PID Controller Design for Controlling Integrating Processes with Dead Time using Generalized Stability Boundary Locus

Serdal Atic*, Erdal Cokmez, Fuat Peker**, Ibrahim Kaya****

** Electricity and Energy Department, Vocational High School, Batman University, Batman, Turkey, (e-mail: serdal.atic@batman.edu.tr).*
*** Electrical and Electronics Engineering Department, Dicle University, Diyarbakır, Turkey, (e-mail: erdal.cokmez@dicle.edu.tr, fuat.peker@dicle.edu.tr, ikaya@dicle.edu.tr)*

Abstract: This paper proposes a method so that all PID controller tuning parameters, which are satisfying stability of any integrating time delay processes, can be calculated by forming the stability boundary loci. Processes having a higher order transfer function must first be modeled by an integrating plus first order plus dead time (IFOPDT) transfer function in order to apply the method. Later, IFOPDT process transfer function and the controller transfer function are converted to normalized forms to obtain the stability boundary locus in $(KK_cT, KK_c(T^2/T_i))$, (KK_cT, KK_cT_d) and $(KK_c(T^2/T_i), KK_cT_d)$ planes for PID controller design. PID controller parameter values achieving stability of the control system can be determined by the obtained stability boundary loci. The advantage of the method given in this study compared with previous studies in this subject is to remove the need of re-plotting the stability boundary locus as the process transfer function changes. That is, the approach results in somehow generalized stability boundary loci for integrating plus time delay processes under a PID controller. Application of the method has been clarified with examples.

Keywords: Stability, PI controller, PID controller, transfer function, dead time, modeling.

1. INTRODUCTION

Researchers have always been interested in PID controllers which are generally used industrial control systems owing to their simple structure and performing robustly. Compared to PD controllers, PI controllers have a larger usage. For this reason, determination of tuning parameters of a PI or PID controller is quite important (Aström and Hagglund, 2001).

Most commonly used methods for determination of PID controllers are Ziegler and Nichols (1942), Cohen and Coon (1953) and Aström and Haggland (1984) methods. Methods based on integral performance criteria (Zhuang and Atherton, 1993) are among very standard approaches as well. Other methods that used for calculating PID controller tuning parameters are Internal Model Control (IMC) (Morari and Zafiriou) and controller synthesis (Smith and Corripio, 1997) methods.

Special interest has been paid to determination of all stabilizing PI and PID controller parameters after the study of Ho et al. (1996, 1997a, 1997b, 1997c). Thanks to these studies, all integral and derivative gain values of a PID controller can be shown in the same plane for a fixed proportional gain value. Although the method provides calculation of all PI and PID controller tuning parameters, application of the method takes time. For that reason, researchers have gravitated to develop different approaches. Munro and Söylemez (2000) and Söylemez et al. (2003) find out a method that provided a faster calculation of all PID controller tuning parameters. Shafiei and Shenton (1997) and Huang and Wang (2000) provided graphical solutions for determination of all stabilizing PID controller parameter values. Tan et al. (2003) and Tan (2005) suggested a new approach providing a faster calculation of all stabilizing PI or PID controller tuning parameters, based on stability boundary locus calculation. This approach has been used in different studies up to date. Zàvackà et al. (2013) suggested a robust PI controller design for a continuous stirred tank reactor with multiple steady-states. Sandeep and Yogesh (2014) gave design of a PID controller for an inverted pendulum. Yogesh (2016) provided a PI controller design for one joint robotic arm. Deniz et al. (2016) recommended an integer order approximation method based on stability boundary locus for fractional order derivative/integrator operators. All of the studies mentioned above consider the case of a specific plant transfer function.

