

PID and PID-like controller design by pole assignment within D-stable regions

*Yongji Wang**, *M. Schinkel*, *Tilmann Schmitt-Hartmann* and *Ken J. Hunt*

Centre for Systems and Control,
Dept. of Mechanical Engineering,
James Watt Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland, U.K.
Tel: (+44) 141 330 4370 Fax: (+44) 141 330 4343
email: {ywang, mschinke, k.hunt}@mech.gla.ac.uk

Abstract

This paper presents a new PID and PID-like controller design method that permits the designer to control the desired dynamic performance of a closed-loop system by first specifying a set of desired D-stable regions in the complex plane and then running a numerical optimisation algorithm to find the controller parameters such that all the roots of the closed-loop system are within the specified regions. This method can be used for stable and unstable plants with high order degree, for plants with time delay, for controller with more than three design parameters, and for various controller configurations. It also allows a unified treatment of the controller design for both continuous and discrete systems. Examples and comparative simulation results are provided to illustrate its merit.

Keywords: PID and PID-like controller, Cascade compensator, D stability, continuous system, discrete system

* Corresponding author. Email: ywang@mech.gla.ac.uk

1. Introduction

Fig. 1 (a) to (d) show some of the most widely used industrial control systems. Fig. 1 (a) has a standard PID controller [1,5,8]. The controller in Fig. 1 (b) stands for a cascade compensator used for an integrating process in order to improve its transient performance [7]. The controller in Fig. 1 (c), here referred to as the **type I** of the modified PID controller, is used in [6] and the feedback is to filter the high-frequency measurement noise. In Fig. 1 (d), the controller structure, referred to as the **type II** of the modified PID controller, is used to improve the regulating performance and the tracking performance [6,12,17].

Since the early work of Ziegler and Nichols [20] on the empirical tuning of PID controller, the development of a systematic approach has been an interesting topic. The motivation behind the research is twofold: First, the majority of industrial control systems still use PID and PID-like controllers. Second, many PID controllers are poorly tuned in practice [15]. It is a common experience that we are not certain which tuning method should be chosen to provide good control to a given process. It would be desirable if there is a design method that works well for a general process model.

Methods developed for a PID or PID-like controller design can be classified in numerous ways according to the plant models. Plant models can be classified into: 1. A general n -th order rational transfer function without time delay [17]; 2. A stable first or second-order transfer function plus time delay [15]; 3. An unstable first or second-order transfer function plus time delay [8]; 4. A general n -th order rational stable or unstable transfer function model with time delay [17]. Many tuning formulas are particularly designed for the simple plant models of types 2 and 3, and therefore are not applicable to the general case. In this paper we will consider the most general plant model, i.e. the 4-th case.

When tuning the controller parameters, certain criteria must be used. The widely-used criteria include the gain and phase margins in *frequency domain* [8], which are particularly suitable for the tuning of the PI-type controller with first or second-order plant models because the two design parameters in a PI controller can be uniquely determined by the two requirements: gain and phase margins [8]. However when a controller with more than two design parameters needs to be designed, the method by the gain and phase margins is not sufficient. With the help of the Nichols chart, the maximum peak resonance and the gain and phase margins were used to tune the PID controller [15]. A similar technique to that used in [15] for the tuning of a PID controller with an integral and a first-order stable plant without time-delay was presented in [13]. This non-symmetrical optimum method is based on the specification of the closed-loop resonant peak and the gain optimum condition. *Time domain performance criteria* have also been used. In [10] Khan et. al. considered the tuning of a PI controller for a simple first-order plant with a large normalised time delay. For this special plant model and PI controller, overshoot and rise time specifications are used. In a recent paper by Tan et. al. [18], the criterion is a desired

closed-loop frequency shape based on H_∞ theory. PID controller is designed for a stable and integrating plant with time delay, which matches the desired shape.

