Delay Handling Method in Dominant Pole Placement based PID Controller Design

Saptarshi Das, *Member, IEEE*, Kaushik Halder, and Amitava Gupta

Abstract—Time delay handling is a major challenge in dominant pole placement design due to variable number of poles and zeros arising from the approximation of the delay term. We propose a new theory for continuous time PID controller design using dominant pole placement method mapped on to the discrete time domain with an appropriate choice of the sampling time to convert the delays in to finite number of poles. The method is developed to handle linear systems, represented by second order plus time delay (SOPTD) transfer function models. The proposed method does not contain finite term approximations like various orders of Pade, for handling the time delays which may affect the number and orientation of the resulting poles/zeros. Effectiveness of the proposed method have been shown using numerical simulations on nine SOPTD test-bench processes and another six challenging processes including single, double integrators and process with zero damping.

Index Terms—dominant pole placement, PID controller, SOPTD process, pole-zero matching, Euler's discretization

I. INTRODUCTION

AMONGST various methodologies of PID controller design, the dominant pole placement is quite popular for delay-free systems [1]. Because in this method, the control designers can choose their desired performance (e.g. closed-loop time-constant/frequency, damping ratio) in both continuous [2] and discrete time [3]. However, these methods are difficult to extend for time-delay systems as mentioned in [1], due to having high number of poles and zeros resulting from the Pade approximation of the time delay term (e^{-Ls}). Other contemporary researchers have proposed methods to find regions of closed-loop pole locations inside the unit circle by designing stabilizing discrete-time state-feedback controller using LMI approach [4], [5], [6]. These methods used either with state augmentation [5] or without it [6]. Whereas, only few attempts e.g. [7], [8] have been made to design dominant pole placement PID controller for time delay systems in discrete time. In [7], root locus and modified Nyquist plot

have been used to design guaranteed dominant pole placement based PID controller. In [7], the PID controller parameters $\{K_i, K_d\}$ are calculated depending on K_p which is obtained from Nyquist plot, thus does not have the flexibility of choosing the three controller gains independently. In [9], a matrix method has been used to obtain the sufficient condition for finding dominant pole locations of the discrete-time delayed system, considering parametric uncertainty.

This paper reports the dominant pole placement PID controller design methodology for controlling continuous time SOPTD systems while extending earlier concepts for delay-free systems in [2]. Many process industries involve control problems with both sluggish and oscillatory open loop dynamics as well as different lag to delay ratio and can be modeled using the SOPTD template which makes it a natural choice for considering as a template for a self-regulating process with delay [10]. It is quite challenging to apply the pole placement based PID controller tuning method directly to the SOPTD systems because in the presence of time delay, the corresponding order of the closed loop system becomes infinite under an infinite term Maclaurin series expansion of the exponential term (e^{-Ls}) [11]. The presence of time delays in the characteristic polynomial makes it a quasi-polynomial [12], [13] and thus makes it harder for not only guaranteeing stability but also satisfying user-defined specifications. Under a scenario of truncated series or Pade approximation of the delay term with a chosen order, the number of poles and zeros in the complex s-plane may be arbitrary. To eliminate this increased order problem and handling them with just two PID controller zeros, Wang *et al.* [8] have suggested to map the time delay term in to the discrete time domain, as an integer multiple of the sampling time (T_s) such that the time delay term (e^{-Ls}) becomes multiple concurrent poles at the center of the complex z-plane. In practice, the choice of the sampling time should be sufficiently small unlike the recommendations like $T_s \approx L$ as in [8]. Our proposed method does not impose such restrictions and can easily be extended for large delay systems, because ideally the sampling time should be much smaller than both the system's open loop time constant and delays i.e. $T_s \ll \{L, \tau\}$.

In this paper, we relax the restriction in [8] on the choice of sampling time i.e. $T_s \approx L$, but rather choose T_s such that the delay to sampling time ratio becomes a positive integer i.e. $L/T_s = n, n \in \mathbb{Z}_+$. This allows one to get a fixed number of poles to handle in the complex z-plane for pole placement based PID controller design [14], rather than a variable or

S. Das is with the Department of Mathematics, University of Exeter, Penryn Campus, Cornwall TR10 9FE, U.K. (e-mail: s.das3@exeter.ac.uk).

K. Halder is with the Department of Mechanical Engineering Sciences, University of Surrey, Guildford GU2 7XH, UK (e-mail: k.halder@surrey.ac.uk).

A. Gupta is with the Department of Power Engineering, Jadavpur University, Salt Lake Campus, LB-8, Sector 3, Kolkata-700098, India. (email: amitava.gupta@jadavpuruniversity.in).

even infinite number of poles due to high order Pade approximations [11]. Thus, mapping of the dominant pole placement design in discrete time, transforms the quasi-polynomial in s-domain to a finite term rational polynomial in z-domain [15], [8], which can easily accommodate user's specifications using a coefficient matching method. Therefore, the dominant poles can now be individually mapped between the complex $s \leftrightarrow z$ domain after their locations are determined from the continuous time domain specifications, set by the control designer.

Here, the continuous time PID controller is discretized by using the Euler's method with a choice of sampling time (T_s). The continuous time PID controller cannot be discretized by pole-zero matching method because at $s = 0$ the dc-gain of the PID controller becomes infinite. Next, the coefficient matching method [16], [17] is used to determine PID controller gains $\{K_p, K_i, K_d\}$ for the SOPTD process, by satisfying the user's specifications. The discrete time PID controller which is equivalent to its continuous time version for sufficiently small sampling time, can then be used to control the continuous time SOPTD process. Following the method discussed in [8], our method uses the discretization step to map the time delays as a finite number of poles in the complex z-plane to allow a mapped dominant pole-placement in discrete time. After the controller gains and the pole placement parameters are determined in discrete time domain, they can be brought back to the continuous time domain as such the whole design method was carried out for continuous time SOPTD processes.

II. THEORETICAL FORMULATION

A. Dominant Pole Placement Design in Discrete Time

To design dominant pole placement based discrete PID controller for a SOPTD system as shown in Figure 1, the following steps are to be followed to discretize the continuous time SOPTD processes as in [14]:

Step 1: Transform the poles and zeros form continuous time to discrete time using $z = e^{sT_s}$ and a chosen sampling time T_s such that the time delay $e^{-Ls} = z^{-n}$ where, $L/T_s = n, n \in \mathbb{Z}_+$.

Step2: Zeros/poles at $s = -\infty$ is mapped at $z = 0$ and zeros/poles at $s = 0$ is mapped at $z = 1$.

Step 3: The dc-gain of the open loop system between $s \leftrightarrow z$ domain needs to be matched as:

$$K = G(s)\big|_{s=0} = G(z)\big|_{z=1}. \tag{1}$$

Now, let us consider the open loop SOPTD system in continuous time as

$$G(s) = Ke^{-Ls}/\left(\tau^2 s^2 + 2\zeta_{ol}\tau s + 1\right), \tag{2}$$

where, $\{K, L, \tau, \zeta_{ol}\}$ represents the dc gain, time delay, time constant and damping ratio of the open loop SOPTD system (2) respectively.

The corresponding PID controller in continuous time can be represented as:

$$C(s) = K_p + K_i/s + K_d s, \tag{3}$$

where, $\{K_p, K_i, K_d\}$ represent the proportional and integro-differential gains of continuous time PID controller.

It is seen that the delay-free part of the SOPTD system (2) has two open loop poles in continuous time, located at:

$$s_{1,2} = -\left(\zeta_{ol}/\tau\right) \pm \left(j/\tau\right)\sqrt{1 - \zeta_{ol}^2}. \tag{4}$$

Again (4) can be rewritten using the natural frequency ($\omega_{ol} = 1/\tau$) instead of the time constant as:

$$s_{1,2} = -\zeta_{ol}\omega_{ol} \pm j\omega_{ol}\sqrt{1 - \zeta_{ol}^2}. \tag{5}$$
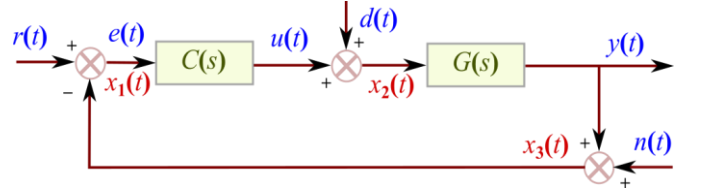

Figure 1: Schematic diagram of the PID controlled SOPTD system.

