ICIC Express Letters ICIC International ©2020 ISSN 1881-803X
Volume **14**, Number **10**, October **2020** pp. 993–1000

DESIGN OF POSICAST PIDA CONTROLLER USING KITTI'S METHOD

Pittaya Pannil, Teenapong Sungsorn, Thanit Trisuwannawat
Kitti Tirasesth and Prapart Ukakimaparn

Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Bangkok 10520, Thailand
{ pittaya.pa; thanit.tr; prapart.uk }@kmitl.ac.th; nex_teenapong@hotmail.com

Received March 2020; accepted June 2020

Abstract. *This paper proposes a successful technique for designing a PIDA (proportional-integral-derivative-acceleration) controller in both continuous-time and discrete-time frameworks, which provides better transient response specifications in comparison with PID (proportional-integral-derivative) controller for third-order plant. The proposed design technique consists of three major steps. First, the PIDA controller is designed by using Kitti's method based on root locus technique in the control loop. Second, the maximum percentage overshoot can be decreased to satisfy specification by applying the forward controller. Based on these two steps, all desired specifications can be achieved without trial and error method for tuning controller parameters. Lastly, the Posicast controller is simply designed because the controlled system can be approximated as a standard second-order system. The performances of the designed PIDA controller are confirmed through MATLAB simulation results.*

Keywords: Continuous-time/Discrete-time PIDA controllers, Kitti's method, Posicast controller, Third-order plant, Overshoot

1. **Introduction.** The well-known PID (proportional-integral-derivative) controller is widely used in industrial control system because it can be simply designed for second-order plants. However, the PID tuning has generally high percentage overshooting of step response as well as quite difficultly design for third-order plant which are used in many control applications because the order of plant is greater than the number of zeros provided by the PID controller. This is the reason that Dr. Dorf has proposed the new controller structure as the PIDA (proportional-integral-derivative-acceleration) controller [1] by adding the zero in PID controller. Other approach for third-order or higher-order plant, an analytical approach of PID and $n-2$ stage PD as cascade controller was proposed by Dr. Kitti [2]. The design technique called KM (Kitti's method) is based on root-locus technique placing almost zeros provided by the controller (except one zero) to close with the poles of the controlled plant. The remained zero is required to find its location along with the controller gain that satisfies system stability as desired specifications. This design technique is also extended to the PIDA controller design both continuous-time and discrete-time system [3]. In [4], PIDA controller design by using KM and DA (Dorf's approach) has better performance compared to PID controller designed for third-order plant. Recently, a new approach of PIDA controller design by using CF (closed-form) formulas was proposed [5]. In this approach, the system transfer function is formulated in vector-matrix forms, and then the parameters of PIDA controllers can easily find to satisfy the designed specification. The performance of this approach is also confirmed in the designing of a PIDAJ (proportional-integral-derivative-acceleration-jerk) controller

DOI: 10.24507/icicel.14.10.993

for fourth-order plants [6]. However, the controlled systems need to adjust the controller gain from the first designed value which is obtained by these three analytical approaches (DA, KM and CF formula) to satisfy all desired specification and still have an overshoot. This paper aims at controller design which will satisfy all desired specification on the first design value of the controller gain and eliminate the overshoot. The original Posicast control can be applied to achieving a good steady-state performance without overshoot [7]. There are many researches confirming the performance of Posicast control such as the PID $\times$ $(n-2)$ stage PD cascade controllers for unstable nonlinear system [8,9], the design of PID Posicast control for uncertain system [10] and uncertain oscillatory system [11], respectively. Thus, the authors proposed to apply the Posicast PIDA control for the third-order plant to satisfy specification and eliminate overshoot without any adjusting the controller gain or trial and error tuning. The Posicast PIDA controller using KM and the other analytical methods of PIDA controller design are compared in this paper.

The article in this paper is organized into five sections. Methodology of the proposed design for PIDA controller applied to simplified induction motor is included in Section 2 both continuous-time and discrete-time systems. The other design approaches of PIDA controller are presented in Section 3, and the comparison results are shown in Section 4 through MATLAB simulation, respectively. The article is concluded in Section 5.