In this paper, the approach suggested by Kaya and Atic (2016) for obtaining all stabilizing PI controllers to control open loop stable time delay processes has been extended to all stabilizing PID controllers to control integrating and time delay processes. In this approach, modelling of higher order processes by a first order plus integrating plus dead time (IFOPDT) model is required. It is assumed that relay feedback identification method of Kaya and Atherton (2001) can be used for this purpose. The relay feedback method gives exact solutions if there are no measurement errors and disturbances entering the control system. Process transfer function model and the controller transfer function are first converted into normalized forms and then used to form stability boundary loci for obtaining all stabilizing PID controller tuning parameters for varying normalized dead time $\tau = \theta / T$. The advantage of the method is to eliminate the need of re-plotting the stability boundary locus whenever the transfer function changes so that calculation of all stabilizing PID controllers becomes easier.

The rest of paper is organized as follows. Next section gives the procedure to obtain stability boundary locus in $\left(KK_c(T^2/T_i), KK_cT_d\right)$ plane for a fixed value of KK_cT to obtain all stabilizing PID controllers. In Section 3, the application of method is illustrated with several examples. Conclusions are given in Section 4.

2. PID CONTROLLER DESIGN FOR THE INTEGRATING PROCESSES

Consider single-input single-output control system depicted in Fig. 1.

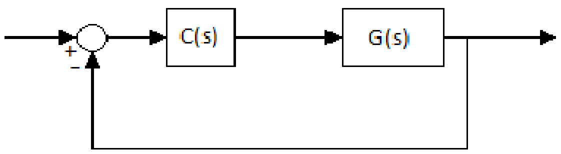

Fig. 1. SISO control system

$C(s)$ and $G(s)$ are the controller and the process transfer functions, respectively. Transfer function for ideal PID controller is:

$$C(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{1}$$

and the IFOPDT model of process transfer function is assumed to be given by:

$$G(s) = \frac{Ke^{-\theta s}}{s\left(Ts+1\right)} \tag{2}$$

By substituting $Ts = \bar{s}$ in (1) and (2), the normalized controller and process transfer functions were obtained:

$$C(\bar{s}) = K_c\left(1 + \frac{T}{T_i\bar{s}} + \frac{\bar{s}T_d}{T}\right) \tag{3}$$

$$G(\bar{s}) = \frac{KTe^{-\frac{\theta}{T}\bar{s}}}{\bar{s}^2+\bar{s}} = \frac{KTe^{-\tau\bar{s}}}{\bar{s}^2+\bar{s}}. \tag{4}$$

Here, the aim is to calculate all controller parameter values in (1) to satisfy the stability of the control system shown in Fig. 1. Closed-loop characteristic equation of the system is $1 + C(\bar{s})G(\bar{s})$. Hence, substituting $C(\bar{s})$ and $G(\bar{s})$, correspondingly, from (3) and (4), the closed loop characteristic equation can be found to be given by:

$$\Delta(\bar{s}) = KK_cT^2T_i\bar{s}e^{-\tau\bar{s}} + KK_cT^3e^{-\tau\bar{s}} + KK_cTT_iT_d\bar{s}^2e^{-\tau\bar{s}} + TT_i\bar{s}^3 - TT_i\bar{s}^2 \tag{5}$$

The numerator and the denominator of (2) have been decomposed into their even and odd parts and $\bar{s} = j\bar{\omega}$ is replaced in order to achieve

$$G(j\bar{\omega}) = \frac{N_e(-\bar{\omega}^2) + j\bar{\omega}N_o(-\bar{\omega}^2)}{D_e(-\bar{\omega}^2) + j\bar{\omega}D_o(-\bar{\omega}^2)}. \tag{6}$$

Dropping the dash over ω for simplicity, the characteristic equation can be written as:

$$\begin{aligned}\Delta\left(j\omega\right) &= j\omega KK_cT^2T_i\cos\left(\omega\tau\right) \\ &\quad + \omega KK_cT^2T_i\sin\left(\omega\tau\right) + KK_cT^3\cos\left(\omega\tau\right) \\ &\quad - jKK_cT^3\sin\left(\omega\tau\right) - \omega^2KK_cTT_iT_d\cos\left(\omega\tau\right) \\ &\quad + j\omega^2KK_cTT_iT_d\sin\left(\omega\tau\right) \\ &\quad - j\omega^3TT_i - \omega^2TT_i \\ &= R_\Delta + jI_\Delta = 0.\end{aligned} \tag{7}$$