B. Discretized SOPTD System using Pole-zero Matching

First, we map the continuous-time poles in (5) on to the discrete-time domain with a chosen sampling interval T_s as in [14] which yields:

$$z_1 = e^{s_1 T_s}, z_2 = e^{s_2 T_s}. \tag{6}$$

Using (6) and poles arising from the time delay term in step 1 of the previous sub-section i.e. $e^{-Ls} = z^{-n}$, the corresponding discretized open loop transfer function can be represented as:

$$G(z) = \tilde{K}/\left(z^n\left(z - z_1\right)\left(z - z_2\right)\right), \tag{7}$$

where, $\tilde{K}$ is the equivalent static gain of the open loop discrete time process in continuous time. The value of $\tilde{K}$ can be obtained following the method in [14] as:

$$\left.\frac{K}{\tau^2 s^2 + 2\zeta_{ol}\tau s + 1}e^{-Ls}\right|_{s=0} = \left.\frac{\tilde{K}}{z^n\left(z - z_1\right)\left(z - z_2\right)}\right|_{z=1}, \tag{8}$$

which implies,

$$\tilde{K} = K\left(1 - z_1\right)\left(1 - z_2\right). \tag{9}$$

C. Discrete PID Controller Design using Euler's Method

Here, the continuous time PID controller is transformed into the discretized form by using the Euler's discretization or the backward difference formula given by:

$$s \to \left(1 - z^{-1}\right)/T_s. \tag{10}$$

Substituting (10) in (3), the corresponding discrete time PID controller can be represented as:

$$C(z) = \frac{\left(K_i T_s^2 + K_p T_s + K_d\right)z^2 + \left(-K_p T_s - 2K_d\right)z + K_d}{T_s\left(z^2 - z\right)}. \tag{11}$$

Therefore, the characteristic polynomial for discretized SOPTD process (7) controlled by the discrete PID controller (11) in closed loop can be represented as:

$$1 + C(z)G(z) = 0, \tag{12}$$

which implies,

$$\frac{\left(\begin{array}{c} \left[z^n (z - z_1)(z - z_2)(z^2 - z)T_s \right] + \\ \tilde{K} \left(\begin{array}{c} \left(K_i T_s^2 + K_p T_s + K_d \right) z^2 \\ + \left(-K_p T_s - 2K_d \right) z + K_d \end{array} \right) \end{array} \right)}{z^n (z - z_1)(z - z_2)(z^2 - z)T_s} = 0. \tag{13}$$

This yields the following expression:

$$\left[z^n (z - z_1)(z - z_2)(z^2 - z)T_s \right] + \\ \tilde{K} \left[\left(K_i T_s^2 + K_p T_s + K_d \right) z^2 + \left(-K_p T_s - 2K_d \right) z + K_d \right] = 0 \tag{14}$$

It is evident that (14) has $(n+4)$ number of roots. In fact, the number and orientation of the zeros and poles of the closed loop system can be modulated by a proper choice of the PID controller gains $\{K_p, K_i, K_d\}$ and sampling time (T_s). Now, the PID controller gains $\{K_p, K_i, K_d\}$ can be obtained from the unique expressions in the $(n+4)^{th}$ order polynomial (14). From the closed loop characteristic polynomial (14), it is interesting to note that the PID controller gains $\{K_p, K_i, K_d\}$ are connected only with the terms z^0 to z^2 i.e. the expression with the coefficient $\tilde{K}$ and rest of the coefficients i.e. for z^3 to z^{n+4} do not contain any PID controller parameters. Therefore, in order to obtain the PID controller gains $\{K_d, K_p, K_i\}$ using the coefficient matching method, only the last three coefficients of the expanded polynomial equation i.e. (z^0, z^1, z^2) in (14) needs to be compared with the same coefficient of the desired closed loop characteristic polynomial. To allow dominant pole placement by coefficient matching, we choose the desired closed-loop characteristic polynomial of $(n+4)^{th}$ order considering two specified dominant poles and rest being non-dominant in nature. Here, the non-dominant poles are considered to have three different characteristics i.e. all real, all complex conjugates where the dominant pole placement parameter (m) being connected with either the real or both real/imaginary parts of the non-dominant poles. In order to obtain the dominant pole placement based PID controller, the coefficient matching method has been adopted as in [16], [17], and the results are given next.

D. Expressions for Stabilizing PID Controller Gains

Now, expressions for the discrete PID controller gains $\{K_p, K_i, K_d\}$ are obtained using coefficient matching between the characteristic polynomial of the closed loop system with the given set of open loop process parameters $\{K, L, \tau, \zeta_{ol}, \omega_{ol}\}$ vs. the desired characteristic polynomial with the user-defined

closed loop specifications $\{m, \zeta_{cl}, \omega_{cl}\}$, representing the pole-placement parameter, closed loop damping ratio and frequency respectively [2]. Now, in order to obtain the expressions for the PID controller gains, the desired closed loop characteristic equation should contain two dominant poles satisfying the user's specifications apart from the finite number of non-dominant poles. The number of non-dominant poles in z-domain will depend on the sampling time and hence order of the closed loop system. With a proper selection of the user's specifications $\{m, \zeta_{cl}, \omega_{cl}\}$ and utilizing the analytical expressions, obtained from the coefficient matching for the characteristic polynomial in z-domain, one can find out the PID controller gains $\{K_p, K_i, K_d\}$ that not only stabilizes the closed loop system but also minimizes certain time-domain objective function. The detailed derivations for these cases to obtain the PID controller gains are shown next.

III. DOMINANT POLE PLACEMENT BASED PID CONTROLLER DESIGN IN DISCRETE TIME DOMAIN

The discrete time PID controller structure (11) is used to find out the gains using dominant pole placement method via the coefficient matching method to handle the discrete time system (7), originally obtained from the SOPTD model. The analytical expressions for PID controller gains differs due to the nature (i.e. real/imaginary) and whether the pole-placement parameter m is present in the real/imaginary parts of the non-dominant poles. Derivations for these three cases are given in the next sub-sections.

A. All Non-Dominant Real Poles

In order to ensure dominant pole placement with the discrete PID controller gains $\{K_p, K_i, K_d\}$, the dominant complex conjugate poles ($s_{1,2}^d$) and the non-dominant (s_3^{nd}) real poles of the desired closed loop characteristic polynomial in continuous time are considered as:

$$s_{1,2}^d = \left(-\zeta_{cl}\omega_{cl} \pm j\omega_{cl}\sqrt{1 - \zeta_{cl}^2} \right), \quad s_3^{nd} = -m\zeta_{cl}\omega_{cl}. \tag{15}$$

Now, after mapping of the poles in (15) with sampling time T_s, the z-plane locations of the closed loop dominant and non-dominant poles are:

$$z_1^d = e^{s_1^d T_s}, z_2^d = e^{s_2^d T_s}, \quad z_3^{nd} = e^{s_3^{nd} T_s}. \tag{16}$$

Then using (16), the $(n+4)^{th}$ order discretized desired characteristic polynomial while considering two dominant and rest of all non-dominant real poles can be represented as:

$$\Delta_1(z) = (z - z_1^d)(z - z_2^d)(z - z_3^{nd})^{n+2} = 0. \tag{17}$$

Now using the binomial expansion for the polynomial term in (17) containing the non-dominant poles yields:

$$z^{n+4}\left[\binom{n+2}{0}\left(-z_3^{nd}\right)^0\right]+$$

$$z^{n+3}\left[\binom{n+2}{1}\left(-z_3^{nd}\right)^1-\binom{n+2}{0}\left(z_1^d+z_2^d\right)\left(-z_3^{nd}\right)^0\right]+\cdots+$$

$$z^2\left[\begin{array}{l}\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}\\-\binom{n+2}{n+1}\left(-z_3^{nd}\right)^{n+1}\left(z_1^d+z_2^d\right)\\+\binom{n+2}{n}\left(-z_3^{nd}\right)^n z_1^d z_2^d\end{array}\right]+$$

$$z^1\left[\begin{array}{l}-\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}\left(z_1^d+z_2^d\right)\\+\binom{n+2}{n+1}\left(-z_3^{nd}\right)^{n+1}z_1^d z_2^d\end{array}\right]+$$

$$z^0\left[\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}z_1^d z_2^d\right]=0, \tag{18}$$

where, $\binom{n}{k}={}^nC_k=\dfrac{n!}{k!(n-k)!}$ is the binomial coefficient.

Now, matching the coefficient of the z^0, z^1 and z^2 terms of both the equations (14) and (18) yields the corresponding PID controller gains as:

$$z^0: K_d=\left(\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}z_1^d z_2^d\right)\bigg/\tilde{K}$$

$$z^1: K_p=-\left[\left[\begin{array}{l}-\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}\left(z_1^d+z_2^d\right)\\+\binom{n+2}{n+1}\left(-z_3^{nd}\right)^{n+1}z_1^d z_2^d\end{array}\right]\bigg/\tilde{K}T_s\right]-\frac{2K_d}{T_s}$$

$$z^2: K_i=\left[\left[\begin{array}{l}\binom{n+2}{n+2}\left(-z_3^{nd}\right)^{n+2}\\-\binom{n+2}{n+1}\left(-z_3^{nd}\right)^{n+1}\left(z_1^d+z_2^d\right)\\+\binom{n+2}{n}\left(-z_3^{nd}\right)^n z_1^d z_2^d\end{array}\right]\bigg/\tilde{K}T_s^2\right]-\frac{K_p}{T_s}-\frac{K_d}{T_s^2}. \tag{19}$$