It is known that many factors have effects on the design of a PID and PID-like controller. These include: 1.The number of the controller design parameters; 2.The plant model: stable or unstable, with or without time delay, and plant order; 3.The structure of the control system; 4.The criteria: frequency or time domain specifications; 5.The shape of the D regions. Hence a general synthesis method that fulfils some of the above requirements is desirable.

A region located in the left half of the complex plane is said to be a D-stable region for a continuous system. It is known that the transient performance and stability of a linear time invariant system is mainly determined by its pole distributions, particularly by its dominant poles. The damping ratio and the natural frequency resulting from the dominant poles for a higher order system can be used to determine the boundary of a desired D-stable region within which all the roots must be located. In this paper we propose a novel and unified approach for PID and PID-like controller design. By controlling the roots of a closed-loop system within a set of specified D stability regions, its desired stability and transient performance can be achieved. It is noted that the left half complex plane for continuous systems and the unit circle for discrete systems are two special cases of the general D stability regions used in this paper. This feature makes it possible to design controller for both continuous and discrete systems in a unified way.

The idea of controlling pole distribution is not new [9,19]. For example the root locus method in classical control uses this idea. However it is only applicable to one unknown design parameter. In contrast to the root locus method, the number of the unknown parameters that can be dealt with by the approach developed in this paper may be more than one. Furthermore the optimal search algorithm is faster than the gridding algorithm used in the root locus method.

2. D-stable region, inequality representation, and boundary determination

D-stable region: We first introduce an important concept: **D stability**. In this paper

we use “complex plane” to represent the s -plane for the continuous case and the z -plane for the discrete case. Let D denote the interior of a region, ∂D denote its boundary. For a continuous time system, if a region is located in the left half of the complex plane this region is said to be a D -stable region, and its boundary is said to be a D -stable boundary. Similarly, for a discrete time system if a region is within the open unit circle centred at the origin, this region is said to be a D -stable region, and the boundary is said to be a D -stable boundary. For control system design all the closed-loop poles must be confined to a *desired D -stable region* or a set of disjoint *desired D -stable regions*. We shall say that the closed-loop system is D -stable if all the roots of its characteristic polynomial are located in the desired D -stable regions.

Inequality representation: A distinguished feature that makes the approach developed in this paper different from others is that the D stable region is expressed by inequalities in order to develop efficient numerical algorithms. We call an inequality of the form $g(x) < 0$ a *strict* one, and an inequality of the form $g(x) \leq 0$ a *non-strict* one. Although a region D may have different shapes, it is always bounded by an algebraic curve or a set of piecewise algebraic curves like straight lines, circles, ellipse, hyperbolas, parabolas and so on. Thus the interior D of a region may be described by a set of strict inequalities, its boundary ∂D can be described by a set of equalities, and $D \cup \partial D$ (the union of the interior and the boundary) can be described by a set of non-strict inequalities. A general inequality and equality description of the desired pole regions is (see Fig. 2):

$$D: \quad g_l(\sigma, \omega) < 0, \quad l=1, 2, \dots, m \quad (1)$$

$$\partial D: \quad g_l(\sigma, \omega) = 0, \quad l=1, 2, \dots, m \quad (2)$$

$$D \cup \partial D: \quad g_l(\sigma, \omega) \leq 0, \quad l=1, 2, \dots, m \quad (3)$$

where the complex variable s is $s = \sigma + j\omega$. As a result, the D stable requirement means that all the roots must satisfy (1) or (3), depending on whether the roots are allowed to be on the boundary. We assume that the roots may be allowed to locate on the boundary of the D stability region and we will use the term “inequality” to replace “non-strict inequality” unless otherwise stated. *Obviously a root is located within a D stable region if and only if it satisfies inequalities (3).*

