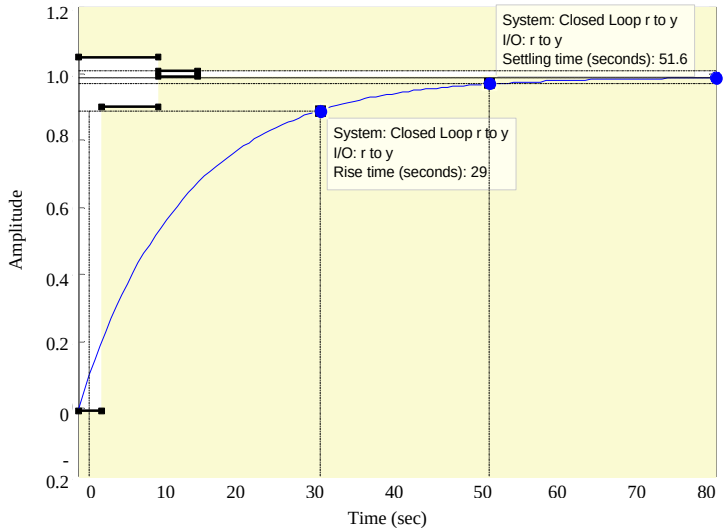

Fig. 2: Closed loop step response of the heating plant without the controller (first order plant)

- An output with a rise time (to 90% of the steady state value) <3 sec
- Settling time <10 sec
- An overshoot <5%
- A steady-state error <1%

For the first order model and for our study purposes, we assume a target temperature value of 90°C (K_0 = 90) and a time constant of τ = 1200 sec and hence, the plant transfer function is Eq. 7:

$$G_p(s) = \frac{90}{1200s+1} \quad (7)$$

The design target is the output response to be within the limits of the performance criteria. Figure 2 shows the system closed loop step response for the first order model without the controller. The goal of the controller is to adjust the output, so that, it tracks the value of the reference input, r (t) within the design criteria defined in the rise time, settling time, percent overshoot and steady state errors. The resulting step response (blue color) is in the shaded background showing that the system is not satisfying the criteria (rise time = 29 sec, settling time 51.6 sec, no overshoot, steady state error nearly zero). The curve needs to trace in the white background region of the figure window to satisfy the requirements. The region is superimposed over the step response curve after we specify the design requirements in the step response window.

Now, let us add a proportional controller (p-type) with the proportional gain K_p in the forward loop. The proportional controller reduces the rise time, improves the steady state error to some extent. As shown in Fig. 3, a value of K_p = 12, satisfies the criteria improving the rise time, the steady state error and the overshoot criteria. The curve traces in the white background part that shows the region within the requirements (rise time = 2.44 sec, settling time = 4.34 sec, no overshoot, steady state error nearly zero). In this case, there is no need to go the other type of PID controllers as long as the proportional controller meets the criteria.

The need of the PID type controller can be illustrated by studying the response of a second order system (plant). Let, us assume a second order system as given in Eq. 8 with parameters K_0 = 90, τ_1 = 20, τ_2 = 15:

$$G_p(s) = \frac{90}{300s^2+35s+1} \quad (8)$$

Figure 4 shows the closed response of the system without the controller showing a big overshoot and high oscillations, very high settling time (rise time = 2.06 sec, overshoot = 71.5%, settling time = 64.3 sec) not fulfilling all the design requirements.

Unlike the first order system, the proportional controller does not give a satisfactory performance with respect to the requirements. Figure 5 and 6 show the output step responses when the proportional controller gain is K_p = 10 and K_p = 0.1, respectively. In case of K_p = 10, the overshoot increased (rise time = 0.617 sec, overshoot = 90%, settling time = 65.5sec) but the rise time reduced whereas in case of K_p = 0.1, the overshoot reduced but the rise time increased (rise time = 7.4 sec,