B. All Non-Dominant Complex Conjugate Poles with m in Both Real and Imaginary Parts

In this section, within the dominant pole placement method, all the non-dominant poles are considered as complex conjugates. It is possible only when the order of the discretized closed loop characteristic polynomial (14) is even i.e. $(n/2)\in\mathbb{Z}_+$. In this case, the non-dominant complex conjugate poles can further be described of two types i.e. the pole placement parameter (m) connected with both the real and imaginary part of the complex conjugate non-dominant closed loop pole and the other case being m connected with

the real part only which is described in the next sub-section. In the first case, the non-dominant pole locations are:

$$s_{4,5}^{nd}=m\left(-\zeta_{cl}\omega_{cl}\pm j\omega_{cl}\sqrt{1-\zeta_{cl}^2}\right). \tag{20}$$

Then discrete time equivalent of (20) can be represented as:

$$z_4^{nd}=e^{s_4^{nd}T_s}, z_5^{nd}=e^{s_5^{nd}T_s}. \tag{21}$$

Now using (16) and (21) and also for complex conjugate non-dominant poles, the desired closed loop characteristic polynomial can be represented as:

$$\Delta_2(z)=\left(z-z_1^d\right)\left(z-z_2^d\right)\left(z-z_4^{nd}\right)^{\frac{n+2}{2}}\left(z-z_5^{nd}\right)^{\frac{n+2}{2}}=0. \tag{22}$$

Using binomial expansion, equation (22) yields:

$$z^{n+4}\left[\binom{\frac{n+2}{2}}{0}^2\left(-z_4^{nd}\right)^0\left(-z_5^{nd}\right)^0\right]+$$

$$z^{n+3}\left[\begin{array}{l}-\binom{\frac{n+2}{2}}{0}^2\left(z_1^d+z_2^d\right)\\+\binom{\frac{n+2}{2}}{0}\binom{\frac{n+2}{2}}{1}\left(-z_4^{nd}\right)^0\left(-z_5^{nd}\right)^1\\+\binom{\frac{n+2}{2}}{0}\binom{\frac{n+2}{2}}{1}\left(-z_4^{nd}\right)^1\left(-z_5^{nd}\right)^0\end{array}\right]+\cdots+$$

$$z^2\left[\begin{array}{l}\binom{\frac{n+2}{2}}{\frac{n-2}{2}}\binom{\frac{n+2}{2}}{\frac{n+2}{2}}z_1^d z_2^d\\\times\left(\left(-z_4^{nd}\right)^{\frac{n-2}{2}}\left(-z_5^{nd}\right)^{\frac{n+2}{2}}+\left(-z_4^{nd}\right)^{\frac{n+2}{2}}\left(-z_5^{nd}\right)^{\frac{n-2}{2}}\right)\\+\binom{\frac{n+2}{2}}{\frac{n}{2}}^2 z_1^d z_2^d\left(-z_4^{nd}\right)^{\frac{n}{2}}\left(-z_5^{nd}\right)^{\frac{n}{2}}\\-\binom{\frac{n+2}{2}}{\frac{n}{2}}\binom{\frac{n+2}{2}}{\frac{n+2}{2}}\left(z_1^d+z_2^d\right)\\\times\left(\left(-z_4^{nd}\right)^{\frac{n}{2}}\left(-z_5^{nd}\right)^{\frac{n+2}{2}}+\left(-z_4^{nd}\right)^{\frac{n+2}{2}}\left(-z_5^{nd}\right)^{\frac{n}{2}}\right)\\+\binom{\frac{n+2}{2}}{\frac{n+2}{2}}^2\left(-z_4^{nd}\right)^{\frac{n+2}{2}}\left(-z_5^{nd}\right)^{\frac{n+2}{2}}\end{array}\right]+$$

$$\left[\begin{pmatrix}\dfrac{n+2}{2}\\[4pt]\dfrac{n}{2}\end{pmatrix}\begin{pmatrix}\dfrac{n+2}{2}\\[4pt]\dfrac{n+2}{2}\end{pmatrix}z_1^d z_2^d\right]$$

$$z^1\times\left[\left((-z_4^{nd})^{\frac{n}{2}}(-z_5^{nd})^{\frac{n+2}{2}}+(-z_4^{nd})^{\frac{n+2}{2}}(-z_5^{nd})^{\frac{n}{2}}\right)\right]+$$

$$\left[-\begin{pmatrix}\dfrac{n+2}{2}\\[4pt]\dfrac{n+2}{2}\end{pmatrix}^2(-z_4^{nd})^{\frac{n+2}{2}}(-z_5^{nd})^{\frac{n+2}{2}}(z_1^d+z_2^d)\right]$$

$$z^0\left[\begin{pmatrix}\dfrac{n+2}{2}\\[4pt]\dfrac{n+2}{2}\end{pmatrix}^2 z_1^d z_2^d(-z_4^{nd})^{\frac{n+2}{2}}(-z_5^{nd})^{\frac{n+2}{2}}\right]=0 . \tag{23}$$

The coefficients $(A_j, j=\{n+4, n+3, \cdots, 2, 1, 0\})$ of this characteristic polynomial can easily be calculated from the open loop system parameters $\{K, L, \tau, \zeta_{ol}, \omega_{ol}\}$ and the user-defined specifications $\{m, \zeta_{cl}, \omega_{cl}\}$ which yields:

$$\Delta_2(z)=A_{n+4}z^{n+4}+A_{n+3}z^{n+3}+\cdots+A_2 z^2+A_1 z^1+A_0 z^0=0 . \tag{24}$$

Now matching the coefficient of (14) and (24), the corresponding PID controller gains are obtained as:

$$z^0: K_d=A_0\big/\tilde{K}$$
$$z^1: K_p=-\left(A_1\big/\left(\tilde{K}T_s\right)\right)-\left(2K_d\big/(T_s)\right)$$
$$z^2: K_i=\left(A_2\big/\left(\tilde{K}T_s^2\right)\right)-\left(K_p/T_s\right)-\left(K_d/T_s^2\right) . \tag{25}$$

C. All Non-Dominant Complex Conjugate Poles with m Only in the Real Part

Similar to the treatment in the above sub-sections, here the non-dominant complex conjugate closed loop poles in continuous time can be described as:

$$s_{6,7}^{nd}=-m\zeta_{cl}\omega_{cl}\pm j\omega_{cl}\sqrt{1-\zeta_{cl}^2} . \tag{26}$$

In (26), only real part of the non-dominant complex conjugate pole is dependent on the pole placement parameter (m). Now, its discrete time equivalent poles can be obtained as:

$$z_6^{nd}=e^{s_6^{nd}T_s}, z_7^{nd}=e^{s_7^{nd}T_s} . \tag{27}$$

Now using (16) and (27), the desired closed loop characteristic polynomial can be represented as:

$$\Delta_3(z)=\left(z-z_1^d\right)\left(z-z_2^d\right)\left(z-z_6^{nd}\right)^{\frac{n+2}{2}}\left(z-z_7^{nd}\right)^{\frac{n+2}{2}}=0 . \tag{28}$$

This expression for the characteristic equation is exactly the same as in the previous case (22), although the non-dominant poles are now chosen using (26), instead of (20) which alters only the real part instead of both the real/imaginary parts. This

modifies the coefficients of the characteristic equation (28) although the open-loop and desired closed-loop parameters may have the same value which is represented as:

$$\Delta_3(z)=\tilde{A}_{n+4}z^{n+4}+\tilde{A}_{n+3}z^{n+3}+\cdots+\tilde{A}_2 z^2+\tilde{A}_1 z^1+\tilde{A}_0 z^0=0 . \tag{29}$$

Now using the coefficient matching method similar to the previous case one gets the controller gains as:

$$z^0: K_d=A_0\big/\tilde{K}$$
$$z^1: K_p=-\left(A_1\big/\left(\tilde{K}T_s\right)\right)-\left(2K_d/T_s\right)$$
$$z^2: K_i=\left(A_2\big/\left(\tilde{K}T_s^2\right)\right)-\left(K_p/T_s\right)-\left(K_d/T_s^2\right) . \tag{30}$$

IV. RESULTS AND DISCUSSIONS

A. Optimization Based Specification and Controller Design

In this section, we use nine test-bench processes to show the effectiveness of the proposed methodology while using various expressions for the PID controller gains. Each of the nine test bench processes under consideration has different open-loop characteristics viz. lag-dominant (G_1-G_3), balanced (G_4-G_6) and delay dominant (G_7-G_9) with different L/T ratio i.e. $(L/T)<,=,>1$, which are further divided in three different damping scenarios for representing different open-loop oscillation levels i.e. under-damped, critically-damped, over-damped ($\zeta_{ol}<,=,>1$) respectively [18]. Table 1 describes these realistic process models along with their open-loop parameters e.g. process G_9 represents an HVAC system model between fan speed to the supply air pressure control loop [19]. Now the three different expressions (19), (25) and (30) can be used for obtaining the PID controller gains for a chosen sampling time T_s = 0.01 sec for each test-bench process. However, the choice of the closed loop parameters can vary widely and for certain demanded specifications, it might not be possible to get any feasible solution at all [20], [21]. Therefore, we choose the three closed loop performance parameters $\{m, \zeta_{cl}, \omega_{cl}\}$ simultaneously using a random search and optimization method, in particular the particle swarm optimization (PSO) algorithm due to its well-known capability to quickly search a large parameter space even for complex cost function landscape. This can often be advantageous than employing a rejection sampling algorithm like [18] to filter out only the stabilizing set of specifications and the corresponding controller gains. We use here the integral of squared error (ISE) criteria (31) for the PSO based search of the optimum specification $\theta_{opt}=\{m, \zeta_{cl}, \omega_{cl}\}$ and each feasible point in the search space can also be mapped on to the corresponding PID controller parameter space using the respective analytical expressions given above. Here, the ISE cost function has been considered for a step-change in the set-point and the corresponding responses vary with random selection of the three design parameters within a chosen interval of the search domain:

$$\theta_{opt}=\arg\min_{\theta\in\Theta} J(\theta), J(\theta)=\int_0^T\left(r-y(\theta)\right)^2 dt . \tag{31}$$

