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

Serdal ATİÇ [1], İbrahim KAYA[2]

[1] Electricity and Energy Department, Vocational High School, Batman University, Batman, Turkey
serdal.atic@batman.edu.tr
[2] Electrical and Electronics Engineering Department, Dicle University, Diyarbakır, Turkey
ikaya@dicle.edu.tr

Abstract

This work represents a new approach for plotting generalized stability boundary loci to achieve all stabilizing PI controllers for integrating processes plus dead time. For this purpose, integrating processes with dead time are modeled by integrating plus first order plus dead time model (IFOPDT). Normalized form of the obtained IFOPDT model and PI controller transfer functions are used together for plotting the stability boundary loci plane. This allows achieving all stabilizing PI controllers from the generalized stability boundary loci plots for processes that can be modeled by an IFOPDT model. The approach eliminates the requirement of plotting the stability boundary locus again and again as the process transfer function changes where this is the case in the literature being followed so far. The effectiveness of the approach is shown by simulation examples.

1. Introduction

Proportional-Integral-Derivative (PID) controllers are well known and commonly being used in industrial applications because of their simple structures, performing robustly for many control problems, and finding its tuning parameters straightforwardly. It has been stated in the literature that in industrial applications almost 90 % of controllers are PID type and most of the time only the PI part is used as the D part is usually not needed [1].

Many different approaches to identify tuning parameters of PID controllers can be found in the literature. The open loop and closed loop tuning methods of Ziegler and Nichols [2], open loop tuning of Cohen and Coon [3] and gain and phase margin based automatic tuning of Aström and Hagglund [4] can be mentioned as the most popular ones. Integral performance criteria based tuning suggested by Zhuang and Atherton [5], Internal Model Control (IMC) by Rivera et al. [6] are among other most frequently used methods for calculating PID tuning rules.

Ho et al. [7, 9] suggested a different approach to design all stabilizing PID controllers. Since then, many researchers have been interested in developing different approaches for all stabilizing PID controllers. Since, the approach given by Ho et al. [7, 9] requires sweeping over the controller gain to determine all stabilizing PI and PID controllers, the approach is time taking for computations. Therefore, an alternative method achieving faster calculations of all stabilizing PID controllers was suggested by Munro and Söylemez [10] and Söylemez et al. [11]. All stabilizing controllers based on graphical solutions were given by Shafiei and Shenton [12] and Huang and Wang [13]. Tan et al. [14] and Tan [15] suggested a new and fast approach to compute all stabilizing PI or PID controllers based on stability boundary locus calculation. Nevertheless, all of the above cited works find the stability boundary locus for a given specific plant transfer function. Therefore, as the plant transfer function changes calculations for plotting stability boundary locus has to be repeated.

In this paper a generalized stability boundary locus for achieving all stabilizing PI controllers to control integrating processes plus dead time will be presented. This work is an extension of authors' previously presented work [16]. There the generalized stability boundary locus for designing PI controllers to control open loop stable process with dead time was given. For the procedure, first an integrating plus first order plus dead time (IFOPDT) model must be obtained. Parameters of the IFOPDT model are estimated by using the relay feedback identification method [17]. The relay feedback identification method used for parameter estimation of the IFOPDT model leads to exact solutions under the assumption that there are no measurement errors and disturbances. Normalized forms of the identified IFOPDT model and PI controller are then used to plot stability boundary locus in (KK_cT , $KK_cT(T/T_i)$) plane for a given value of normalized dead time, $\tau = \theta / T$. Here, θ stands for the time delay and T stands for time constant of the IFOPDT model. Suggested procedure makes the procedure of obtaining the all stabilizing PI controllers easier, since the necessity of re-plotting the stability boundary locus as the plant transfer function varies has been removed.

The paper is structured as follows. In the following section equations to plot stability boundary loci for calculating all stabilizing PI controllers for integrating processes plus dead time will be derived. Simulation examples demonstrating the use of presented method are given Section 3, followed by conclusions provided in Section 4.

2. All Stabilizing PI Controller Design for Integrating Processes

This section provides derivation of equations to plot stability boundary locus to be used in finding all stabilizing PI controllers to control integrating processes with time delay. Consider the block diagram of a control system demonstrated in Fig. 1.