By equating the real and imaginary parts of the characteristic equation to zero, the following equations are obtained:

$$KK_cT\left[\omega\sin\left(\omega\tau\right)\right] + \frac{KK_cT^2}{T_i}\left[\cos\left(\omega\tau\right)\right] + KK_cT_d\left[-\omega^2\cos\left(\omega\tau\right)\right] = \omega^2 \tag{8}$$

$$KK_cT\left[\omega\cos\left(\omega\tau\right)\right] + \frac{KK_cT^2}{T_i}\left[-\sin\left(\omega\tau\right)\right] + KK_cT_d\left[\omega^2\sin\left(\omega\tau\right)\right] = \omega^3. \tag{9}$$

Defining the following equations,

$$\begin{aligned}Q(\omega) &= \omega\sin\left(\omega\tau\right), \\ R\left(\omega\right) &= \cos\left(\omega\tau\right), \\ F\left(\omega\right) &= -\omega^2\cos\left(\omega\tau\right), \\ X\left(\omega\right) &= \omega^2 + \omega^2KK_cT_d\cos\left(\omega\tau\right), \\ H\left(\omega\right) &= \omega^2 - \frac{KK_cT^2}{T_i}\cos\left(\omega\tau\right).\end{aligned} \tag{10}$$

and

$$\begin{aligned}S\left(\omega\right) &= \omega\cos\left(\omega\tau\right), \\ U\left(\omega\right) &= -\sin\left(\omega\tau\right), \\ B\left(\omega\right) &= \omega^2\sin\left(\omega\tau\right), \\ Y\left(\omega\right) &= \omega^3 - \omega^2KK_cT_d\sin\left(\omega\tau\right), \\ N\left(\omega\right) &= \omega^3 + \frac{KK_cT^2}{T_i}\sin\left(\omega\tau\right).\end{aligned} \tag{11}$$

Equations (8) and (9) are rewritten as follows:

$$\begin{aligned}KK_cTQ(\omega) + KK_c\frac{T^2}{T_i}R(\omega) &= X(\omega), \\ KK_cTS(\omega) + KK_c\frac{T^2}{T_i}U(\omega) &= Y(\omega).\end{aligned} \tag{12}$$

and

$$\begin{aligned}KK_cTQ(\omega) + KK_cT_dF(\omega) &= H(\omega), \\ KK_cTS(\omega) + KK_cT_dB(\omega) &= N(\omega).\end{aligned} \tag{13}$$

Equations (12) and (13) can be solved to obtain the following expressions:

$$KK_cT = \frac{X(\omega)U(\omega) - Y(\omega)R(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)}, \tag{14}$$

$$KK_c\frac{T^2}{T_i} = \frac{Y(\omega)Q(\omega) - X(\omega)S(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)}, \tag{15}$$

and

$$KK_cT_d = \frac{N(\omega)Q(\omega) - H(\omega)S(\omega)}{Q(\omega)B(\omega) - F(\omega)S(\omega)} \tag{16}$$

Equations (10) and (11) are substituted into (14), (15) and (16) to gain the following equations:

$$KK_cT = \omega\sin(\omega\tau) + \omega^2\cos(\omega\tau), \tag{17}$$

$$KK_c\frac{T^2}{T_i} = -\omega^3\sin(\omega\tau) + \omega^2\cos(\omega\tau) + \omega^2 KK_cT_d, \tag{18}$$

and

$$KK_cT_d = \omega\sin(\omega\tau) - \cos(\omega\tau) + \omega^{-2}\frac{KK_cT^2}{T_i}. \tag{19}$$

Stability boundary loci in $(KK_cT, KK_c(T^2/T_i))$ plane for the normalized dead time value of $\tau = 1$ and fixed $KK_c\mathrm{T}_d$ values of 1 and 0.5 are drawn, by using (17) and (18), in Fig. 2. In Fig. 3 illustrates the stability boundary loci in $(KK_cT, KK_cT_d))$ plane for the normalized dead time value of $\tau = 1$ and fixed $KK_c(T^2/T_i)$ values of 1 and 0.5, by the use of (17) and (19).