The PSO algorithm was run in the Matlab environment with a swarm size of $N_{particle} = 300$. The self and social adjustment weights were chosen as $\phi_p = \phi_g = 1.49$ with an adaptive inertia $W \in [0.1, 1.1]$ chosen in the PSO update equation [22]:

$$\vec{v} = w\vec{v} + \phi_p r_p \left(\vec{p} - \vec{x}\right) + \phi_g r_g \left(\vec{g} - \vec{x}\right), \{r_p, r_g\} \in \mathcal{U}(0,1) \quad (32)$$
$$\vec{x} \leftarrow \vec{x} + \vec{v}$$
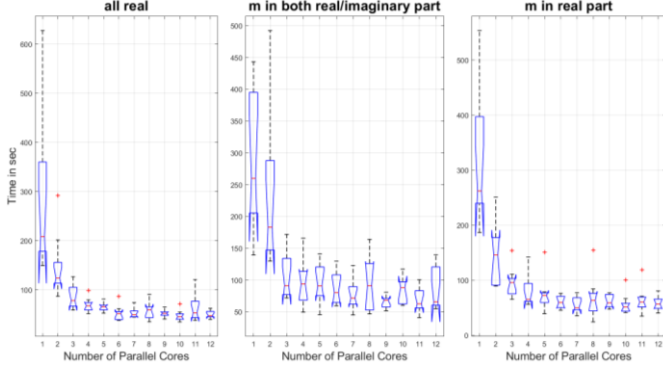
Figure 2: Run-time distribution of the search algorithm on 1-12 core CPU.

TABLE 1:
TEST-BENCH SOPTD PROCESSES AND THE OPEN LOOP PARAMETERS

Process Model	K	L	ζ_{ol}	τ	L/τ
$G_1 = e^{-s}/(9s^2 + 2.4s + 1)$ [18]	1.00	1	0.40	3.00	0.33
$G_2 = e^{-0.8s}/(s^2 + 2s + 1)$ [18]	1.00	0.8	1.00	1.00	0.80
$G_3 = e^{-2s}/(1 + 10s)(1 + 4s)$ [18]	1.00	2	1.11	6.32	0.32
$G_4 = 0.5e^{-s}/(s^2 + 1.2s + 1)$ [18]	0.50	1	0.60	1.00	1.00
$G_5 = e^{-s}/(1 + s)^2$ [18]	1.00	1	1.00	1.00	1.00
$G_6 = e^{-3s}/(9s^2 + 24s + 1)$ [18]	1.00	3	4.00	3.00	1.00
$G_7 = e^{-10s}/(s^2 + 0.5s + 1)$ [23]	1.00	10	0.25	1.00	10.00
$G_8 = e^{-10s}/(s + 1)^2$ [18]	1.00	10	1.00	1.00	10.00
$G_9 = e^{-2s}/(0.12s^2 + 1.33s + 1.24)$ [19]	0.81	2	1.72	0.31	6.43

The simulations were run on a 64-bit Windows PC with 64 GB memory and an AMD Ryzen 7, 3.6 GHz processor where simulations on a single core takes 302 sec $\approx$ 5 min time on an average and the same setup running on 12 parallel cores takes an average run time of 64 sec $\approx$ 1 min. The search ranges for (31) are taken as $\Theta = \{m \in [0, 20], \zeta_{cl} \in [0, 5], \omega_{cl} \in [0, 20]\}$. To avoid search in the infeasible and unstable regions of the parameter space and for highly oscillatory closed loop responses, the penalty method has been adopted where such worse solutions are penalized with a large value of ISE $\approx 10^6$. The adopted stability check includes calculating the real part of closed loop poles without the delay for a randomly guessed specification and PID controller set and lying in the negative half of the s-plane. The scalability of the PSO based parallel search algorithm on 1 to 12 CPU cores have been shown in Figure 2. Such a randomized search not only yields the optimum set of design parameters but also the optimum PID controller gains corresponding to the minimum ISE criteria (31) and has been reported in the supplementary material along with the number of objective function calls (N_{feval}).

B. Control Performance of the Test-bench SOPTD Processes

As shown in Figure 1, for different inputs the following nine transfer functions play a significant role in guaranteeing internal stability and also for evaluating different performance measures of the feedback control loop [24]:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} e \\ u+d \\ y+n \end{bmatrix} = \frac{1}{1+GC} \begin{bmatrix} 1 & -G & -1 \\ C & 1 & -C \\ GC & G & 1 \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}. \quad (33)$$

Amongst these nine, the four transfer functions play a major role to characterize the control system performances [21], [24], [25], i.e. sensitivity $S_e(s)$, complementary sensitivity $T(s)$, disturbance sensitivity $S_d(s)$ and control sensitivity $S_u(s)$ as follows:

$$\begin{aligned} S_e(s) &= 1/(1 + G_{ol}(s)) = G_{re} = G_{dx_2} = G_{nx_3}, \\ T(s) &= G_{ol}(s)/(1 + G_{ol}(s)) = G_{rx_3}, \ S_e(s) + T(s) = 1, \\ S_d(s) &= G(s)/(1 + G_{ol}(s)) = G_{dx_3}, \\ S_u(s) &= C(s)/(1 + G_{ol}(s)) = G_{rx_2}. \end{aligned} \quad (34)$$

However, for standard PID controller structure without a derivative filter, the control sensitivity becomes improper transfer function with more zeros than poles which forbids direct calculation using *step* command in Matlab. As an alternative approach, the control signal can be computed with an impulse input to transfer function S_u/s if the control sensitivity is improper, as in the present case. Rest of the responses in the manipulated (u) and control variables (y) for unit change in the set-point (r) and disturbance inputs (d) can be calculated as:

$$\begin{aligned} y_r(t) &= \mathbb{L}^{-1}(d_1(s)T(s)), d_1(s) = H(s), \\ y_d(t) &= \mathbb{L}^{-1}(d_1(s)S_d(s)), d_1(s) = H(s), \\ u_r(t) &= \mathbb{L}^{-1}(d_2(s)d_1(s)S_u(s)), d_2(s) = \delta(s), \\ u_d(t) &= \mathbb{L}^{-1}(d_1(s)S_e(s)), d_1(s) = H(s). \end{aligned} \quad (35)$$

Here, $H(s) = 1/s$ and $\delta(s) = 1$ represent the Heaviside step function and the Dirac delta function respectively and $\mathbb{L}^{-1}(\cdot)$ represent the inverse Laplace transform operator.

Figure 3-Figure 4 show the time responses of the controlled variable (y) due to a step changes in set-point and disturbance input respectively. The corresponding manipulated variables are shown in Figure 5-Figure 6 respectively, with the three different non-dominant pole types for the nine test-bench processes. These time-domain responses remain almost unchanged when the natures of non-dominant poles are all real and all complex conjugate with m being present in the real part only as shown in Figure 3-Figure 6. More oscillatory time responses are obtained when all non-dominant poles are of complex conjugates type where m is attached with both real and complex parts.
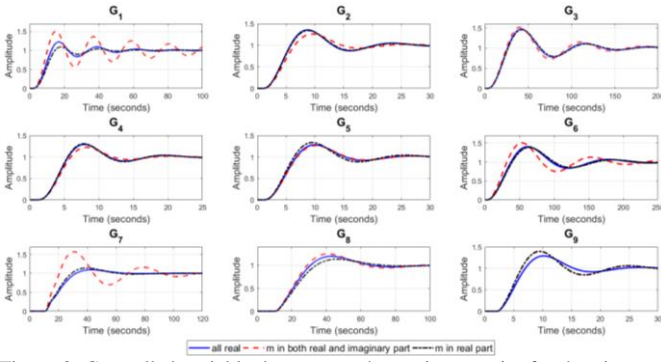
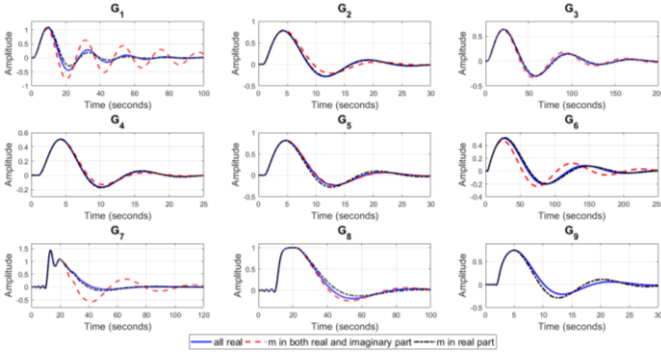Figure 3: Controlled variable due to step change in set-point for the nine test-bench processes G_1-G_9.