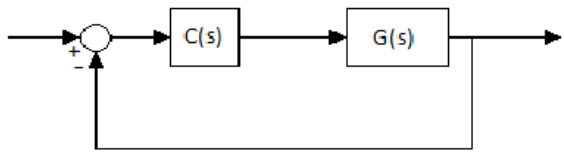

Fig. 1. SISO control system

In the figure, $C(s)$ and $G(s)$ are the controller and the plant transfer functions, respectively. It is assumed that the controller is an ideal PI controller with the following transfer function:

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

The IFOPDT plant transfer function model is assumed to be given by

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

In order to gain the normalized controller and process transfer functions, $Ts = \bar{s}$ was substituted in (1) and (2):

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

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

In order to determine PI controller tuning parameters, closed loop characteristic equation of the system demonstrated in Fig. 1 can easily be shown to be $1 + C(\bar{s})G(\bar{s})$. 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:

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

The numerator and the denominator of (2) have been disintegrated into their even and odd parts and is exchanged in order to achieve

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

For the sake of avoiding the complexity in the equations, the dash over ω will be dropped from now on. Thus, the characteristic equation of the system illustrated in Fig. 1 can straightforwardly be found to be given by the following equation:

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

The following two equations have been calculated by equating the real and imaginary parts of (7) to zero:

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

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

One can define the following equations:

$$\begin{aligned}Q(\omega) &= \omega\sin(\omega\tau), \\ R(\omega) &= \cos(\omega\tau), \\ X(\omega) &= \omega^2,\end{aligned} \tag{10}$$

and

$$\begin{aligned}S(\omega) &= \omega\cos(\omega\tau), \\ U(\omega) &= -\sin(\omega\tau), \\ Y(\omega) &= \omega^3.\end{aligned} \tag{11}$$

Thus, (8) and (9) can be reworked to gain the following equations:

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

The following equations are obtained by solving (10), (11) and (12):

$$KK_cT = \frac{X(\omega)U(\omega) - Y(\omega)R(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)} \tag{13}$$

$$KK_c\frac{T^2}{T_i} = \frac{Y(\omega)Q(\omega) - X(\omega)S(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)} \tag{14}$$

The following two equations are achieved by putting (10) and (11) into (13) and (14):

$$KK_cT = \omega\sin(\omega\tau) + \omega^2\cos(\omega\tau) \tag{15}$$

$$KK_c\frac{T^2}{T_i} = -\omega^3\sin(\omega\tau) + \omega^2\cos(\omega\tau) \tag{16}$$

Consequently, stability boundary locus in (KK_cT , $KK_cT(T/T_i)$) plane for a specified value of normalized dead time, τ , can be plotted using (15) and (16). The use of these two equations eliminates the requirement re-plotting the stability boundary locus again and again as the plant transfer function changes. The relay feedback identification will result in exact parameter estimations when the actual plant transfer function matches the IFOPDT model precisely. Thus, plotted stability boundary locus using the obtained IFOPDT model will not include any approximation. On the other hand, higher order actual plant transfer functions may cause insignificant approximations in the obtained IFOPDT model, particularly near the frequency where the open loop transfer function phase is equal to -180o axis. This frequency is called the critical frequency and shown by ω_c . This approximation for higher order actual plant transfer functions does not lead to any noteworthy problem as it will be shown later by examples.

It would be appropriate to state that controllers work in range of $\omega \in [0, \omega_c]$. Therefore, plotting the stability boundary locus for this range is enough [15]. It is straightforward to show

$$\tan^{-1}\left(\frac{\omega N_o}{N_e}\right) - \tan^{-1}\left(\frac{\omega D_o}{D_e}\right) - \omega\theta = -\pi. \qquad (17)$$

Equations derived in (15) and (16) for the IFOPDT model are general and can be used for plotting generalized stability boundary locus in (KK_cT , $KK_cT(T/T_i)$) plane. Once stability boundary locus for different normalized dead time ratios are obtained, then they can be used for achieving all stabilizing PI controllers. This is performed in Figs. 2, 3 and 4 for different normalized dead time ratios of $0.25 \le \tau \le 1$, $1.25 \le \tau \le 2$ and for $2.25 \le \tau \le 4$. Simulation examples provided in the following section illustrate the use of the suggested approach.