Also, it is worth mentioning that plotting stability boundary locus for $\omega \in [0, \omega_c]$ will be enough since the controller operates in this frequency range (Tan, 2005). Here, ω_c is the critical frequency value where the Nyquist plot of a plant transfer function intersects the negative real axis, or open loop transfer function phase is equal to -180^o. Therefore, with the help of these graphs, the following four linear equations are obtained by using $KK_c(T^2/T_i)$ and KK_cT_d values corresponding to a constant KK_cT value.

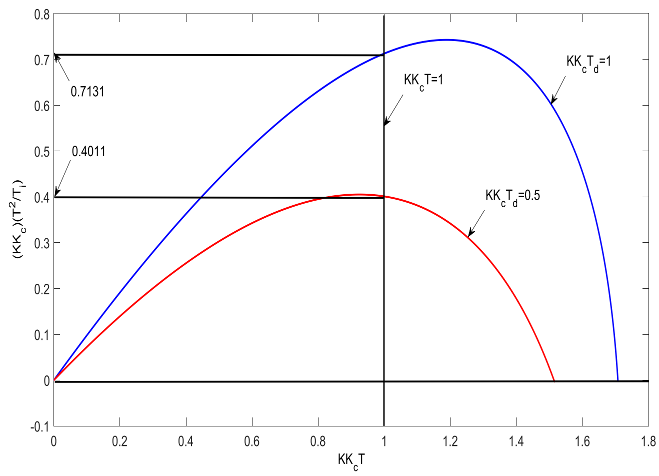

Fig. 2. Stability boundary locus in $\left(KK_cT, KK_c\left(T^2/T_i\right)\right)$ plane for fixed values of KK_cT_d.

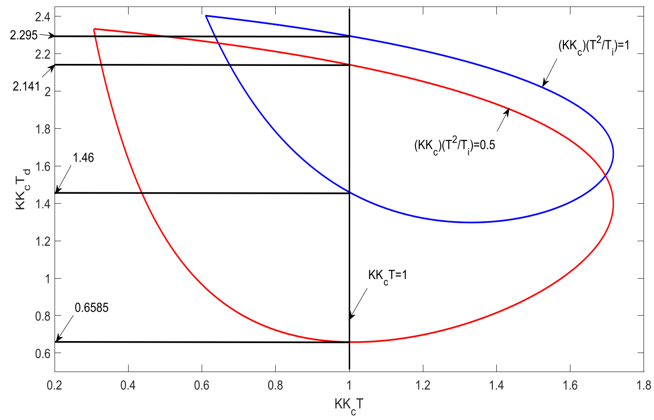

Fig. 3. Stability boundary locus in (KK_cT, KK_cT_d) plane for fixed values of $KK_c(T^2/T_i)$.

$$l_1:\ KK_cT_d = 1.602\left(KK_c\left(T^2/T\right)_i\right) - 0.142,$$

$$l_2:\ KK_c\left(T^2/T\right)_i = 0,$$

$$l_3:\ KK_cT_d = 1.603\left(KK_c\left(T^2/T\right)_i\right) - 0.143,$$

$$l_4:\ KK_cT_d = 0.308\left(KK_c\left(T^2/T_i\right)\right) + 1.987. \tag{20}$$

Using the above obtained linear equations, stability boundary locus in $\left(KK_c(T^2/T_i), KK_cT_d\right)$ plane has been formed for the normalized dead time $\tau = 1$ and fixed value of $KK_cT = 1$. The result is depicted in Fig. 4.