Figure 4: Controlled variable due to step change in disturbance input for the nine test-bench processes G_1-G_9.
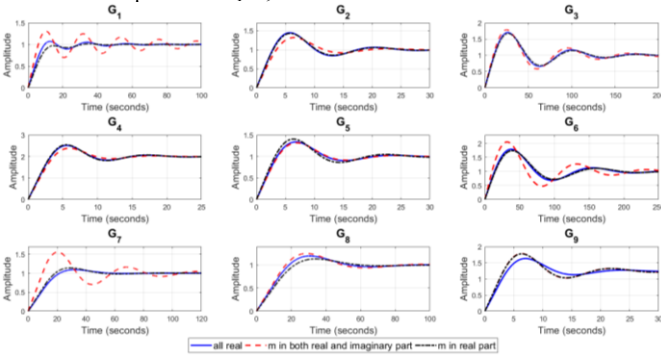

Figure 5: Manipulated variable due to step change in set-point for the nine test-bench processes G_1-G_9.
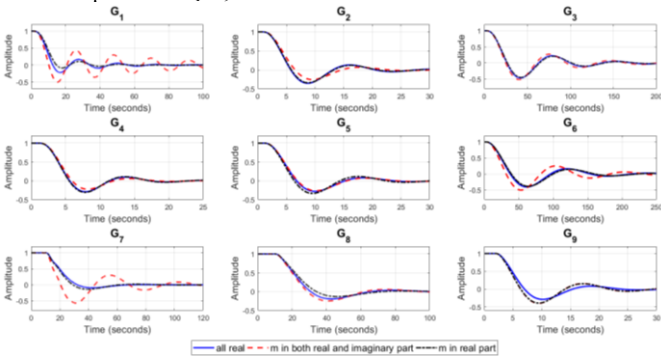

Figure 6: Manipulated variable due to step change in disturbance input for the nine test-bench processes G_1-G_9.

The magnitude plots of sensitivity $|S_e(j\omega)|$ and complementary sensitivity $|T(j\omega)|$ trade-offs [21] can be seen in Figure 7. The sensitivity has high-pass and co-sensitivity has got low-pass characteristics, as expected. It is evident that for the case of m in real/imaginary part has a

better sensitivity response. Conversely, the all real case has better complementary sensitivity response, implying better noise rejection characteristics. Also, the pole type with best sensitivity response has the worst complementary sensitivity response due to their inherent design conflicts [21].
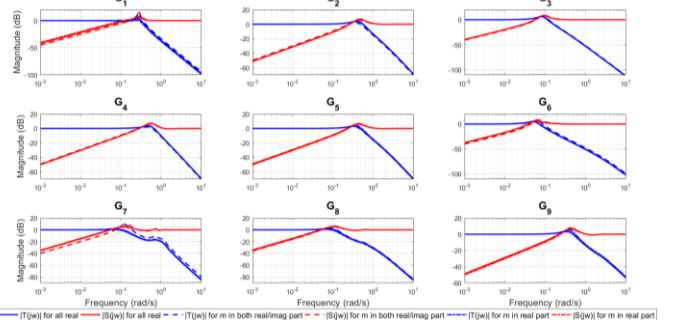
Figure 7: Sensitivity and complementary sensitivity functions using various non-dominant pole types for the nine test-bench processes G_1-G_9.
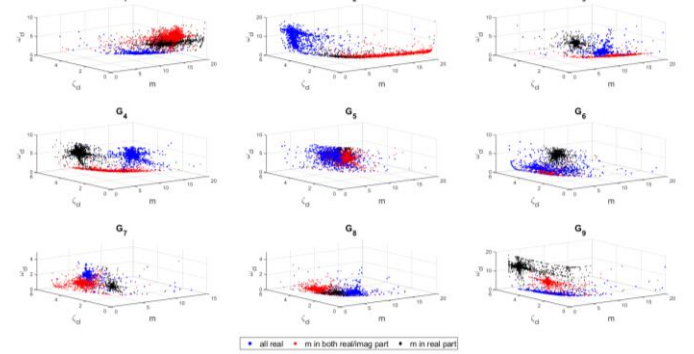

Figure 8: Stability region obtained by PSO based sampling in the three-design parameter space $\{m, \zeta_{cl}, \omega_{cl}\}$ for the nine test-bench processes G_1-G_9.
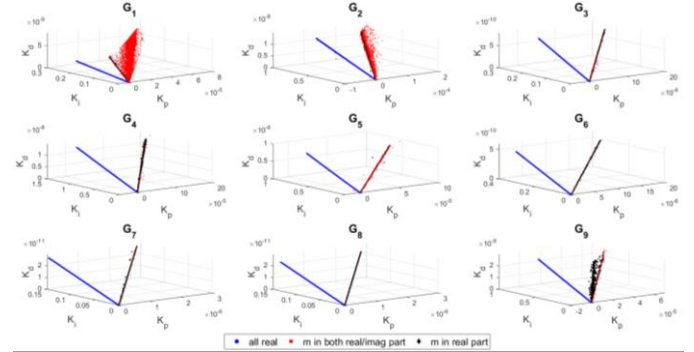

Figure 9: Stability region obtained by PSO based sampling in the 3D space of PID controller parameters $\{K_p, K_i, K_d\}$ for the 9 test-bench processes G_1-G_9.

It is worth noting that although the specifications $\{m, \zeta_{cl}, \omega_{cl}\}$ are different after the optimization as shown in the supplementary material, some of the closed loop performances are quite similar, despite the fact the controller gains are derived from completely different mathematical expressions. During the search process by the PSO algorithm, it explores the entire 3D design parameter space and the feasible solutions are shown in Figure 8 for the three different pole types. Each feasible sampled data-point in the design parameter space can also be mapped on to the corresponding 3D space of PID controller gains as shown in Figure 9, which are widely studied as the stability regions in [12], [13]. It is worth noting that here they take a form of straight line or a plane which can be justified from the expressions for PID gains. Also, depending on the process characteristics and the desired

closed loop specifications, it is possible to get a relatively low value for the derivative and proportional gains compared to the integral gain as observed from Figure 9. For the three expressions of non-dominant pole types and nine test-bench processes, the distribution of the feasible sampled points or the stability region can also be viewed in terms of the ISE values in the joint parameter spaces as shown in Figure 10-Figure 12, where the colorbar represent the ISE value.
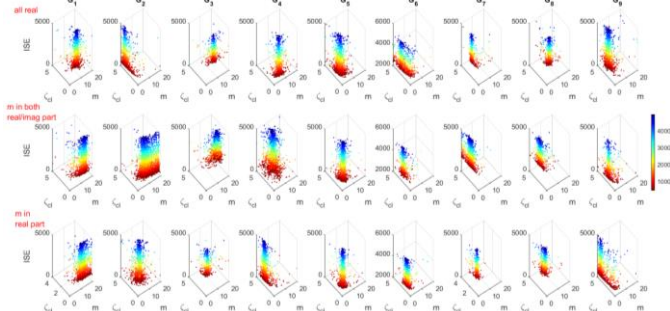

Figure 10: Distributions of ISE as functions of $\{m, \zeta_{cl}\}$ for G_1-G_9.
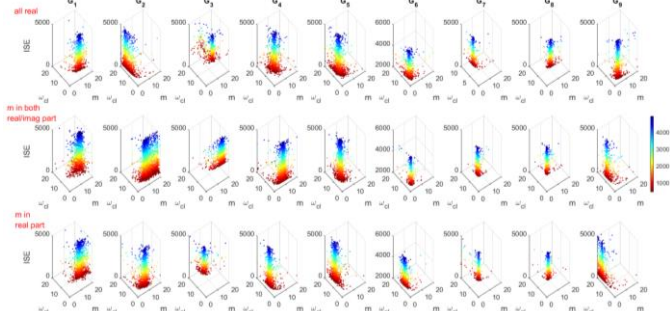
Figure 11: Distributions of ISE as functions of $\{m, \omega_{cl}\}$ for G_1-G_9.
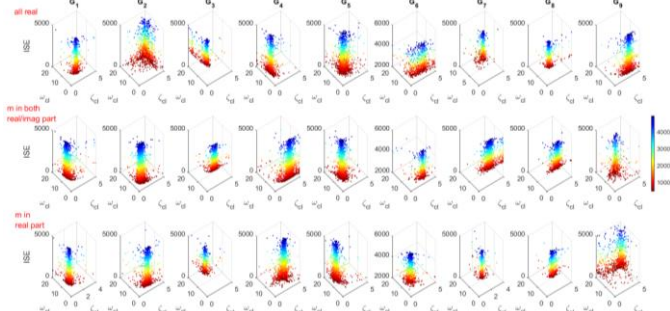

Figure 12: Distributions of ISE as functions of $\{\zeta_{cl}, \omega_{cl}\}$ for G_1-G_9.