2. **Methodology.** Figure 1 shows a general architecture of control system. The transfer function of the required PIDA controller can be stated as

$$K(s) = K_p + \frac{K_i}{s} + K_d s + K_a s^2 = \frac{K_i + K_p s + K_d s^2 + K_a s^3}{s} = \frac{K(s+a)(s+b)(s+c)}{s}, \quad (1)$$

where K_p, K_i, K_d, and K_a denote a proportional gain, an integral gain, a derivative gain, and an acceleration gain, while K is controller gain and a, b, c are the zero of PIDA controller respectively. From [2], the original design technique is aimed to satisfy the desired specifications without trial and error in controller tuning procedure. Then, the forward controller is employed to decrease the overshoot that is greater than usual, and the controlled system structure becomes 2-DOF (two degree of freedom) systems. By placing the zeros of the designed PIDA controller in the way of KM, the overall controlled system is approximated as a standard second-order system. Then, the Posicast controller is easily designed by reshaping the reference input from the maximum overshoot M_p which depends on the damping ratio ζ only, and the first peak time t_p as shown in Figure 2 [7].

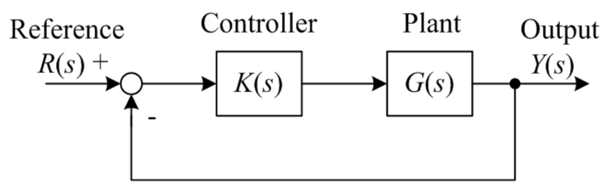


FIGURE 1. General structure of control system

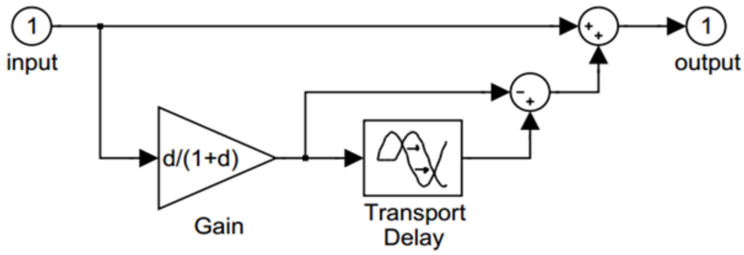


FIGURE 2. SIMULINK diagram

2.1. **Continuous-time framework.** PIDA controller is designed for simplified induction motor position control model that has been implemented in [1], and its transfer function is

$$G(s) = \frac{168.0436}{s\left(s^2 + 25.921s + 168.0436\right)} = \frac{168.0436}{s(s + 12.961 + j0.263)(s + 12.961 - j0.263)}. \quad (2)$$

Then, the open-loop transfer function for the PIDA controllers $K(s)$ and the controlled plant $G(s)$ is obtained as

$$\left\{ K(s)G(s) = K\frac{(s+a)(s+b)(s+c)K_G}{s^2 \cdot (s+p_1)(s+p_2)}, \right. \quad (3)$$

where $K_G = 168.0436$ is a gain of plant, while $p_{1,2} = -12.961 \pm j0.263$ are poles of plant, respectively. By using KM, the zero of the proposed controller as $a, b = -13.061 \pm j0.263$ are firstly assigned close to the pole of the controlled plant p_1, p_2, and then we can find only c and K by using root locus angle and magnitude condition which are expressed as

$$\begin{aligned} &\measuredangle K(s)G(s) = \pm(2k+1)\pi, \quad k = 0, 1, 2, \ldots, \text{ (angle condition)}, \\ &|K(s)G(s)| = 1 \text{ (magnitude condition)}. \end{aligned} \quad (4)$$

The desired specifications for the controller design are usually specified in terms of transient and steady-state response characteristics to a unit-step input, exhibited by a pair of complex-conjugate dominant closed-loop poles $s_{d\pm} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$ as follows:

$$\left\{ \begin{aligned} &\text{Percent Overshoot } (P.O.) = e^{\left(-\zeta\pi/\sqrt{1-\zeta^2}\right)} \times 100\% \le 5\%, \\ &\text{settling time } (t_s) = -\ln\left(0.02\sqrt{1-\zeta^2}\right) \Big/ \zeta\omega_n \le 2 \text{ s}, \ (\pm 2\%). \end{aligned} \right. \quad (5)$$