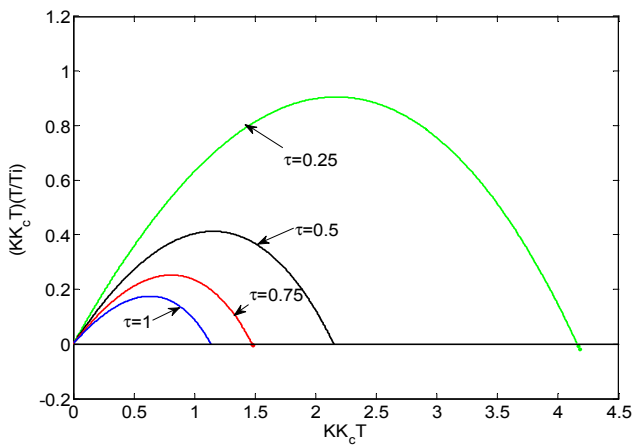

Fig. 2. Stabilizing regions for normalized dead time range of $0.25 \le \tau \le 1$

3. Examples

In this section several examples are provided to show the use of suggested all stabilizing PI controller design approach. In the first example, the actual plant transfer function is chosen to be fitting the IFOPDT model exactly. Therefore, stability regions supplied in Figs. 2, 3 and 4, where it is appropriate according to the normalized dead time ratio, can directly be used to design all stabilizing PI controllers. In example 2 and 3, higher order actual plant transfer functions are assumed. In this case, as the IFOPDT model is not matching with the actual plant transfer function exactly, a recommendation will be given on how to use stability regions given in Figs. 2, 3 and 4.

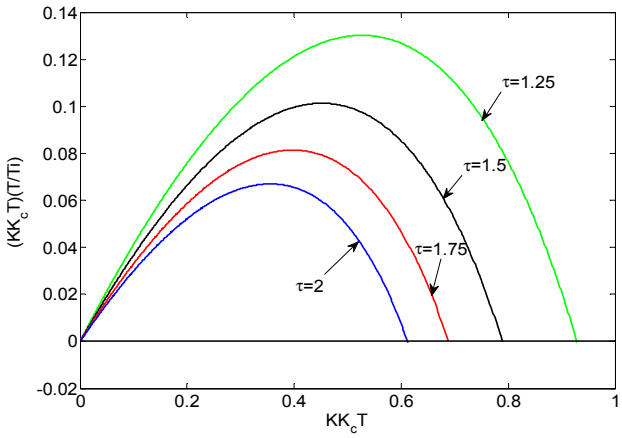

Fig. 3. Stabilizing regions for normalized dead time range of $1.25 \le \tau \le 2$

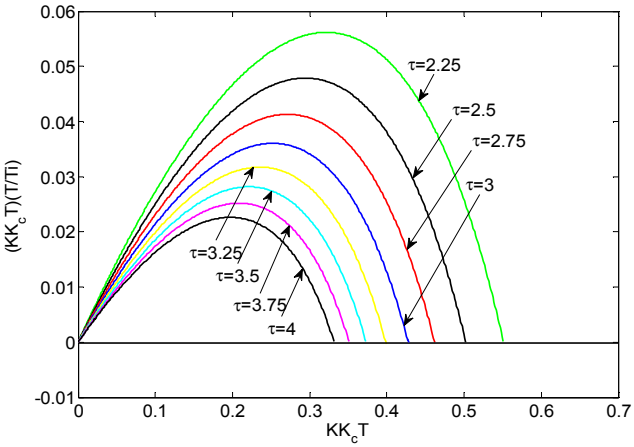

Fig. 4. Stabilizing regions for normalized dead time range of $2.25 \le \tau \le 4$

3.1. Example 1

Consider an actual plant transfer function that is fitting to the IFOPDT model exactly, $G(s) = e^{-2s} / s(s+1)$. The normalized dead time is given by $\tau = 2$. In this case, the actual plant transfer function and the FOPDT model plant transfer matches precisely. Hence, using the relay feedback identification method [17], the IFOPDT model parameters are identified exactly. Therefore, the stability region provided in Fig. 3 marked with blue line (that is, the line corresponding to dead time ratio of $\tau = 2$) is used to determine all stabilizing PI controller tuning parameters. Some points inside the stability region are chosen and the PI tuning parameters corresponding those selected points are calculated. Table 1 summarizes this process. Closed

loop responses for unit step set-point change for calculated PI controllers are illustrated in Fig. 5. It is seen that stable closed loop responses are achieved for all PI controllers where its tuning parameters are chosen from the stability region corresponding to the appropriate dead time ratio, $\tau = 2$ in this case. Also, it is observed from the obtained closed responses that PI controllers determined from the points selected near to the stability region result in quite oscillatory closed loop performances which is expected.