TABLE 2:
TEST-BENCH OF DIFFICULT PROCESSES AND THE OPEN LOOP PARAMETERS

Process Model	K	L	ζ_{ol}	τ	L/τ
IPD: G_{10}=e^{-s}/s [26]	10^4	1	5×10^3	1	1
I^{2}PD: G_{11}=e^{-s}/s^2 [26]	10^8	1	1	10^4	10^{-4}
Only delay: G_{12}=e^{-s} [26]	1	1	1	10^{-4}	10^4
FOLIPD: G_{13}=e^{-s}/s(s+1) [27]	10^4	1	50.005	100	0.01
FOPTD: G_{14}=e^{-s}/(1+s) [28]	1	1	50.005	0.01	100
Undamped: G_{15}=e^{-s}/(s^2+1) [29]	1	1	10^{-4}	1	1

C. Restrictions on the SOPTD Process Parameters

Although the method is generic for any SOPTD processes, there is some restrictions on the choice of τ vs. ζ_{ol} as shown in Figure 13, showing that out of the four quadrants, two parameters needs to be both positive or negative which does not include open loop unstable processes. In order to verify this the following few nodes are chosen in the stable region of Figure 13 which also indicates some special cases of the

generalized SOPTD template in terms of nearly zero values (ε) and unit squares on both the system parameters:
1) P, P*, S, S*: represent process with only delay (G_{12}),
2) Q, Q*: represent critically damped SOPTD process with repeated poles (G_2, G_5, G_8),
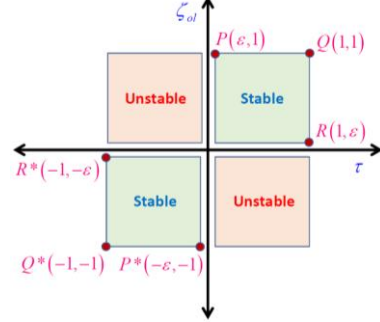3) R, R*: undamped system with delay (G_{15}).

Figure 13: Stable regions in the τ vs. ζ_{ol} plane for the controller design.

D. Extending the Design for Other Difficult Processes

Although the methodology was developed to stabilize SOPTD processes, it is explored in this section that certain classes of difficult processes can also be cast into the SOPTD template (at least asymptotically) for which the same stabilization process can be applied. This typical include several classes of processes with integrators. In order to show the generalization capability of the proposed method, the second test-bench of processes are shown in Table 2.
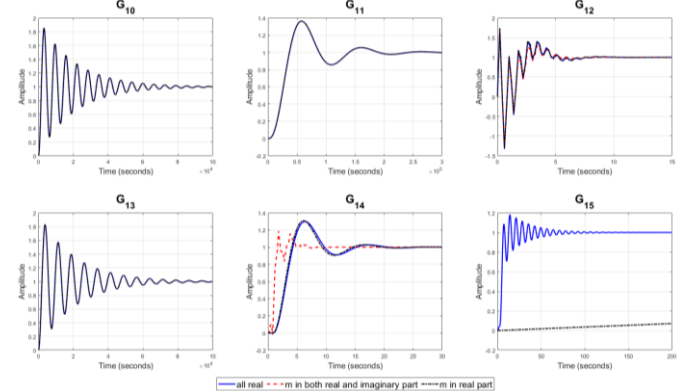

Figure 14: Controlled variable due to step change in set-point for the six-complex test-bench processes G_{10}-G_{15}.

Now using only real poles, the SOPTD process in (2) can also be expressed as:

$$G(s) = \frac{Ke^{-Ls}}{(\tau_1 s+1)(\tau_2 s+1)} = \left(K/\tau_1\tau_2\right)e^{-Ls}\bigg/\left(\left(s+\frac{1}{\tau_1}\right)\left(s+\frac{1}{\tau_2}\right)\right). \tag{36}$$

Therefore, the denominator of (36) becomes:

$$(\tau_1 s+1)(\tau_2 s+1) = \tau_1\tau_2 s^2 + (\tau_1+\tau_2)s + 1. \tag{37}$$

Comparing (37) with the standard form $\left(\tau^2 s^2 + 2\zeta_{ol}\tau s + 1\right)$:

$$\left.\begin{array}{r}\tau_1\tau_2 = \tau^2, \\ \tau_1+\tau_2 = 2\zeta_{ol}\tau\end{array}\right\} \Rightarrow \left.\begin{array}{r}\tau = \sqrt{\tau_1\tau_2}, \\ \zeta_{ol} = (\tau_1+\tau_2)\big/\left(2\sqrt{\tau_1\tau_2}\right)\end{array}\right\}, \tag{38}$$

where, the DC gain of $G(s)$ in (36) is $K = \overline{K}\tau_1\tau_2$ which transforms it in the SOPTD template in (2). For $K = \overline{K}\tau_1\tau_2 = 1$, we get $\overline{K} = 1/\tau_1\tau_2$. Now with suitable choice of τ_1, τ_2 we can extend the design to accommodate more difficult processes as follows:

1) *Integrator Plus Time Delay (IPD):* $G(s) = \overline{K}e^{-Ls}/s$, (39)

with $\tau_1 \to \infty, \tau_2 \to 0$. For implementation purpose one can assume $\tau_1 = 10^4, \tau_2 = 10^{-4}$ which yields:

$$G(s) = (K/\tau_1)e^{-Ls}\bigg/\left(\left(s + \frac{1}{\tau_1}\right)(\tau_2 s + 1)\right) = \frac{\overline{K}e^{-Ls}}{s}, \overline{K} = 1 = K/\tau_1.$$

(40)

2) *Double Integrator Plus Time Delay (I²PD):*
$G(s) = \overline{K}e^{-Ls}/s^2$, (41)

with $\tau_1 \to \infty, \tau_2 \to \infty$. For implementation purpose one can assume $\tau_1 = 10^4, \tau_2 = 10^4$ in (36) which yields the I²PD process (41) with $\overline{K} = 1 = K/\tau_1\tau_2$.

3) *Only Delay:* $G(s) = \overline{K}e^{-Ls}$ (42)

where, $\tau_1 \to 0, \tau_2 \to 0$. For implementation purpose one can assume $\tau_1 = 10^{-4}, \tau_2 = 10^{-4}$ in (36) which yields the only delay process (42) with $\overline{K} = 1 = K$.

4) *First Order Lag with Integral Plus Delay (FOLIPD):*
$G(s) = \overline{K}e^{-Ls}/(s(\tau s + 1))$ (43)

where, $\tau_1 \to \infty, \tau_2 \to \tau$. For implementation purpose one can assume $\tau_1 = 10^4, \tau_2 = \tau$ in (40) which yields the FOLIPD system (43) with $\overline{K} = 1 = K/\tau_1$.

5) *First Order Plus Time Delay (FOPTD):*
$G(s) = \overline{K}e^{-Ls}/(\tau s + 1)$ (44)

where, $\tau_1 \to 0, \tau_2 \to \tau$. For implementation purpose one can assume $\tau_1 = 10^{-4}, \tau_2 = \tau$ in (36) which yields the FOPTD system (44) with $\overline{K} = 1 = K$.

6) *Undamped System with Delay:*
$G(s) = \overline{K}e^{-Ls}/(\tau^2 s^2 + 1)$ (45)

where, $\zeta_{ol} \to 0$. In this case, one can assume $\zeta_{ol} = 10^{-4}$ in (2) which yields the undamped system with delay (45) with $\overline{K} = 1 = K$.

A similar method has been applied to design PID controllers using three pole types and PSO based sampling and the optimal controller parameters are given in the supplementary material. The time domain responses for set-point tracking, disturbance rejection, the control signals are shown in Figure 14-Figure 17 respectively for the six test-bench processes in Table 2. The sensitivity and complementary sensitivity plots are compared in Figure 18 which also shows the superiority of the all real pole-based design compared to the other two for this test-bench of difficult processes with integrators.
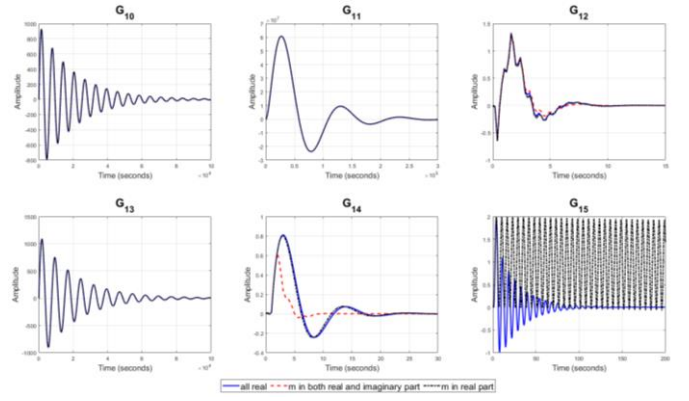