From computing (5), the damping ratio $\zeta = 0.69$ and undamped natural frequency $\omega_n = 3.068649$ rad/s are obtained. Hence, one of the dominant closed-loop poles is located at $s_d = -2.118 + j2.221$. The open-loop transfer function in (3) without $(s + c)$ at dominant poles s_d and controller gain K is

$$\left\{ K(s_d)G(s_d)|_{without\, z_c=(s+c)} = \frac{(s_d+a)(s_d+b)K_G}{s_d^2(s_d+p_1)(s_d+p_2)} = 18.163\measuredangle 92.518^\circ. \right. \quad (6)$$

Then the angle from the zero c of the proposed controller to dominant poles s_d is

$$\arg[c] = \pi - \arg\left(K(s_d)G(s_d)|_{without\, z_c=(s+c)}\right) = 87.483^\circ. \quad (7)$$

Hence, the location of the zero c of the proposed controller can be obtained as $c = 2.215$ and the controller gain K can calculate from the magnitude condition (4) as follows:

$$K = \frac{|s_d|^2|s_d + 12.961 + j0.263||s_d + 12.961 - j0.263|}{168.0436|s_d + 13.061 + j0.263||s_d + 13.061 - j0.263||s_d + 2.215|} = 0.025. \quad (8)$$

Finally, the PIDA controller transfer function can be expressed as

$$K(s) = \frac{0.025(s + 13.061 + j0.263)(s + 13.061 - j0.263)(s + 2.215)}{s}. \quad (9)$$

Overshoot can be decreased by adding the zero c of the proposed controller to the open-loop transfer function. The followed forward controller is introduced as $K_f(s) = c/(s+c)$ [2] and then the overall system is approximated in standard form of second-order system as

$$\begin{aligned} \frac{Y(s)}{R(s)} &\approx \frac{Kc168.0436}{s^2 + (K168.0436)s + (Kc168.0436)} \approx \frac{9.22}{s^2 + 2 \cdot 0.685 \cdot 3.037s + 9.22} \\ &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \end{aligned} \quad (10)$$

From the response to a unit-step input of a standard second-order system, the maximum overshoot can be expressed as

$$M_p = e^{-\zeta\pi \big/ \sqrt{1-\zeta^2}} = 0.052, \quad \zeta = 0.685. \tag{11}$$

This maximum overshoot occurred at the peak time t_p,

$$t_p = \pi \big/ \omega_n \sqrt{1-\zeta^2} = 1.421 \text{ s}, \quad \omega_n = 3.037 \text{ rad/s}. \tag{12}$$

In order to achieve the response with no overshoot, the unit-step input will be rescaled by the factor $(1 + M_p)$ in two parts as follows:

$$\frac{1}{1+M_p} + \frac{M_p}{1+M_p} e^{-t_p s} = 1 - \frac{M_p}{1+M_p} + \frac{M_p}{1+M_p} e^{-t_p s}. \tag{13}$$

2.2. **Discrete-time framework.** In discrete-time design procedure, once an analog controller in s-domain is obtained, then a digital controller is achieved by controller discretization. There are several ways for mapping from the s-plane to z-plane. The exact conversion between the Laplace and z-plane is $z = e^{sT}$ where T is the sampling time. However, this conversion involves a transcendental function and its transfer function cannot be represented in form of a ratio of the polynomials. This makes it difficult to implement such a control algorithm on a digital computer. Therefore, an approximate conversion will be used instead. The trapezoidal approximation or bilinear transformation is used to approximate in this research by substituting $s = 2/T((z-1)/(z+1))$. Then the analog controller in (9) can be discretized to obtain discrete-time controller $K(z)$ as

$$K(z) = 25480 \frac{\left(z - 0.974 \pm j5.125 \times 10^{-4}\right)(z - 0.996)}{(z-1)(z+1)^2}. \tag{14}$$

The controlled plant in (2) is also discretized to be discrete-time plant $G(z)$ as

$$G(z) = \frac{1.638 \times 10^{-7}(z+1)^3}{(z-1)\left(z - 0.974 \pm j5.121 \times 10^{-4}\right)}. \tag{15}$$

Then, the open-loop transfer function can be approximated as

$$K(z)G(z) \approx \frac{4.174 \times 10^{-3}(z - 0.996)(z+1)}{(z-1)(z-1)}. \tag{16}$$