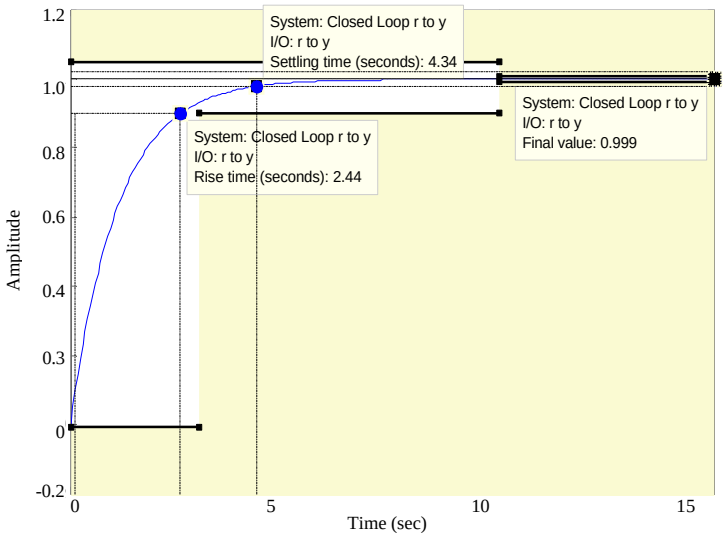

Fig. 3: Closed loop step response with Proportional (P) controller, $K_p = 12$

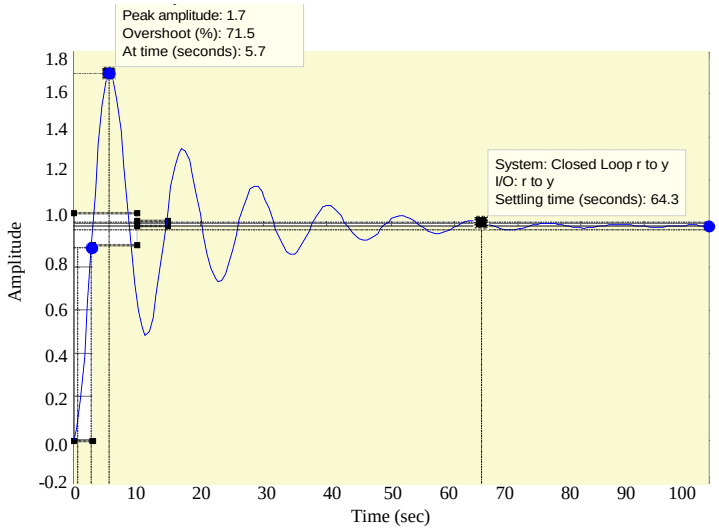

Fig. 4: Closed loop step response of the heating plant without the controller (second order plant)

overshoot = 34.7%, settling time = 61.2 sec). In both cases, the performance requirements are not satisfied.

Figure 7 shows the Proportional-Integral and Derivative (PID) controller with $K_p = 30$, $K_i = 10$ and $K_d = 20$. In this case, we got a much better performance compared to the proportional controller (rise time = 0.246 sec, overshoot = 14.2%, settling time =1.98 sec), however, we still need to improve the overshoot to fulfill the design requirements. The PID controller transfer function for Fig. 7 as in Eq. 9:

$$G_c(s) = \frac{20(s+1)(s+0.5)}{s} \quad (9)$$

Figure 8 shows the step response of the system for PID controller with $K_p = 110$, $K_i = 10$ and $K_d = 100$ satisfying all the requirements (rise time = 0.0671 sec, overshoot 2.74%, settling time = 0.579 sec). With such PID controller system, the desired temperature level can be controlled continuously without the need of manual operation which saves time and manpower from automation. The PID controller automatically responds to the system so, that, the system is stabilized near the desired set point. The PID controller transfer function for Fig. 8 is in Eq. 10:

$$G_c(s) = \frac{100(s+1)(s+0.1)}{s} \quad (10)$$

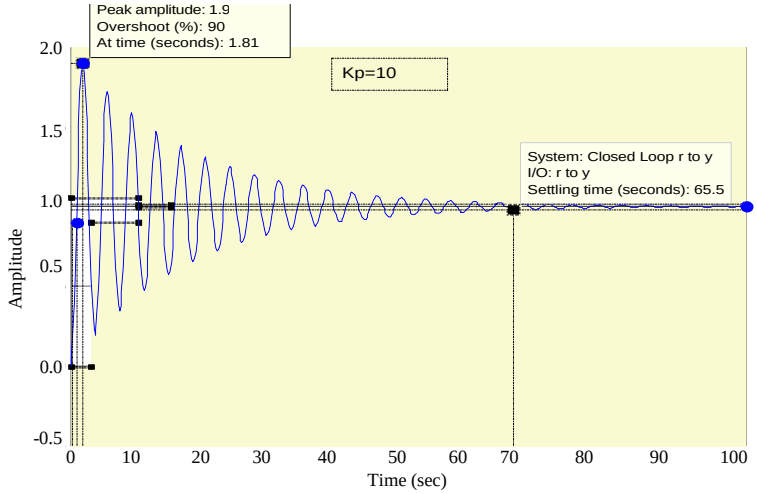

Fig. 5: Closed loop step response with Proportional (P) controller, $K_p = 10$

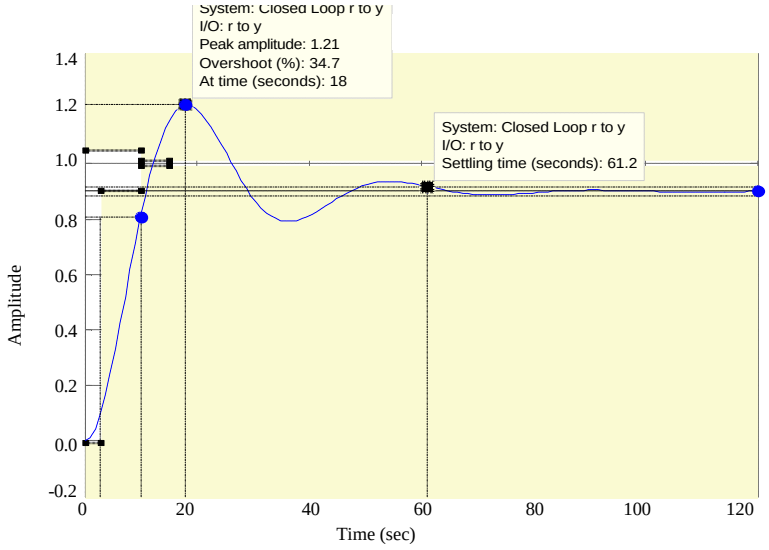

Fig. 6: Closed loop step response with Proportional (P) controller, $K_p = 0.1$

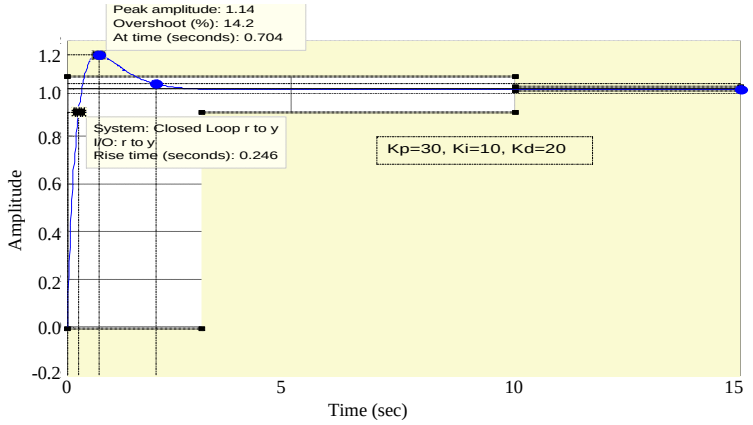

Fig. 7: Closed loop step response with Proportional-Integral and Derivative (PID) controller, $K_p = 30$, $K_i = 10$, $K_d = 20$