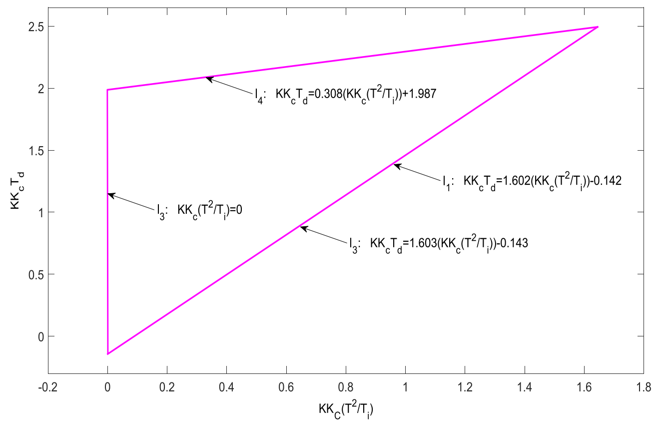

Fig 4. Stability region for $KK_cT = 1$ and $\tau = 1$ in $\left(KK_c\left(T^2/T_i\right), KK_cT_d\right)$ plane.

Similar computations are carried out for normalized dead time values of $\tau = 0.75$, $\tau = 0.5$ and $\tau = 0.25$ so that generalized stability boundary locus are formed in $(KK_c(\mathrm{T}^2/T_i), KK_cT_d))$ plane. Stability boundary loci corresponding to those cases are presented in Fig. 5. The stability boundary loci given in Fig. 5 can be considered as generalized, since, once the IFOPDT model is known, all stabilizing PID controller tuning parameters can be found from Fig. 5 for the fixed value of $KK_cT = 1$ and varying values of normalized dead time. If it is required, stability boundary loci can be plotted for different normalized dead time and KK_cT values. By this way, the approach can be made more generalized.

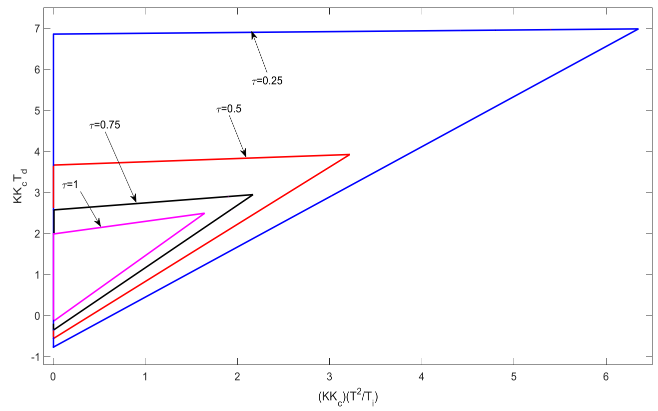

Fig. 5. Stability region in $\left(KK_c\left(T^2/T_i\right), KK_cT_d\right)$ plane for different normalized dead time ratios and $KK_cT=1$.

3. EXAMPLES

3.1 Example 1: Let's consider a process transfer function of $G(s)=e^{-s}/s(s+1)$. The normalized dead time value for this transfer function is $\tau=1$. Since the actual system transfer function exactly matches the IFOPDT model transfer function, the relay feedback identification method (Kaya and Atherton, 2001) will give exact solutions for the IFOPDT model. In Fig. 5, the region remaining inside of $\tau=1$ can be used to determine all stabilizing PID controller tuning parameters. Some points taken from the stability region corresponding to $\tau=1$ and the resultant PID tuning parameters are summarized in Table 1. Note that the controller gain $K_c=1$ in all cases, as $K=1$, $T=1$ for this example. Fig. 6, shows the unit step responses of the closed loop system for the determined PID controllers. The figure proves the validity of the obtained stability region.

Table 1. Some calculated tuning parameters for example 1

case	Selected points		Calculated tuning parameters	
	$KK_c\left(T^2/T_i\right)$	KK_cT_d	T_i	T_d
a	0.2	1.2	5	1.2
b	0.4	1.4	2.5	1.4
c	0.6	1.6	1.66	1.6
d	0.8	1.8	1.25	1.8
e	1	2	1	2
f	1.2	2.2	0.83	2.2