The closed-loop transfer function before applying the forward controller is

$$\begin{aligned} \frac{Y(z)}{R(z)} &\approx \frac{4.174 \times 10^{-3}(z-0.996)(z+1)}{(z-1)(z-1) + 4.174 \times 10^{-3}(z-0.996)(z+1)} \\ &\approx \frac{4.174 \times 10^{-3}(z-0.996)(z+1)}{1.004z^2 - 2z + 0.996}. \end{aligned} \tag{17}$$

Here, the forward controller is $K_f(z) = 0.0022(z-1)/(z-0.996)$. Then, the overall transfer function for $K_f(z)$ series (17) can be written in standard second-order system as

$$\frac{Y(z)}{R(z)} \approx \frac{9.225 \times 10^{-6}(z+1)^2}{1.004z^2 - 2z + 0.996}. \tag{18}$$

3. **Other Methods.** In this section, an analytic technique in [1], DA, the CF formulas in [6] are considered as follows.

3.1. **Dorf's approach.** There are two characteristic equations to be equated, and the first characteristic equation is

$$(s + r)(s + R)(s + q)\,(s + \hat{q}) = 0, \tag{19}$$

where r, R, q and $\hat{q}$ are desired root locations with specifications based on the design criteria. The second characteristic equation is formed from the nominal control structure of the plant in (2) and controller in (4). Hence, the second characteristic equation can be written as

$$\begin{aligned} F_{\text{actual}}(s) &= 1 + K(s)G(s) = 1 + \frac{K(s+a)(s+b)(s+c)168.0436}{s \cdot s\,(s^2 + 25.921s + 168.0436)} = 0, \\ &= s^4 + [25.921 + 168.0436K]s^3 + [168.0436 + 168.0436K(a+b+c)]s^2 \\ &\quad + [168.0436K(ab+bc+ca)]s + 168.0436Kabc. \end{aligned} \tag{20}$$

For a given q, $\hat{q} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -2.1 \pm j2.0$, $R = -13$ and $r = -30$, then the desired characteristic equation in (20) can be written as

$$F_{\text{desired}}(s) = s^4 + 47.2s^3 + 579s^2 + 1999.6s + 3279.9 = 0. \tag{21}$$

Equating coefficients of the same power between (20) and (21) will obtain $K = 0.1266$, $(a + b + c) = 19.317$, $(ab + bc + ca) = 93.9913$ and $(abc) = 154.1718$ respectively. The remained a, b and c can be obtained by considering the third order polynomial as follows:

$$(s + a)(s + b)(s + c) = s^3 + (a + b + c)s^2 + (ab + bc + ca)s + abc. \tag{22}$$

Then, the designed PIDA controller can be obtained from the roots of (22) as follows:

$$K(s) = \frac{K\left(s^2 + 6.3184s + 11.8605\right)(s + 12.9986)}{s}, \quad K = 0.1266. \tag{23}$$

3.2. **Closed-form formulas.** There are 2 cases for this method: the first case is continuous-time and the second is discrete-time case.

3.2.1. *Continuous-time case.* In this article, the third-order plant $G(s)$ in (2) is controlled by the PIDA controller, and its transfer function is assumed as

$$G(s) = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}, \tag{24}$$

where a_2, a_1, a_0 and b_1, b_0 are known coefficients from the controlled plant $G(s)$.

Here, the actual characteristic equation is as follows:

$$\begin{cases} F_{\text{actual}}(s) = 1 + \dfrac{\left[K_p s + K_i + K_d s^2 + K_a s^3\right](b_1 s + b_0)}{[s]\,(s^3 + a_2 s^2 + a_1 s + a_0)}, \\ \dfrac{F_{\text{actual}}(s)}{(1 + K_a b_1)} = s^4 + \dfrac{(a_2 + K_d b_1 + K_a b_0)}{(1 + K_a b_1)} s^3 + \dfrac{(a_1 + K_p b_1 + K_d b_0)}{(1 + K_a b_1)} s^2 \\ \qquad + \dfrac{(a_0 + K_i b_1 + K_p b_0)}{(1 + K_a b_1)} s + \dfrac{K_i b_0}{(1 + K_a b_1)}. \end{cases} \tag{25}$$