Table 1. Some calculated tuning parameters for example 1

Case	*Selected points*		*Calculated tuning parameters*	
	KK_cT	$(KK_c)(T^2/T_i)$	K_c	T_i
a	*0.1*	*0.005*	*0.1*	*20*
b	*0.2*	*0.018*	*0.2*	*11.111*
c	*0.29*	*0.028*	*0.29*	*10.357*
d	*0.34*	*0.026*	*0.34*	*13.077*
e	*0.4*	*0.024*	*0.4*	*16.667*
f	*0.5*	*0.003*	*0.5*	*166.667*

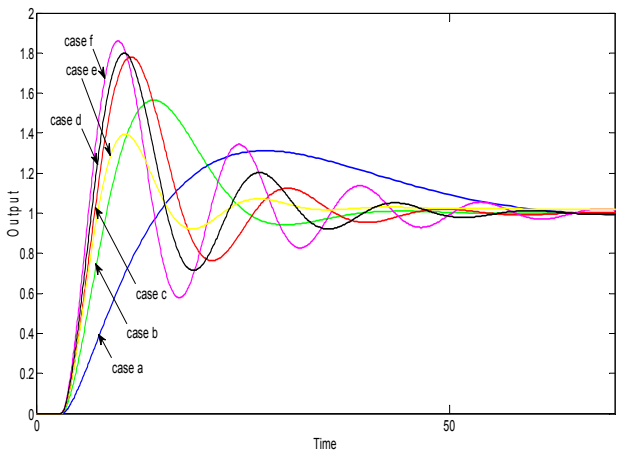

Fig. 5. Unit step-input performances for identified PI controllers for example 1

3.2. Example 2

Here, an actual plant transfer function of $G(s) = e^{-0.2s} / s(0.1s+1)(s+1.2)$, which is an higher order integrating plant transfer function, is considered. The relay feedback identification method [17] was used to identify the IFOPDT model as $G_m(s) = 0.843e^{-0.299s} / s(1.072s+1)$. The normalized dead time is given by $\tau = 0.2789$.

As the actual plant transfer function is higher order, the IFOPDT model plant transfer function will involve approximations. Hence, it would be pertinent to demonstrate the approximation in stability regions of the actual and the achieved IFOPDT model transfer functions. Stability boundary loci of both the actual and IFOPDT model transfer functions are illustrated in Fig. 6.

It is observed from the figure that stability boundary locus obtained from the model always lies inside the actual system boundary locus. This is good because selecting points within the stability region of the obtained IFOPDT model will lead to all stabilizing PI controllers for the actual system. This is a typical characteristic for all cases.

So, the stability boundary locus corresponding to $\tau = 0.25$, which is the nearest dead time ratio to the obtained model dead time ratio, in Fig. 2 can be used to calculate all stabilizing PI controllers. Some selected points and corresponding PI tuning parameters are summarized in Table 2. Unit step set-point change responses for identified PI controllers are shown in Fig. 7 proving the stable closed loop performances.

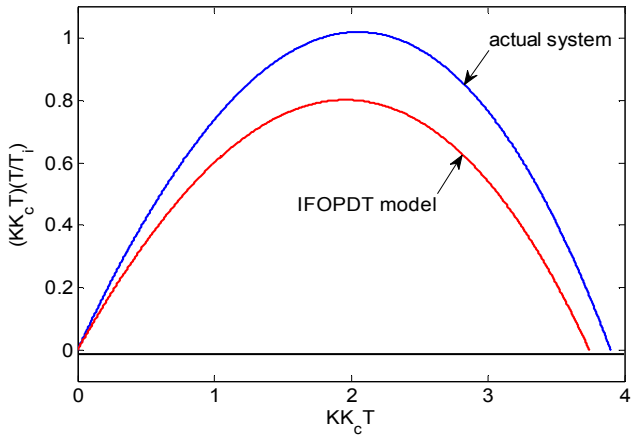

Fig. 6. Stability regions for actual and model plant transfer function for example 2