Fig. 3(a) to (f) show some widely-used D -stable regions and their boundaries for continuous systems, and they are represented respectively by the following inequalities:

$$a. \quad (\sigma + a)^2 + \omega^2 \leq R^2 \quad (4a)$$

$$b. \quad a^- \leq \sigma \leq a^+ \text{ and } b^- \leq \omega \leq b^+ \quad (4b)$$

$$c. \quad -(\sigma/a)^2 + (\omega/b)^2 \leq -1 \quad (4c)$$

$$d. \quad \sigma + a \leq 0 \text{ and } R_1^2 \leq (\sigma + a)^2 + \omega^2 \leq R_2^2 \quad (4d)$$

$$e. \quad \sigma + a \leq 0, b\sigma + a\omega \leq 0, \text{ and } b\sigma - a\omega \leq 0 \quad (4e)$$

$$f. \quad \sigma + a \leq 0 \text{ or } (\sigma + a_1)^2 + \omega^2 \leq R^2 \text{ or} \\ (\sigma + a_2)^2 + (\omega + b)^2 \leq R^2 \text{ or } (\sigma + a_2)^2 + (\omega - b)^2 \leq R^2 \quad (4f)$$

Establishment of the boundaries of the desired D stable region: When the approach developed in this paper is used to design a PID or PID-like controller, the D stable region boundary must be specified according to the requirement for the transient performance of a closed-loop system which is determined by its poles and zeros. It is known that it is impossible to give a simple, direct mapping of the poles and zeros to the transient performance, except for a first or second-order system. However the dynamic response of a higher order system can be approximately described by its dominant poles. Therefore a method used in linear control system design is to specify the desired dominant poles and then assign the other poles far away from them. This can be done by a user specification of the desired damping ratio ζ and the natural frequency ω_0 for the dominant poles, from which the parameters for the boundary of a D stable region can be determined. For example when the wedge region of Fig. 3 (e) is used, the relationship between (ζ, ω_0) and the boundary parameters (a,b) can be determined by: $a = \zeta \omega_0$, $b = \omega_0(1 - \zeta^2)^{1/2}$.

3. A general PID and PID-like controller design formulation with D stable regions

It is assumed that the plant has the following general transfer function with time delay

$$G_x(s) = \frac{N_x(s)}{D_x(s)} = \frac{a_0 + a_1s + \dots + a_{na-1}s^{na-1} + a_{na}s^{na}}{b_0 + b_1s + \dots + b_{nb-1}s^{nb-1} + s^{nb}} e^{-\tau s} \quad (5)$$

where $nb \geq na$, the coefficients a_i ($i=0,1,\dots,na$) and b_j ($j=0,1,\dots,nb-1$) are all real, and τ

is the pure time delay.

The delay may be approximated by a first order Pade approximation

$$e^{-\tau s} = \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s} = \frac{1 - Ts}{1 + Ts} \quad (6)$$

where $T=0.5\tau$. Substituting (6) into (5) yields

$$G_p(s) = \frac{c_0 + c_1s + \dots + c_ms^m + c_{na+1}s^{na+1}}{d_0 + d_1s + \dots + d_ns^n + d_{nb+1}s^{nb+1}} = \frac{N_p(s)}{D_p(s)} \quad (7)$$

where $c_0=a_0$, $c_i=a_i-Ta_{i-1}$, ($i=1,2,\dots,na$), $c_{na+1}=-Ta_{na}$

$$d_0=b_0, d_j=b_j+Tb_{j-1}, (j=1,2,\dots,nb-1), d_{nb}=1+Tb_{nb-1}, d_{nb+1}=T$$

Note that if there is a large time delay, a higher order Pade approximation may be used to increase the approximation accuracy. The resulting plant model is still a rational transfer function.

Let $G_c(s)$ denote the *PID* controller in Fig. 1 (a)

$$G_c(s) = \frac{N_c(s)}{D_c(s)} = \frac{K_d s^2 + K_p s + K_i}{s} \quad (8)$$

where K_p is the proportional gain, K_i the integral gain, and K_d the derivative gain.