The problem statement of PIDA controller design is to find the parameters K_p, K_i, K_d, and K_a of the controller. The four closed-loop poles are the roots of actual characteristic equation in (25) placing at the locations with exhibiting the output response as desired. The desired locations of these closed-loop poles can be expressed by the characteristic equation as follows:

$$\begin{cases} F_{\text{design}}(s) = \left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)(s + R)(s + r) = 0 \\ \qquad = (s + q)\,(s + \hat{q})\,(s + R)(s + r), \\ \qquad = s^4 + \{\sigma + (r + R)\}\,s^3 + \{\omega_n^2 + \sigma(r + R) + rR\}\,s^2 \\ \qquad\quad + \{\omega_n^2(r + R) + \sigma rR\}\,s + \omega_n^2 rR, \end{cases} \tag{26}$$

where $\sigma = 2\zeta\omega_n$. For the real poles in the factor $(s+R)$ and negligible pole $(s+r)$, the designer chooses these poles in the same region as shown in [1], for example, $R \geq \zeta\omega_n$ and $r \geq R$ to be faster than the poles in quadratic pair $(s^2 + 2\zeta\omega_n + \omega_n^2)$ so that the response is dominated by the second-order characteristic. Equating coefficients of the same power between (25) and (26), then will have the equation which is a simple linear system and can be written in vector-matrix form $Ax = b$ as follows:

$$\begin{cases} A = \begin{bmatrix} 0 & 0 & b_1 & \{b_0 - b_1(\sigma + (r+R))\} \\ b_1 & 0 & b_0 & -b_1\{\omega_n^2 + \sigma(r+R) + rR\} \\ b_0 & b_1 & 0 & -b_1\{\omega_n^2(r+R) + \sigma rR\} \\ 0 & b_0 & 0 & b_1\omega_n^2 rR \end{bmatrix}, \\ b = \begin{bmatrix} -a_2 + (\sigma + (r+R)) \\ -a_1 + \{\omega_n^2 + \sigma(r+R) + rR\} \\ -a_0 + \{\omega_n^2(r+R) + \sigma rR\} \\ \omega_n^2 rR \end{bmatrix}, \\ x^T = \begin{bmatrix} K_p & K_i & K_d & K_a \end{bmatrix}^T. \end{cases} \tag{27}$$

From Section 3.1, variables r, R, q and $\hat{q}$ are desired root locations with specifications based on the design criteria. Hence, the formula for finding K_p, K_i, K_d, and K_a of the PIDA controller is

$$\begin{bmatrix} K_p & K_i & K_d & K_a \end{bmatrix}^T = A^{-1}b. \tag{28}$$

Comparison of controlled plant $G(s)$ in (2) and transfer function $G(s)$ in (24) is expressed as

$$\left\{ G(s) = \frac{168.0436}{s^3 + 25.921s^2 + 168.0436s} \equiv \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}. \right. \tag{29}$$

Substitution a_2, a_1, a_0, b_1, b_0 from (29), $\zeta = 0.69$ and $\omega_n = 3.069$ rad/s from the desired specifications to be designed in (5), $r = 30$ and $R = 13$ from experience in [1], into (27) yields

$$x^T = \begin{bmatrix} 12.2377 & 21.8543 & 2.4605 & 0.1268 \end{bmatrix}^T. \tag{30}$$

Then, the PIDA controller's transfer function designed by using the closed-form formula can be written as follows:

$$K(s) = K_p + \frac{K_i}{s} + K_d s + K_a s^2 = \frac{0.1268(s + 3.2024 \pm j1.7335)(s + 12.9943)}{s}. \tag{31}$$

3.2.2. *Discrete-time system.* By using bilinear transform, the discrete-time PIDA controller is easy to obtain from (31) as follows:

$$\begin{cases} K(z) = \left(K_p + \dfrac{K_i}{s} + K_d s + K_a s^2 \right) \Bigg|_{s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)} = \dfrac{\beta_3 z^3 + \beta_2 z^2 + \beta_1 z + \beta_0}{(z-1)(z+1)(z+1)}, \text{ where} \\ \begin{bmatrix} \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix} = \dfrac{1}{2T^2} \begin{bmatrix} 2T^2 & T^3 & 4T & 8 \\ 2T^2 & 3T^3 & -4T & -24 \\ -2T^2 & 3T^3 & -4T & 24 \\ -2T^2 & T^3 & 4T & -8 \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \\ K_a \end{bmatrix}. \end{cases} \tag{32}$$