Table 2. Some calculated tuning parameters for example 2

Case	*Selected points*		*Calculated tuning parameters*	
	KK_cT	$(KK_c)(T^2/T_i)$	K_c	T_i
a	*0.5*	*0.1*	*0.553*	*5.36*
b	*1*	*0.3*	*1.107*	*3.573*
c	*1.5*	*0.4*	*1.660*	*4.020*
d	*2.2*	*0.6*	*2.434*	*3.931*
e	*2.5*	*0.5*	*2.766*	*5.360*
f	*3.2*	*0.2*	*3.541*	*17.152*

3.3. Example 3

In this example, again a higher order integrating process having plant transfer function of $G(s) = e^{-0.5s} / s(s+1)(0.5s+1)(0.2s+1(0.1s+1)$ is considered. The relay feedback identification method [17] was used to identify the IFOPDT model as $G_m(s) = e^{-1.123s} / s(1.756s+1)$. The normalized dead time for this example is given by $\tau = 0.6395$. Stability boundary locus of both the actual and IFOPDT model transfer functions are illustrated in Fig. 7 to show the approximation in stability region obtained by the proposed approach. Again it is seen that the stability region

obtained by using the obtained IFOPDT model always lies in the stability region of the actual plant transfer function. Therefore, the stability boundary locus corresponding to dead time ratio of $\tau = 0.5$ in Fig. 2 can be used to calculate all stabilizing PI controllers.

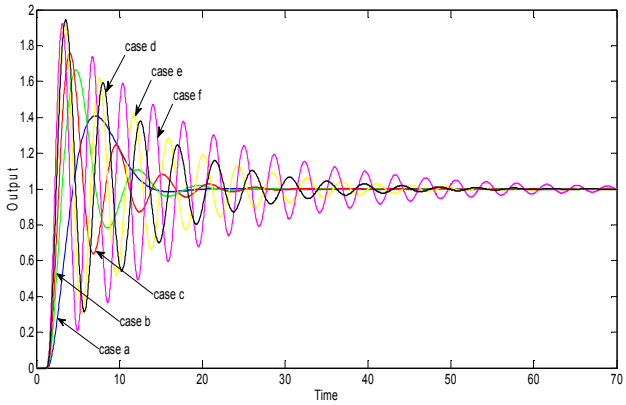

Fig. 7. Unit step-input performances for identified PI controllers for example 2

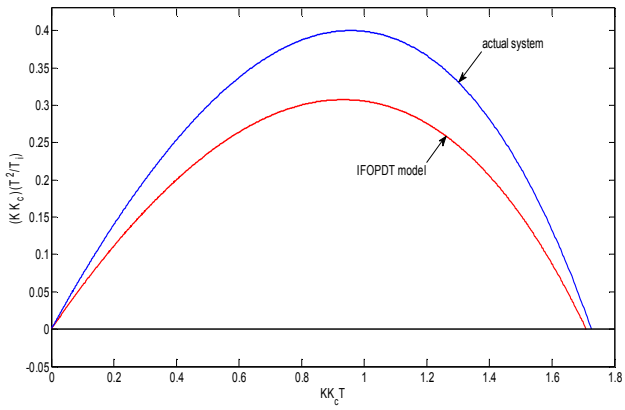

Fig. 7. Stability regions for actual and model plant transfer function of example 3

Some selected points and corresponding PI tuning parameters are summarized in Table 3 for example 3. Unit step set-point change responses for identified PI controllers are shown in Fig. 8 proving that generalized stability locus indeed results in stable closed loop performances.

Table 3. Some calculated tuning parameters for example 3

Case	*Selected points*		*Calculated tuning parameters*	
	KK_cT	$(KK_c)(T^2/T_i)$	K_c	T_i
a	*0.4*	*0.06*	*0.228*	*11.707*
b	*0.7*	*0.1*	*0.399*	*12.292*
c	*1*	*0.2*	*0.569*	*8.780*
d	*1.2*	*0.17*	*0.683*	*12.395*
e	*1.3*	*0.15*	*0.740*	*15.219*
f	*1.5*	*0.05*	*0.854*	*52.680*