Then the transfer function of the closed-loop system is given as

$$G(s) = \frac{N(s)}{D(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \quad (9)$$

and its characteristic polynomial is obtained as

$$D(s) = N_c(s)N_p(s) + D_c(s)D_p(s) = [c_0K_i + (c_0K_p + c_1K_i)s + (c_0K_d + c_1K_p + c_2K_i)s^2 + \dots + c_{na+1}K_d s^{na+3}] + [d_0s + d_1s^2 + \dots + d_{nb+1}s^{nb+2}] \quad (10)$$

The order of the above $D(s)$ depends on na and nb . When $nb-na \geq 1$ the order is $nb+2$, and when $nb=na$ the order is $nb+3$. When $nb-na \geq 1$, (10) can be generally represented

$$D(s) = c_0K_i + (c_0K_p + c_1K_i + d_0)s + \dots + (c_{na+1}K_d + d_{nb+1})s^{nb+2} \quad (11)$$

Note that the coefficients of polynomial (11) depend on the unknown design parameters $[K_p, K_i, K_d]$.

Similarly we can derive the characteristic polynomials for systems of Fig. 1 (b) to (d).

A common feature of these polynomials is that their coefficients are respectively functions of their unknown controller design parameters. Therefore the PID and PID-

like controller design shown in Fig. 1 (a) to (d) can be reduced to the following general problem.

General PID and PID-like design problem with D stable regions: Given a n -th order characteristic polynomial $D(s, \mathbf{u})$ for a closed-loop system

$$D(s, \mathbf{u}) = a_0(\mathbf{u}) + a_1(\mathbf{u})s + a_2(\mathbf{u})s^2 + \dots + a_n(\mathbf{u})s^n \quad (12)$$

With an unknown vector $\mathbf{u} = [u_1, u_2, \dots, u_m]^T$ representing the m design parameters in the controller, develop an approach to find \mathbf{u} such that the n characteristic roots of the characteristic polynomial (12) are located within a set of specified D stability regions or on their boundaries. For the control system shown in Fig. 1(a), $\mathbf{u} = [K_p, K_i, K_d]^T$, and $m=3$, $n=n_b+2$, $a_0(\mathbf{u})=c_0K_i$, $a_1(\mathbf{u})=c_0K_p+c_1K_i+d_0$, \dots , $a_n(\mathbf{u})=c_{n+1}K_d+d_{n+1}$. For Fig. 1(b), $\mathbf{u} = [T_1, T_2]^T$, for Fig. 1(c), $\mathbf{u} = [K_p, K_i, T_1, T_2]^T$, and for Fig. 1(d), $\mathbf{u} = [K_p, K_i, K_d, F_p, F_i, F_d]^T$.

4. Constraints satisfied by the roots and the design controller parameters \mathbf{u}

Equality constraints between roots and \mathbf{u} : An n -th order polynomial $D(s)$ with real coefficients has n complex characteristic roots (a real root is a special case). Let

$$s_i = \sigma_i + j\omega_i, \quad i=1, 2, \dots, n \quad (13)$$

denote its n roots. To simplify notation in the sequel, let $\Sigma_i = [\sigma_1, \sigma_2, \dots, \sigma_i]$, $\Omega_i = [\omega_1, \omega_2, \dots, \omega_i]$, $i=1, 2, \dots, n$. Therefore $D(s)$ can be written as

$$D(s) = \prod_{i=1}^n (s - s_i) = [\alpha_0(\Sigma_n, \Omega_n) + j\beta_0(\Sigma_n, \Omega_n)] + [\alpha_1(\Sigma_n, \Omega_n) + j\beta_1(\Sigma_n, \Omega_n)]s + \dots + [\alpha_{n-1}(\Sigma_n, \Omega_n) + j\beta_{n-1}(\Sigma_n, \Omega_n)]s^{n-1} + s^n \quad (14)$$

where $\alpha_i(\Sigma_n, \Omega_n)$ and $\beta_i(\Sigma_n, \Omega_n)$, $i=0, 1, 2, \dots, n-1$, which are all functions of $\Sigma_n = [\sigma_1, \sigma_2, \dots, \sigma_n]$, $\Omega_n = [\omega_1, \omega_2, \dots, \omega_n]$. To get expressions for α_i and β_i , one way is to

substitute (13) into (14), and then expand $\prod_{i=1}^n (s - s_i)$. However, as n increases, the

formulas for α_i and β_i become more complicated. So a simple recursive procedure for representing the coefficients of an n -th order polynomial based on those of an $(n-1)$ -th order polynomial is derived.