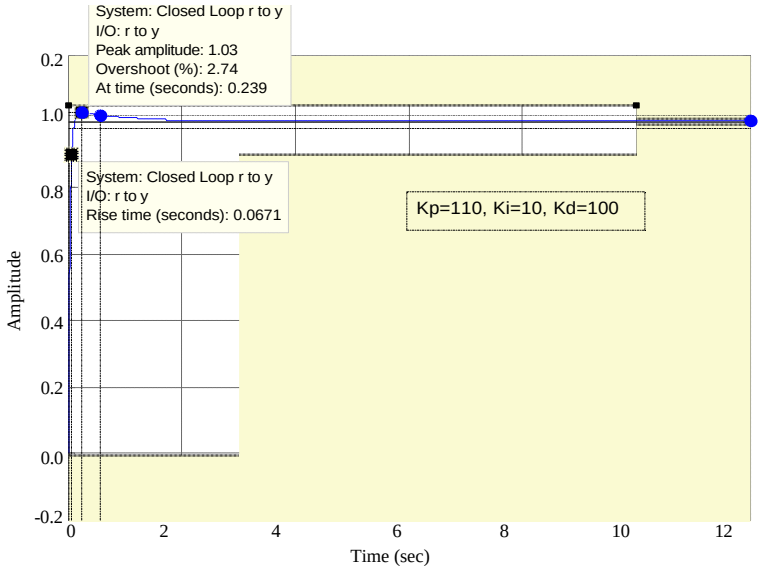

Fig. 8: Step response with Proportional-Integral Derivative (PID) controller (K_p = 110, K_i = 10, K_d = 100)

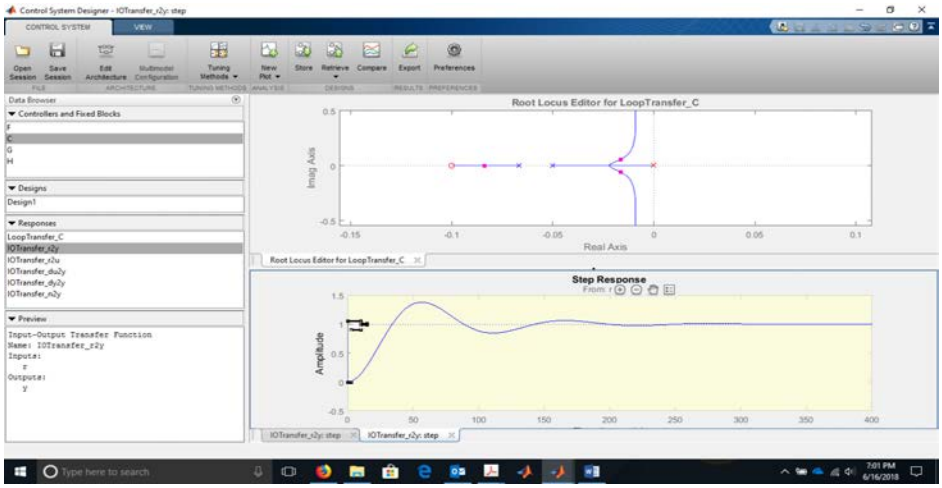

Fig. 9: Tuning the gain of the controller (compensator) using the root locus, PI type controller

Tuning of the pid parameters using the root-locus approach: The root locus technique can be used to manually tune the system parameters until we get the desired performance achievement (Dorf and Bishop, 2011). One approach to manual tuning is to first set the integral and the derivative gains zero (K_i = 0 and K_d = 0) and then slowly increasing the gain K_p until the output of the closed-loop system oscillates just on the edge of instability. Then, K_p is slowly reduced to be in the desired stability region by viewing the root locus plot. The next step in the design process is to increase K_i and K_d manually to achieve the desired step response specifications. By viewing the root locus plot using the MATLAB SISO tool Graphical User Interface (GUI), the parameters K_p, K_i and K_d can be tuned until the desired response is achieved.