Figure 15: Controlled variable due to step change in disturbance input for the six-complex test-bench processes G_{10}-G_{15}.
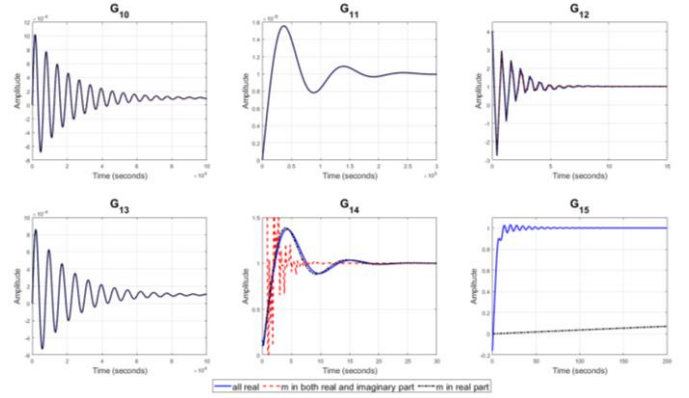

Figure 16: Manipulated variable due to step change in set-point for the six-complex test-bench processes G_{10}-G_{15}.
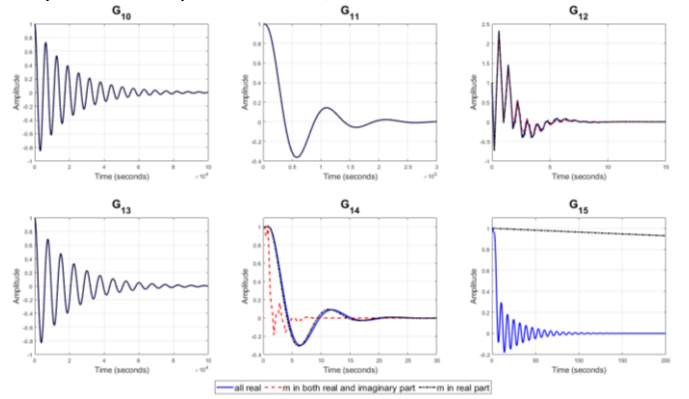

Figure 17: Manipulated variable due to step change in disturbance input for the six-complex test-bench processes G_{10}-G_{15}.
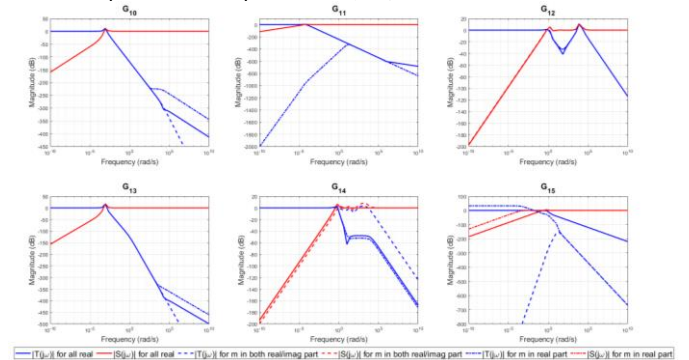

Figure 18: Sensitivity and complementary sensitivity functions using various non-dominant pole types for the nine test-bench processes G_{10}-G_{15}.

The associated stability regions in the 3D design parameter space and the PID controller space are shown in Figure 19 and

Figure 20 respectively which indicates the stability regions lie in different areas of the parameters space and form different patterns, depending on the type of nondominant poles. Also, convergence of the PSO algorithm has been shown in terms of variation in the ISE objective function as three bivariate design parameters for different nondominant pole types. The patterns are found to be more sensitive with respect to the process types compared to the pole types as evident from Figure 21-Figure 23.
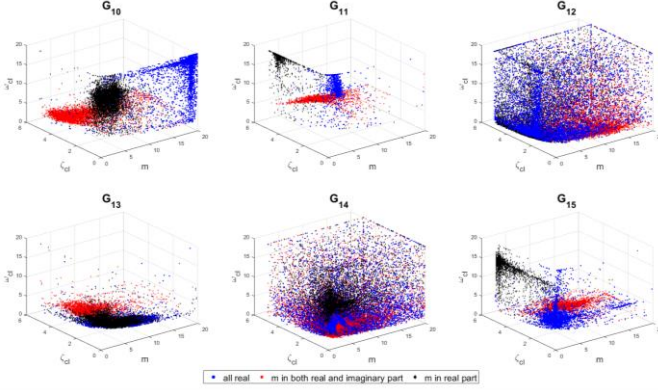

Figure 19: Stability region obtained by PSO based sampling in the three-design parameter space $\{m, \zeta_{cl}, \omega_{cl}\}$ for the test-bench processes G_{10}-G_{15}.
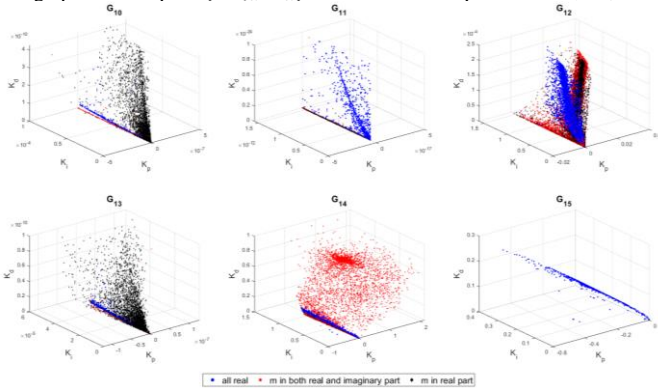

Figure 20: Stability region obtained by PSO based sampling in the three PID controller parameter space $\{K_p, K_i, K_d\}$ for processes G_{10}-G_{15}.

V. NOVELTY OVER EXISTING METHODS OF DELAY HANDLING

To overcome the infinite dimensionality problem of the closed loop system for designing dominant pole placement PID controllers in continuous time domain, this paper first converts the continuous time delay term (e^{-Ls}) to its equivalent discrete time domain (z^{-n}) using a small sampling time (T_s=0.01s). Then it uses the pole-zero matching method to transform the continuous time delay system to its equivalent higher order rational discrete time system. Wang *et al.* [8] have used an approximate pole placement method for infinite dimensional closed loop system in continuous time domain where an empirical equation (consisting of time constant, delay and desired closed loop damping ratio) has been adopted for selecting the sampling time (T_s) and the value of sampling time is greater than the time constant. Therefore, the desired closed loop poles may not be the same as the actual closed loop poles due to the use of large sampling time. Also, the actual closed loop dynamics may be affected by other non-dominant poles in continuous time domain and the first method in [8] does not ensure dominant pole placement with

the requirement of finding the relative pole assignment error and the relative dominance.
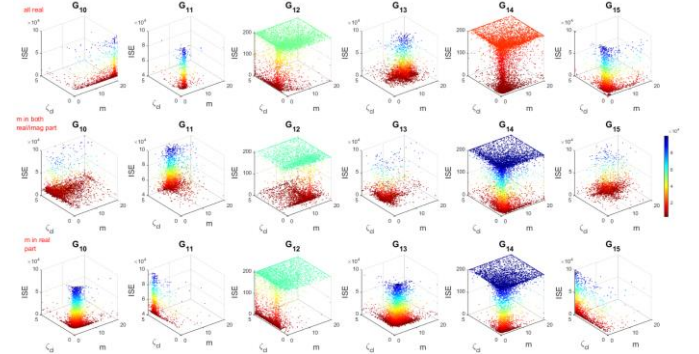

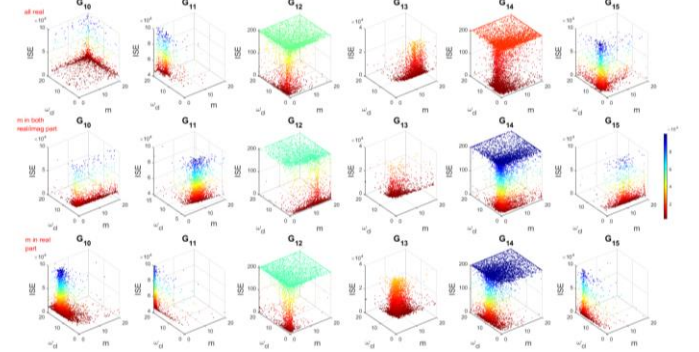Figure 21: Distributions of ISE as functions of $\{m, \zeta_{cl}\}$ for G_{10}-G_{15}.

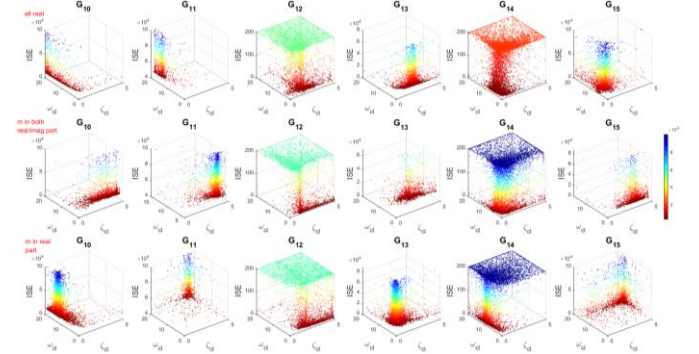Figure 22: Distributions of ISE as functions of $\{m, \omega_{cl}\}$ for G_{10}-G_{15}.

Figure 23: Distributions of ISE as functions of $\{\zeta_{cl}, \omega_{cl}\}$ for G_{10}-G_{15}.