Note that polynomial (14) can be rewritten as

$$D(s) = \prod_{i=1}^n (s - s_i) = Re_n(s, \Sigma_n, \Omega_n) + jIm_n(s, \Sigma_n, \Omega_n) \quad (15)$$

Comparing (14) and (15), we obtain

$$\begin{aligned} Re_n(s, \Sigma_n, \Omega_n) &= \alpha_0(\Sigma_n, \Omega_n) + \alpha_1(\Sigma_n, \Omega_n)s + \dots + \alpha_{n-1}(\Sigma_n, \Omega_n)s^{n-1} + s^n \\ Im_n(s, \Sigma_n, \Omega_n) &= \beta_0(\Sigma_n, \Omega_n) + \beta_1(\Sigma_n, \Omega_n)s + \dots + \beta_{n-1}(\Sigma_n, \Omega_n)s^{n-1} \end{aligned} \quad (16)$$

Since

$$\prod_{i=1}^n (s - s_i) = \prod_{i=1}^{n-1} (s - s_i)(s - s_n) = (Re_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) + jIm_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}))(s - \sigma_n - j\omega_n) \quad (17)$$

where

$$\begin{aligned} Re_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) &= \alpha_0(\Sigma_{n-1}, \Omega_{n-1}) + \alpha_1(\Sigma_{n-1}, \Omega_{n-1})s + \dots + \alpha_{n-2}(\Sigma_{n-1}, \Omega_{n-1})s^{n-2} + s^{n-1} \\ Im_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) &= \beta_0(\Sigma_{n-1}, \Omega_{n-1}) + \beta_1(\Sigma_{n-1}, \Omega_{n-1})s + \dots + \beta_{n-2}(\Sigma_{n-1}, \Omega_{n-1})s^{n-2} \end{aligned} \quad (18)$$

Expanding (17) yields

$$\begin{aligned} Re_n(s, \Sigma_n, \Omega_n) &= sRe_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) - \sigma_n Re_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) + \omega_n Im_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) \\ Im_n(s, \Sigma_n, \Omega_n) &= sIm_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) - \sigma_n Im_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) - \omega_n Re_{n-1}(s, \Sigma_{n-1}, \Omega_{n-1}) \end{aligned} \quad (19)$$

Substituting (16) and (18) into (19), and then equating the coefficients of like powers of s in both sides lead to the following recursive formula

For $n=1$,

$$\alpha_0(\Sigma_1, \Omega_1) = -\sigma_1, \beta_0(\Sigma_1, \Omega_1) = -\omega_1, \quad (20)$$

for $n>1$,

$$\begin{aligned} \alpha_0(\Sigma_n, \Omega_n) &= -\alpha_0(\Sigma_{n-1}, \Omega_{n-1})\sigma_n + \beta_0(\Sigma_{n-1}, \Omega_{n-1})\omega_n \\ \beta_0(\Sigma_n, \Omega_n) &= -\beta_0(\Sigma_{n-1}, \Omega_{n-1})\sigma_n - \alpha_0(\Sigma_{n-1}, \Omega_{n-1})\omega_n \\ \alpha_j(\Sigma_n, \Omega_n) &= \alpha_{j-1}(\Sigma_{n-1}, \Omega_{n-1}) - \alpha_j(\Sigma_{n-1}, \Omega_{n-1})\sigma_n + \beta_j(\Sigma_{n-1}, \Omega_{n-1})\omega_n, j = 1, \dots, n-2 \\ \beta_j(\Sigma_n, \Omega_n) &= \beta_{j-1}(\Sigma_{n-1}, \Omega_{n-1}) - \beta_j(\Sigma_{n-1}, \Omega_{n-1})\sigma_n - \alpha_j(\Sigma_{n-1}, \Omega_{n-1})\omega_n, j = 1, \dots, n-2 \\ \alpha_{n-1}(\Sigma_n, \Omega_n) &= \alpha_{n-2}(\Sigma_{n-1}, \Omega_{n-1}) - \sigma_n \\ \beta_{n-1}(\Sigma_n, \Omega_n) &= \beta_{n-2}(\Sigma_{n-1}, \Omega_{n-1}) - \omega_n \end{aligned} \quad (21)$$