For the second order plant given in Eq. 11, the poles of the transfer function, G_p (s) are $p_1 = -1/\tau 1 = -1/20 = -0.05$ and $p_2 = -1/\tau 2 = -1/15 = -0.067$. The root locus starts at the poles of the loop transfer function (poles of L (s) = G_c (s) G_p (s)) and terminates at the zeros. Figure 9 shows the print screen image of the root locus and the corresponding step response with the PI type controller (compensator) where the controller transfer function is as:

$$G_c(s) = \frac{0.0099972\ (s+0.1)}{s} \qquad (11)$$

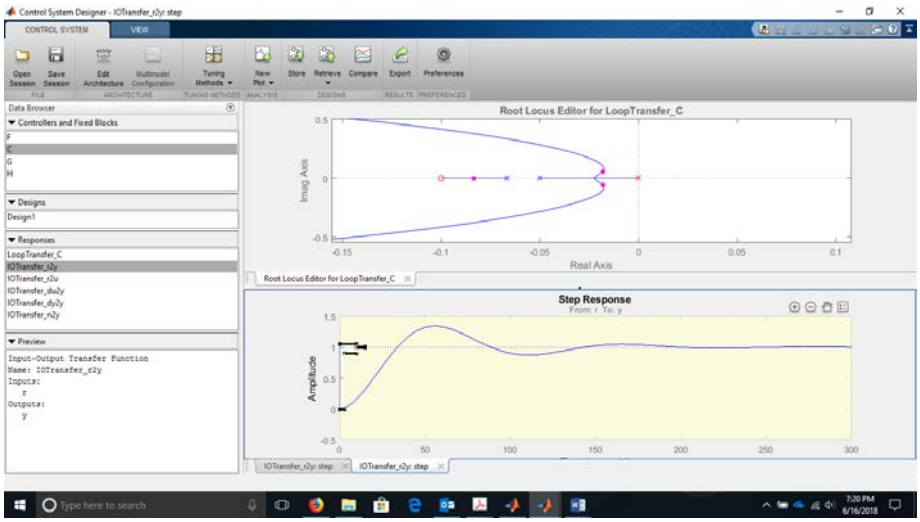

Fig. 10: Tuning the gain of the controller using the root locus, PID type controller (design specifications not satisfied)

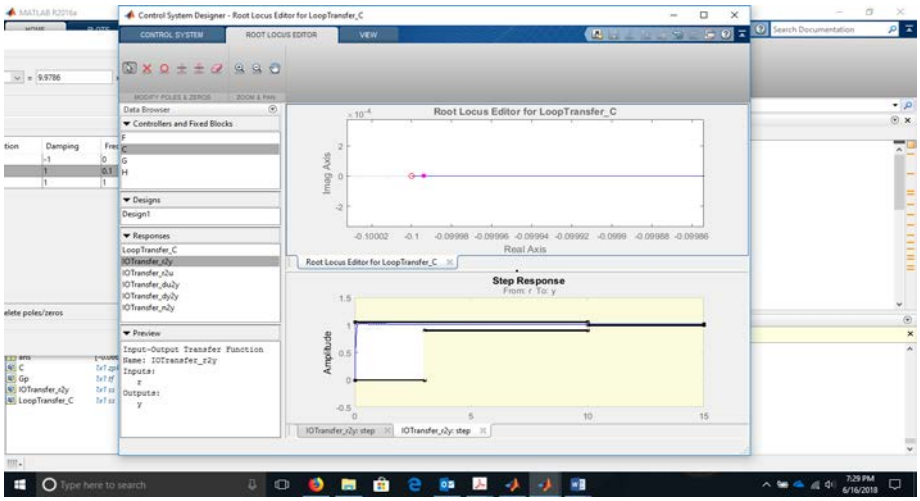

Fig. 11: Tuning the gain of the controller using the root locus, PID type controller (design specifications satisfied)

where, the gain of the controller is 0.0099972. By varying the gain of the controller, the response can be adjusted until the required step response is achieved. The controller gain is again adjusted to make it a PID compensator, this time having an additional real zero at s = -1 where the controller transfer function is Eq. 12:

$$G_c(s) = \frac{0.0099972\ (s+0.1)(s+1)}{s} \qquad (12)$$

The root locus and the step response plot of the system is given in Fig. 10. The gain data tip cursor shown in the figure (pink-square shape) can be used to tune the gain of the controller (compensator) until the desired step response is achieved.