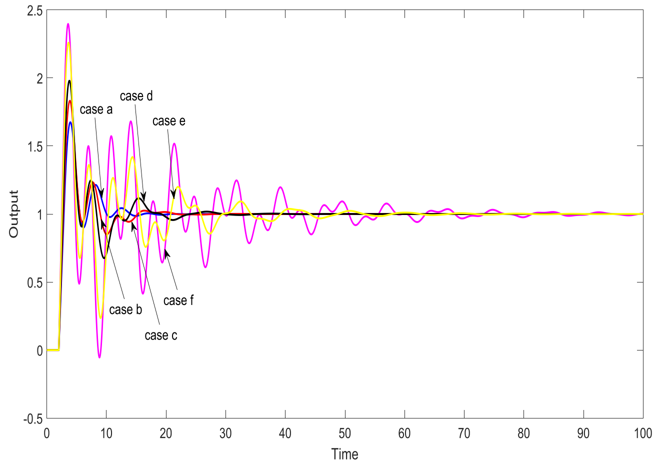

Fig. 6. Step input responses for determined PID controllers for example 1.

3.2 Example 2: In this example, let's take a higher order process transfer function given by $G(s)=e^{-0.2s}/s(0.1s+1)(s+1.2)$. This process transfer function is modelled as IFOPDT model of $G_m(s)=0.843e^{-0.299s}/s(1.072s+1)$ by using relay feedback identification method of Kaya and Atherton (2001). Obtained IFOPDT model transfer function has the normalized dead time of $\tau=0.2789$. Before determining all stabilizing PID controller parameters for this example, it would be appropriate to show the matching between the stability boundary locus of the actual process transfer function and the stability boundary locus of IFOPDT model transfer function. This matching is shown in Fig. 7. As it is seen, a very close matching has been achieved and the stability boundary locus obtained by the actual process transfer function includes the stability boundary locus obtained by the IFOPDT model transfer function. This is a general case observed from many different experiences. This means that the PID controller tuning parameters which are determined by each point taken from the corresponding stability boundary locus will make the system stable.

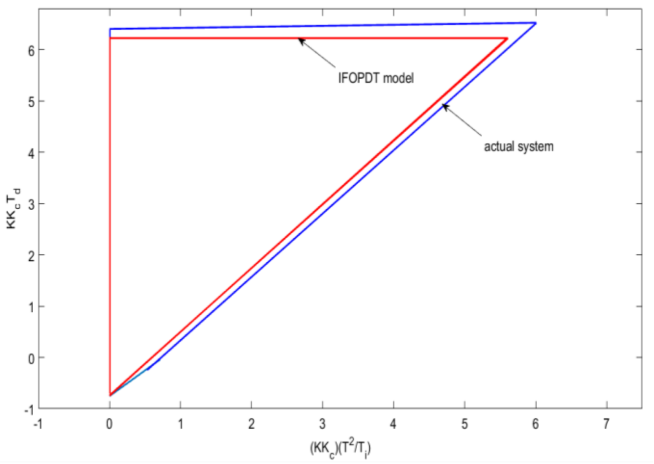

Fig. 7. Stability regions for actual system and IFOPDT model transfer function of example 2.

So, the stability region obtained in Fig. 5 for the value of $\tau = 0.25$, which is the closest value to the normalized dead time value of the IFOPDT model transfer function, is used to determine the PID controller tuning parameters. Table 2 summarizes the results for this example. In this example, the controller gain $K_c = 1.106$ in all cases, as $K = 0.843$, $T = 1.072$. In Fig. 8, unit step responses are given for the determined PID controller parameter values. Again, the validity of the design approach has been verified.

Table 2. Some calculated tuning parameters for example 2

case	Selected points		Calculated tuning parameters	
	$KK_c(T^2/T_i)$	KK_cT_d	T_i	T_d
a	0.5	1	2.144	1.072
b	1.5	3	0.714	3.216
c	2	4.2	0.536	4.502
d	2.5	4.5	0.428	4.824
e	1	5	1.072	5.36
f	3.5	5.2	0.306	5.574