To obtain the observer canonical form (OCF), the controller's transfer function can be rewritten as:

$$K(z) = \frac{M(z)}{E(z)} = \frac{Y(z)}{U(z)} = \frac{\beta_3 z^3 + \beta_2 z^2 + \beta_1 z + \beta_0}{z^3 + \alpha_2 z^2 + \alpha_1 z + \alpha_0}. \tag{33}$$

Then, the state-space model for the discrete-time controller is given by

$$\left\{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\alpha_0 \\ 1 & 0 & -\alpha_1 \\ 0 & 1 & -\alpha_2 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} \beta_0 - \alpha_0\beta_3 \\ \beta_1 - \alpha_1\beta_3 \\ \beta_2 - \alpha_2\beta_3 \end{bmatrix} e(k). \right. \quad (34)$$

With the sampling time $T = 1/500$ s/sample, the coefficients vector in (32) can be obtained as follows

$$\begin{aligned} &\begin{bmatrix} \beta_3 & \beta_2 & \beta_1 & \beta_0 \end{bmatrix}^T \\ &= \begin{bmatrix} 1.2927 \times 10^5 & -3.8285 \times 10^5 & 3.7793 \times 10^5 & -1.2435 \times 10^5 \end{bmatrix}^T. \end{aligned} \quad (35)$$

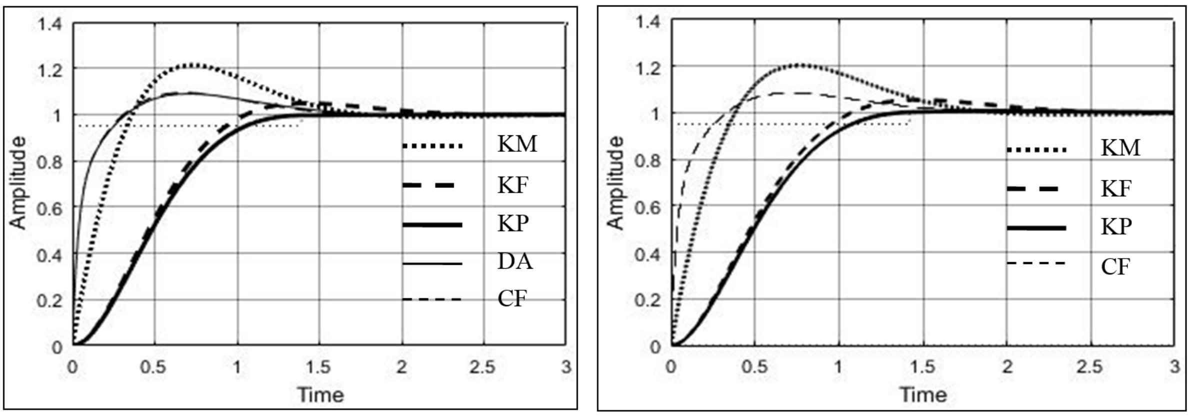
Hence, an alternate form of the discrete-time controller is

$$\left\{\begin{aligned} &K(z) = \frac{K(z-a)(z-b)(z-c)}{(z-1)(z+1)(z+1)}, \\ &K = 1.2927 \times 10^5, \quad c = 0.9749, \quad a, b = 0.9933 \pm j8.202 \times 10^{-4}. \end{aligned}\right. \quad (36)$$

4. **Comparison Results.** The comparative simulation results are shown in Figure 3. For continuous-time system (Figure 3(a)), all design approaches provided settling time less than desired specification but overshoots of KM (thick dot line), DA (thin line) and CF (thin dash line) exceed desired specification. Only KF (KM applying forward controller as thick dash line) and proposed design approach KP (Kitti's method applying forward controller and Posicast as thick line) satisfy all desired specification but KP provided the best result because there is no overshoot from this design approach. For discrete-time system (Figure 3(b)), DA is not applicable for designing in discrete-time system. Only KP and CF provided settling time less than desired specification, but CF has exceeded overshoot compared to desired specification while KP has no overshoot.