Note that what we need is to find the relationship between \mathbf{u} and Σ_n, Ω_n . Recall that the characteristic polynomial (12) has the same roots as (15), we thus have

$$a_n(\mathbf{u})\alpha_j(\Sigma_n, \Omega_n) = a_j(\mathbf{u}), \beta_j(\Sigma_n, \Omega_n) = 0, j=0, 1, \dots, n-1, \quad (22)$$

Inequality constraints satisfied by the roots: Suppose that a general desired D stability region is given in the form of (3). Then all the n roots of the characteristic polynomial (12) must satisfy (3), thus

$$g_l(\sigma_i, \omega_i) \leq 0, \quad l=1, 2, \dots, m, \quad i=1, 2, \dots, n \quad (23)$$

Note that inequalities (23) contain all the unknown roots (Σ_n, Ω_n) . Let us introduce a vector design variable $\mathbf{x} = [\mathbf{u}, \Sigma_n, \Omega_n]$. From (22) and (23) we can observe that if a design vector \mathbf{x} can be found such that the equalities (22) and the inequalities (23) are simultaneously satisfied, then a controller satisfying the D stability requirement is found. Finding \mathbf{x} is a standard problem that will be discussed later.

5. Feasibility Problem (FP) and its solution

A vector point $\mathbf{x} \in R^n$ satisfying a given set of non-linear equalities and inequalities

$$g_i(\mathbf{x})=0; \quad i=1, 2, \dots, m1, \quad g_i(\mathbf{x}) \leq 0; \quad i= m1+1, m1+2, \dots, m2 \quad (24)$$

is said to be a feasible point of the given set. Finding a feasible point is said to be a **feasibility problem (FP)**.

It is obvious that finding \mathbf{u} and the corresponding roots for the PID and PID-like controller design is equivalent to finding a feasible solution \mathbf{x} for the FP. The focus is how to find a solution. Fortunately the FP is a **standard problem** that has been widely studied in the field of numerical analysis and optimisation. Generally speaking there are two classes of techniques to solve the FP. The first contains the direct techniques. A number of algorithms in this class have been developed. For example, the extended Newton's method, the gradient method, and the trust-region approach [3]. The second transfers the FP problem into finding the roots for a set of **non-linear equations**, which is realised in the following way. The inequalities in (24) can be converted to a set of equalities by adding nonnegative slack variables, y_i^2 , $i= m1+1, m1+2, \dots, m2$, where the value of the slack variables are yet unknown. Thus (24) becomes

$$g_i(\mathbf{x})=0; \quad i=1, 2, \dots, m1, \quad g_i(\mathbf{x}) + y_i^2 = 0; \quad i= m1+1, m1+2, \dots, m2 \quad (25)$$

This reduces the FP into finding the solution of systems of non-linear equations (25) with a new design vector $\mathbf{x}_{new}=(\mathbf{x},\mathbf{y})$, where $\mathbf{y}=(y_{m1+1}, \dots, y_{m2})^T$. Newton's method is a well-known and very powerful technique for solving systems of non-linear equations [4]. Although Newton's method has the advantage of efficient convergence near the solution point, it may sometimes diverge when the initial point chosen is far away from the solution. A gradient-based approach, on the other hand, has the advantage of global convergence, but its local convergent rate is slow. This leads to the development of an efficient convergent numerical algorithm, i.e. the **Levenberg-Marquardt's non-linear least square method** which combines the advantages of the Newton and gradient-based approaches, and meanwhile overcomes their shortcomings [11,14]. Finding solution for (25) using the **LM** method is equivalent to finding an optimum point of the following non-linear least square problem

$$\text{Minimise } f(\mathbf{x}_{new}) = \sum_{i=1}^{m1} [g_i(x)]^2 + \sum_{i=m1+1}^{m2} [g_i(x) + y_i^2]^2 \quad (26)$$

For more details and the relevant references on FP, the readers are referred to the latest reference [3].