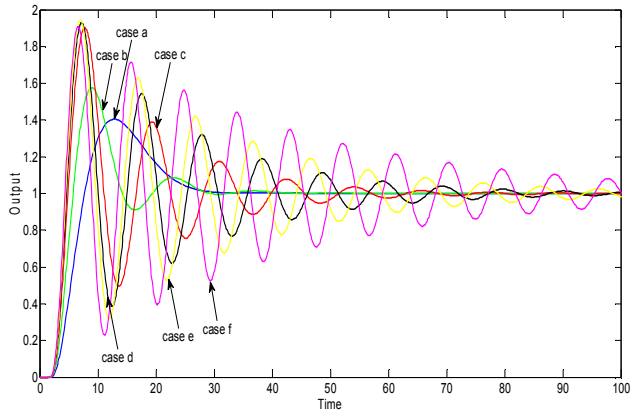

Fig. 8. Unit step-input performances for identified PI controllers for example 3

4. Conclusions

The paper has introduced an approach for plotting generalized boundary locus to calculate all stabilizing PI controllers to control integrating processes with dead time. The approach depends on the IFOPDT model of the actual process. Therefore, if plant transfer function of the actual process matches with IFOPDT model exactly, then obtained boundary locus will result in exact solutions. On the other hand, for higher order plant transfer functions approximate stability regions will be obtained. Nevertheless, it is illustrated that the stability boundary locus obtained from the IFOPDT model always remains in the stability region of the actual system. Hence, no difficulty will be encountered with the use of the proposed approach. Consequently, the proposed approach can be used to eliminate the requirement of re-plotting of stability boundary locus each time as the plant transfer function changes.

5. Acknowledgement

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

6. References

[1] K. J. Åström and T. Hägglund, "The future of PID control," Control Eng. Pract., vol. 9, no. 11, pp. 1163–1175, 2001.

[2] Z. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Trans ASME, vol. 64, pp. 759–768, 1942.

[3] G. A. Cohen, G.H. Coon, "Optimum Settings for Automatic Controllers," Trans. ASME, vol. 64, pp. 758–768, 1942.

[4] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," Automatica, vol. 20, no. 5, pp. 645–651, 1984.

[5] *M. Zhuang and D. P. Atherton, "Automatic tuning of optimum PID controllers," Control Theory Appl. IEE Proc. D, vol. 140, no. 3, pp. 216–224, 1993.*

[6] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," Ind. Eng. Chem. Process Des. Dev., vol. 25, no. 1, pp. 252–265, 1986.

[7] M.-T. Ho, A. Datta, and P. Bhattacharyya, “A New Approach to Feedback Stabilization,” in Conference on Decision and Control, 1996, vol. 4, no. 35th, pp. 4643–4648.
[8] M.-T. Ho, A. Datta, and S. P. Bhattacharyya, “A linear programming characterization of all stabilizing PID controllers,” in Proceedings of the 1997 American Control Conference (Cat. No.97CH36041), 1997, pp. 3922–3928 vol.6.
[9] M. T. Ho, A. Datta, and S. P. Bhattacharyya, “A New Approach to Feedback Design Part I: Generalized Interlacing and Proportional Control,” 1997.
[10] N. Munro and M. T. Söylemez, “Fast Calculation of Stabilizing PID Controllers for Uncertain Parameter Systems,” in Symposium on Robust Control, 2000.
[11] M. T. Söylemez, N. Munro, and H. Baki, “Fast calculation of stabilizing PID controllers,” Automatica, vol. 39, no. 1, pp. 121–126, 2003.
[12] Z. Shafiei and A. T. Shenton, “Frequency-domain design of pid controllers for stable and unstable systems with time delay,” Automatica, vol. 33, no. 12, pp. 2223–2232, 1997.
[13] Y. J. Huang and Y.-J. Wang, “Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem,” ISA Trans., vol. 39, no. 4, pp. 419–431, 2000.
[14] N. Tan, I. Kaya, and D. P. Atherton, “Computation of stabilizing PI and PID controllers,” in IEEE Conference on Control Applications - Proceedings, 2003, vol. 2.
[15] N. Tan, “Computation of Stabilizing PI and PID Controllers for Processes with Time Delay,” ISA Trans., vol. 44, no. 2, pp. 213–223, 2005.
[16] I. Kaya and S. Atiç, “PI Controller Design based on Generalized Stability Boundary Locus,” in International Conference on System Theory, Control and Computing (ICSTCC), 2016, pp. 24–28.
[17] I. Kaya, “Parameter Estimation for integrating processes using relay feedback control under static load disturbances,” Ind. Eng. Chem. Res., vol. 45, no. 13, pp. 4726–4731, 2006.