Table 1. Performance with different design approaches

	Continuous-time					Discrete-time				
Controller	KM	KF	KP	DA	CF	KM	KF	KP	DA	CF
Gain	0.025	0.025	0.025	0.127	0.127	0.0042	0.0042	0.0042	–	0.0212
T_r (sec.)	0.278	0.862	0.760	0.190	0.184	0.292	0.704	0.778	–	0.190
T_s (sec.)	1.581	1.939	1.825	1.460	1.379	3.050	2.119	1.335	–	1.513
$P.O.$ (%)	21.3	4.9	–	8.9	9.4	20.3	5.5	–	–	8.4



(a) Continuous-time (b) Discrete-time

Figure 3. Comparison results of the unit step response

5. **Conclusions.** The PIDA controller which holds the patent by Dr. Dorf is designed by KM both continuous-time and discrete-time. There are placing the zeros of the controller in the way of this method, and then applying the forward controller. The overall controlled system can be approximated as a standard second-order system and promptly applied the Posicast controller in the last. Comparison with other methods found that only KM, the Posicast controller can be applied both continuous-time and discrete-time system for eliminating an overshoot. However, the performance of KM, DA and CF can be improved to satisfy all desired specification by adjusting the controller gain. There remain opportunities to explore and apply the proposed approach to nth order system. In addition, further modelling of the analysis results for uncertain system is also the further work.

REFERENCES

[1] S. Jung and R. C. Dorf, Analytic PIDA controller design technique for a third order system, *Proc. of the 35th Conference on Decision and Control*, Japan, pp.2513-2518, 1996.

[2] T. Trisuwannawat, K. Tirasesth, J. Ngamwiwit and M. Iida, PID $\times$ $(n-2)$ stage PD cascade controller for SISO systems, *Proc. of the 37th SICE Annual Conference*, Chiba, Japan, pp.965-968, 1998.

[3] A. Tongtanee, P. Pannil, P. Ukakimaparn and T. Trisuwannawat, Discrete-time PIDA controller designed by Kitti's method; A third generation, *Proc. of the SICE Annual Conference*, Kanazawa, Japan, 2017.

[4] D. K. Sambariya and D. Paliwal, Comparative design and analysis of PIDA controller using Kitti's and Jung-Dorf approach for third order practical systems, *British Journal of Mathematics & Computer Science*, vol.16, no.5, pp.1-16, 2016.

[5] N. Narkvitul, P. Ukakimaparn, P. Pannil and T. Trisuwannawat, Closed-form formulas for continuous/discrete-time PIDA controllers' parameters, *The 15th International Conference on Control, Automation and Systems*, pp.326-329, 2015.

[6] P. Ukakimaparn, T. Trisuwannawat, K. Tirasesth and P. Pannil, Closed-form formulas for continuous/discrete-time PIDAJ controller's parameters, *ICIC Express Letters*, vol.13, no.7, pp.645-652, 2019.

[7] J. Y. Hung, Posicast control past and present, *IEEE Multidisciplinary Engineering Education Magazine*, vol.2, no.1, pp.7-11, 2007.

[8] P. Surintramon, P. Pannil, P. Ukakimaparn and T. Trisuwannawat, Posicast PID $\times$ $(n-2)$ stage PD cascade controllers for magnetically-levitation system, *Proc. of the International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 2017.

[9] P. Pannil, K. Pool-em, P. Ukakimaparn and T. Trisuwannawat, Discrete-time Posicast PID $\times$ $(n-2)$ stage PD cascade controllers for unstable system, *ICIC Express Letters, Part B: Applications*, vol.9, no.8, pp.797-804, 2018.

[10] J. Oliveira, P. M. Oliveira, T. M. Pinho and J. Boaventura-Cunha, Swarm-based auto-tuning of PID posicast control for uncertain system, *The 25th Mediterranean Conference on Control and Automation*, Valletta, Malta, 2017.

[11] J. Oliveira, P. M. Oliveira, T. M. Pinho and J. Boaventura-Cunha, PID posicast control for uncertain oscillatory systems: A practical experiment, *International Federation of Automatic Control*, vol.51, no.4, pp.416-421, 2018.