Also, the stability regions should vary with the process characteristics. We have chosen a significantly wide parameter range and the PSO based sampling approach identifies feasible solution space within this domain by rejecting the unstable solutions while drawing more samples and gradually converging towards a region with optimum closed loop performance. This PSO based rejection sampling approach is an alternative to the stability boundary approaches in [13]. The novelty here is to extend the idea in Wang *et al.* [8] by properly selecting the sampling time to map the delay term as finite number of poles in discrete time, so that an analogous pole placement can be achieved in the discrete time domain. Then the resulting controller can be mapped back to the continuous time domain. This technique does not need any approximation for the delay term e.g. 3rd order Pade during the pole placement design as reported in Das *et al.* [18]. We also compare the effect of different non-dominant pole types and stability regions obtained from a sampling approach using PSO optimizer with the ISE performance criteria.

A semi-automated pole placement based state feedback controller has been designed by a continuous pole placement method for handling retarded and neutral type delay systems in [30] and [31]. This method has used shifting the rightmost root of the closed loop system to the far possible left-hand side of the s-plane in quasi-continuous (iterative) way. However, these methods are used for only monitoring the real part of the roots. Michiels *et al.* [32] have proposed another method by combining direct pole placement and minimization of the spectral abscissa for determining controller parameters in retarded time-delay systems. This method also ensures specified control performance where the closed loop system dynamics is considered as nonlinear eigenvalue problem. In [33], [34], dominant pole placement PID controller for time delay systems have been proposed to place three dominant poles for tuning the control loop. This method models the generalized dimensionless representation for PID control loops with delays and uses integral absolute error (IAE) criterion to ensure the control performance.

However, the motivation of this paper is to use the dominant pole placement concept while optimizing for the design specifications with minimum ISE criterion for the SOPTD system. Here, the PSO optimizer is used to efficiently explore the feasible solutions or the stability region within a chosen closed loop parameter space by which one can obtain the three PID controller parameters from the three analytical expressions of different non-dominant pole types. This method is based on the concept of dominant pole placement but may have influence of zero dynamics and other non-dominant poles since only two dominant poles can be specified for controlling a higher order delayed system whose order varies with the sampling time.

VI. CONCLUSION

This paper extends the concept of dominant pole placement based PID controller design to handle time delay processes by mapping the whole design in the discrete time domain. We primarily derive different analytical expressions to control the SOPTD processes using three different non-dominant pole types which are further tested on nine test-bench processes with different relative dead-time and damping characteristics. The proposed method does not employ Pade like finite-term approximations to make the quasi-polynomial characteristic equation in a rational polynomial form to satisfy the design specifications. Using the derived expressions, a random search and optimization using PSO algorithm has been shown to reveal an approximate structure of the feasible parameter space as well as the set of stabilizing PID controller gains. Credible numerical simulations show the efficacy of the proposed design. The proposed method has been tested on a test-bench of SOPTD process models and also more difficult processes that can be asymptotically cast as SOPTD processes, especially with integrators. However, for integrating processes the results are not very optimal, since the mathematical framework does not explicitly consider integrators in to the design. In future, the effect of other discretization methods, performance indices and system/controller pairs may be explored in the context of delay handling within the generalized dominant pole placement framework.

APPENDIX

High-resolution images for the simulation results and additional tables are provided in the supplementary material.

REFERENCES

[1] Q.-G. Wang, Z. Zhang, K. J. Astrom, and L. S. Chek, "Guaranteed dominant pole placement with PID controllers," *Journal of Process Control*, vol. 19, no. 2, pp. 349–352, 2009.

[2] S. Saha, S. Das, S. Das, and A. Gupta, "A conformal mapping based fractional order approach for sub-optimal tuning of PID controllers with guaranteed dominant pole placement," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 9, pp. 3628–3642, 2012.

[3] H. Seraji, "Design of discrete PID controllers for pole placement," in *Decision and Control, 1984. The 23rd IEEE Conference on*, 1984, vol. 23, pp. 1721–1722.

[4] S. Xu, J. Lam, and L. Zhang, "Robust D-stability analysis for uncertain discrete singular systems with state delay," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 4, pp. 551–555, 2002.

[5] V. J. Leite and P. L. Peres, "An improved LMI condition for robust D-stability of uncertain polytopic systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 500–504, 2003.

[6] W.-J. Mao and J. Chu, "D-stability and D-stabilization of linear discrete time-delay systems with polytopic uncertainties," *Automatica*, vol. 45, no. 3, pp. 842–846, 2009.

[7] E. Dincel and M. T. Söylemez, "Guaranteed dominant pole placement with discrete-PID controllers: a modified Nyquist plot approach," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3122–3127, 2014.

[8] Q.-G. Wang, M. Liu, and C. C. Hang, "Approximate pole placement with dominance for continuous delay systems by PID controllers," *The Canadian Journal of Chemical Engineering*, vol. 85, no. 4, pp. 549–557, 2007.

[9] A. F. Ergenc, "A matrix technique for dominant pole placement of discrete-time time-delayed systems," *IFAC Proceedings Volumes*, vol. 45, no. 14, pp. 167–172, 2012.

[10] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*. World Scientific, 2009.

[11] D. Xue, Y. Chen, and D. P. Atherton, *Linear Feedback Control: Analysis and Design with MATLAB*, vol. 14. SIAM, 2007.

[12] G. J. Silva, A. Datta, and S. P. Bhattacharyya, "New results on the synthesis of PID controllers," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 241–252, 2002.

[13] G. J. Silva, A. Datta, and S. P. Bhattacharyya, *PID controllers for time-delay systems*. Springer Science & Business Media, 2007.

[14] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems*, vol. 3. Addison-wesley Menlo Park, 1998.

[15] L. Keel, J. Rego, and S. Bhattacharyya, "Digital PID design for maximally deadbeat and time-delay tolerance," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 361–366, 2002.

[16] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*, vol. 10. 1995.

[17] T. K. Kiong, Q. G. Wang, and C. H. Chang, *Advances in PID control*. Springer, 1999.

[18] S. Das, K. Halder, and A. Gupta, "Performance analysis of robust stable PID controllers using dominant pole placement for SOPTD process models," *Knowledge-Based Systems*, vol. 146, pp. 12–43, 2018.

[19] Q. Bi, W.-J. Cai, Q.-G. Wang, C.-C. Hang, E.-L. Lee, Y. Sun, K.-D. Liu, Y. Zhang, and B. Zou, "Advanced controller auto-tuning and its application in HVAC systems," *Control Engineering Practice*, vol. 8, no. 6, pp. 633–644, 2000.

[20] I. Pan and S. Das, "Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization," *International Journal of Electrical Power & Energy Systems*, vol. 51, pp. 106–118, 2013.

[21] S. Das and I. Pan, "On the Mixed H2/H∞ Loop-Shaping Tradeoffs in Fractional-Order Control of the AVR System," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 1982–1991, 2014.

[22] M. E. H. Pedersen, "Good parameters for particle swarm optimization," *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001*, 2010.

[23] G. Marchetti and C. Scali, "Use of modified relay techniques for the design of model-based controllers for chemical processes," *Industrial & Engineering Chemistry Research*, vol. 39, no. 9, pp. 3325–3334, 2000.

[24] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory*. Courier Corporation, 2013.

[25] A. Herreros, E. Baeyens, and J. R. Perán, "Design of PID-type controllers using multiobjective genetic algorithms," *ISA Transactions*, vol. 41, no. 4, pp. 457–472, 2002.

[26] M. Shamsuzzoha and S. Skogestad, "The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning," *Journal of Process Control*, vol. 20, no. 10, pp. 1220–1234, 2010.

[27] E. Poulin and A. Pomerleau, "PI settings for integrating processes based on ultimate cycle information," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 4, pp. 509–511, 1999.

[28] T. Hägglund and K. J. Åström, "Revisiting the Ziegler-Nichols tuning rules for PI control—part II: the frequency response method," *Asian Journal of Control*, vol. 6, no. 4, pp. 469–482, 2004.

[29] R. C. Panda, C.-C. Yu, and H.-P. Huang, "PID tuning rules for SOPDT systems: Review and some new results," *ISA Transactions*, vol. 43, no. 2, pp. 283–295, 2004.

[30] W. Michiels, K. Engelborghs, P. Vansevenant, and D. Roose, "Continuous pole placement for delay equations," *Automatica*, vol. 38, no. 5, pp. 747–761, 2002.

[31] W. Michiels and T. Vyhlidal, "An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type," *Automatica*, vol. 41, no. 6, pp. 991–998, 2005.

[32] W. Michiels, T. Vyhlidal, and P. Zitek, "Control design for time-delay systems based on quasi-direct pole placement," *Journal of Process Control*, vol. 20, no. 3, pp. 337–343, 2010.

[33] P. Zitek, J. Fiser, and T. Vyhlidal, "Dynamic similarity approach to control system design: delayed PID control loop," *International Journal of Control*, pp. 1–10, 2017.

[34] P. Zitek, J. Fiser, and T. Vyhlidal, "Dimensional analysis approach to dominant three-pole placement in delayed PID control loops," *Journal of Process Control*, vol. 23, no. 8, pp. 1063–1074, 2013.