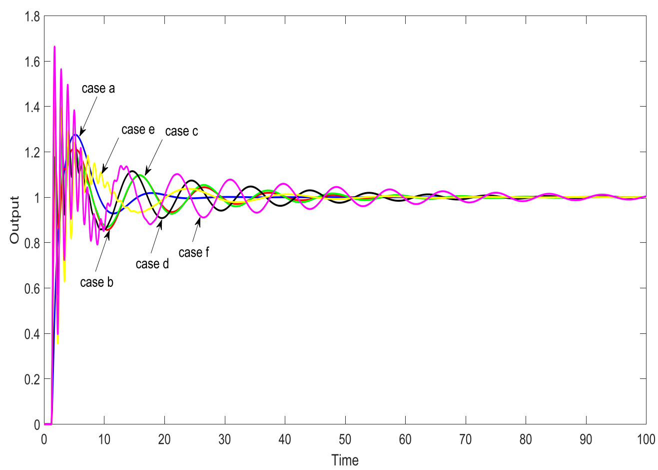

Fig. 8. Step input responses for determined PID controller parameter values for example 2.

3.3 Example 3: In this example, another high order transfer function of $G(s) = e^{-0.5s}/s(s+1)(0.5s+1)(0.2s+1(0.1s+1)$ is studied. This IFOPDT model was obtained as $G_m(s) = e^{-1.123s}/s(1.756s+1)$ by using relay feedback identification method of Kaya and Atherton (2001). IFOPDT model has the normalized dead time of $\tau = 0.6395$. In Fig. 5, the stability region for the normalized dead time of $\tau = 0.75$ can be used to determine all stabilizing PID controllers. The points and corresponding PID controller parameters taken from the inside of stability region corresponding to the normalized dead time of $\tau = 0.6395$ are given in Table 3. Since $K = 1$ and $T = 1.756$ for this example, hence the controller gain $K_c = 0.569$ in all cases. Fig. 9 illustrates unit step responses for designed PID controllers. The validity of the approach has been confirmed once again.

Table 3. Some calculated tuning parameters for example 3

case	Selected points		Calculated tuning parameters	
	$KK_c(T^2/T_i)$	KK_cT_d	T_i	T_d
a	0.25	0.5	7.024	0.878
b	0.6	1.1	2.926	1.931
c	0.4	2.2	4.39	3.863
d	0.9	2	1.951	3.512
e	1.2	1.7	1.463	2.985
f	1.5	2.5	1.17	4.39

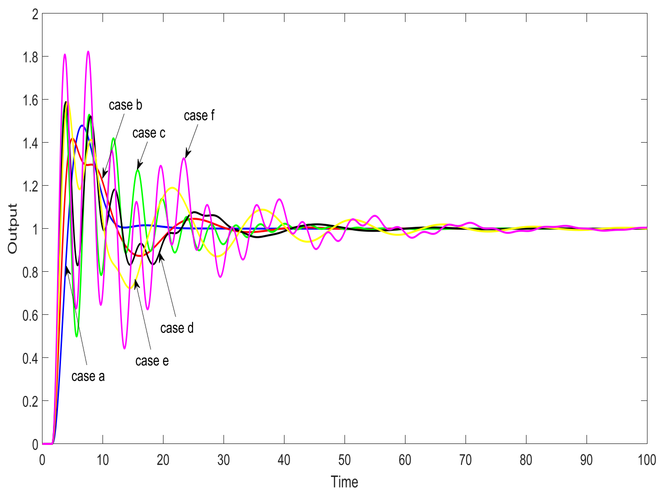

Fig. 9. Step input responses for determined PID controller parameter values for example 3.

4. CONCLUSIONS

In this study, a generalized method has been given for determining all stabilizing PID controllers for stability of integrating plus time delay processes. In order to implement the method, the IFOPDT model of the actual process transfer function has to be obtained. If the actual process and the IFOPDT model transfer functions matches exactly, then obtained stability regions will give exact solutions. If the actual process transfer function is a high-order transfer one, there will be a small mismatch between the stability regions obtained from the actual model transfer functions, but this will not cause any serious problem because it has been shown that the stability boundary locus of the IFOPDT model process transfer function always lies inside the stability boundary locus of the actual transfer function. Thus, the proposed approach removes the necessity of redrawing the stability boundary locus each time as the process transfer function changes.