Finally, Fig. 11 shows the root locus and step response when the controller transfer function is PID type with the gain approximately given as in Eq. 10. Only a portion of the root locus is shown as we have to zoom in the figure to tune the compensator gain as far as we desired. This PID controller satisfied our design criteria (rise time, overshoot, steady state error) as can be seen in the step response. From this example, we can learn that the root locus is an alternative way of controlling the system response until desired step response is achieved.

The PID controller in MATLAB can be integrated with the Arduino or other types of microcontroller (Allam *et al.*, 2016). The controller can be used for example to generate PWM pulses that can be used to excite one of

the PWM digital pins of the arduino which in turn controls the supply voltage of a heating plant, a motor or other process where a certain parameter is to be controlled. A temperature or speed sensor senses the temperature of the heating plant or the speed of the motor and gives the information to the Arduino for further tuning of the process until the desired value of the parameter is obtained as defined in the design performance requirements.

CONCLUSION

This study presents how a PID controller system parameters can be tuned using MATLAB Software. The controlled variable of interest can be a temperature, speed, position or other physical quantity of a certain plant or process. It is required to remain at a certain desired set point within the performance constraints as defined by the design requirements. The desired set point can be achieved by tuning the parameters of the PID controller (K_p, K_i, K_d) using the MATLAB Software. The PID controller can be used to provide the desired results for the rise time, settling time, steady state error and overshoot requirements of the process as shown in the MATLAB simulations. The system is essential in those places where maintaining a certain set point of the process parameter is critical for achieving the required productivity.

REFERENCES

Allam, T., M. Raju and S.S. Kuma, 2016. Design of PID controller for DC motor speed control using arduino microcontroller. Intl. Res. J. Eng. Technol., 3: 791-794.

Bennett, S., 1993. Development of the PID controller. IEEE. Contr. Syst., 13: 58-62.

Bequette, B.W., 2003. Process Control: Modelling, Design and Simulation. Prentice Hall, Upper Saddle River, New Jersey, ISBN-13: 9780133536409, Pages: 769.

Elssadig, R.B.K. and E.M. Hussien, 2016. Function using FOPDT, SOPDT and SKOGESTAD in control system. Intl. J. Eng. Appl. Manage. Sci. Parad., 42: 64-69.

Getu, B.N., 2016. Water level controlling system using pid controller. Intl. J. Appl. Eng. Res., 11: 11223-11227.

Jie, D., 2017. Modeling and simulation of temperature control system of coating plant air conditioner. Proc. Comput. Sci., 107: 196-201.

Kealy, T. and A. O'Dwyer, 2002. Closed Loop Identification of a first order plus dead time process model under PI control. Proceedings of the International Conference on Irish Signals and Systems (ISSC 2002), June 25-26, 2002, University College Cork, Cork, Ireland, pp: 9-14.

Knospe, C., 2006. PID control. IEEE. Contr. Syst., 26: 30-31.

Oppenheim, A.V., A.S. Willsky and S.H. Nawab, 1997. Signals and Systems. Prentice-Hall Internationa, Upper Saddle River, New Jersey, USA., ISBN:9780136511755, Pages: 957.

Richard, C.D. and H.B. Robert, 2011. Modern Control Systems. 12th Edn., Prentice Hall, Upper Saddle River, New Jersey, USA., ISBN:9780136024583, Pages: 1082.

Srivastava, N., D.K. Tanti and M.A. Ahmad, 2014. MATLAB simulation of temperature control of heat exchanger using different controllers. Autom. Contr. Intell. Syst., 2: 1-5.

Suguna, R., V. Usha and S. Chidambaram, 2014. A temperature control by using PID based SCR control system. J. Electron. Commun. Eng., 9: 51-55.