ACKNOWLEDGEMENT

This work is supported by Dicle University Coordinator-ship of Scientific Research Projects (DUBAP) under Grant no. MUHENDISLIK.16.007.

REFERENCES

Aström, K. J. and Hagglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9, 1163-1175.

Ziegler, J.G. and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64, 759–768.

Cohen, G.H. and Coon, G.A. (1953). Theoretical considerations of retarded control. *Transactions of ASME*, 75, 827-834.

Åström, K. J. and Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica,* 20, 645–651.

Zhuang, M. and Atherton, D. P. (1993). Automatic tuning of optimum PID controllers. *IEE Proceedings-D: Control Theory Applications*, 140, 216-224.

Morari, M. and Zafiriou, E. (1989). *Robust process control*, Englewood Cliffs: Prentice-Hall.

Smith, C.A. and Corripio, A.B. (1997). *Principles and Practice of Automatic Process Control*, John Wiley & Sons, New York.

Ho M.T., Datta A. and Bhattacharyya S.P. (1996). A new approach to feedback stabilization. *Proceedings of the 35th CDC*, 4, 4643–4648.

Ho M.T., Datta A. and Bhattacharyya S.P. (1997a). A linear programming characterization of all stabilizing PID controllers. *Proceedings of American Control Conference*, 6, 3922-3928.

Ho M.T., Datta A. and Bhattacharyya S.P. (1997b). A new approach to feedback design Part I: Generalized interlacing and proportional control, Department of Electrical Engineering, Texas A&M University, College Station, TX, Tech. Report TAMU-ECE97-001-A.

Ho M.T., Datta A. and Bhattacharyya S.P. (1997c). A new approach to feedback design Part II: PI and PID controllers, Dept. of Electrical Eng., Texas A& M Univ., College Station, TX, Tech. Report TAMU-ECE97-001-B.

Munro, N. and Söylemez, M.T. (2000) Fast calculation of stabilizing PID controllers for uncertain parameter systems. *Proceedings of Symposium on Robust Control*, Prague.

Söylemez, M.T., Munro, N. and Baki, H. (2003) Fast calculation of stabilizing PID controllers. *Automatica*, 39, 121- 126.

Shafiei, Z. and Shenton, A. T. (1997). Frequency domain design of PID controllers for stable and unstable systems with time delay. *Automatica*, 33, 2223- 2232.

Huang. Y. J. and Wang, Y. J. (2000). Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem. *ISA Transactions*, 39, 419-43 1.

Tan, N., Kaya, I. and Atherton, D. P. (2003). Computation of stabilizing Pl and PID controllers. *Proceedings of the IEEE International Conference on the Control Applications (CCA2003)*, Istanbul, Turkey. 2003.

Tan, N. (2005). Computation of stabilizing PI and PID controllers for processes with time delay. *ISA Transactions*, 44, 213-223.

Kaya, I. and Atherton, D.P. (2001). Parameter estimation from relay autotuning with asymmetric limit cycle data. *Journal of Process Control*, 11, 429–439.

Kaya, I. and Atiç, S. (2016). PI Controller Design based on Generalized Stability Boundary Locus. *International Conference on System Theory, Control and Computing (ICSTCC),* 20, 24–28.

Zàvackà, J., Bakošovà, M. and Matejičkovà, K. (2013). Robust PI controller design for a continuous stirred tank reactor with multiple steady-states. *International Conference on Process Control (PC)*, 468-473.

Sandeep, D.H. and Yogesh, V.H. (2014). Design of PID controller for inverted pendulum using stability boundary locus. *Annual IEEE India Conference (INDICON)*.

Yogesh, V.H. (2016). PI Controller Design for One Joint Robotic Arm. *International Conference on Control, Automation, Robotics & Vision (ICARCV),* 14.

Deniz F.N., Alagoz B.B., Tan, N. and Atherton D.P. (2016). An integer order approximation method based on stability boundary locus for fractional order derivative/integrator operators. *ISA Transactions,* 62, 154-163.