6. Simulation results with different D regions

In this section the approach developed above is illustrated by three examples with various sets of D regions. A comparative study will also be provided.

Example 1: Given an **unstable** plant with time delay

$$G_1(s) = \frac{1}{s-1} e^{-0.2s} \quad (27)$$

find a PI controller such that all the roots are required to locate within two different sets of *D* stability regions.

Using first order Pade approximation, the characteristic polynomial of the system can be derived as

$$D(s, \mathbf{u}) = 0.1s^3 + (0.9 - 0.1K_p)s^2 + (K_p - 0.1K_i - 1)s + K_i \quad (28)$$

Polynomial (28) has three roots: $s_i = \sigma_i + j\omega_i$, $i=1,2,3$. According to (21) and (22), σ_i, ω_i , and K_p, K_i must satisfy equality constraints

$$0.1(-\sigma_1\sigma_2\sigma_3 + \omega_1\omega_2\sigma_3 + \omega_1\sigma_2\omega_3 + \sigma_1\omega_2\omega_3) = K_i, \quad 0.1(\sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_2\sigma_3 - \omega_1\omega_2 - \omega_1\omega_3 - \omega_2\omega_3) = -1 - 0.1K_i + K_p, \quad 0.1(-\sigma_1 - \sigma_2 - \sigma_3) = 0.9 - 0.1K_p, \quad \omega_1\omega_2\omega_3 - \omega_1\sigma_2\sigma_3 - \sigma_1\omega_2\sigma_3 - \sigma_1\sigma_2\omega_3 = 0$$

$$\omega_1\sigma_2 + \sigma_1\omega_2 + \omega_1\sigma_3 + \omega_2\sigma_3 + \omega_3\sigma_1 + \omega_3\sigma_2 = 0, \quad -\omega_1 - \omega_2 - \omega_3 = 0 \quad (29)$$

Now we specify the two different D stable regions.

Case 1: The first set of the D stability regions is described by a parabola of the following form

$$D \cup \partial D: g(\sigma, \omega) = 4\sigma + \omega^2 + e \leq 0 \quad (30)$$

The physical meaning for the change of e is that when e is bigger, the boundary curve for (30) will move leftwards, and thus the allowed region for the roots is smaller (see Fig. 4).

All the roots within the $D \cup \partial D$ described by (30) must satisfy

$$4\sigma_i + \omega_i^2 + e \leq 0 \quad i=1,2,3 \quad (31)$$

By introducing three slack variable y_i , $i=1,2,3$, (31) becomes

$$4\sigma_i + \omega_i^2 + e + y_i^2 = 0 \quad i=1,2,3 \quad (32)$$

The nonlinear equations with the new design variables $\mathbf{x}_{\text{new}} = [K_p, K_i, \sigma_1, \sigma_2, \sigma_3, \omega_1, \omega_2, \omega_3, y_1, y_2, y_3]^T$ are then produced in (29) and (32). For different e , the simulation results for K_p , K_i are given in Table 1, and the D regions and the corresponding closed-loop characteristic roots are shown in Fig. 4. It can be observed that when e increases from 0 to 8.6 the real root of the system moves leftwards, while the other two complex roots move rightwards and become closer to the real axis. For $e=12$, no feasible roots have been found.

Case 2: In this case it is shown that the D stability regions may be a set of disjoint regions. For example, three roots may be required to respectively locate within three different circles described by the following inequalities:

$$\begin{aligned} D_1 \cup \partial D_1: & \quad (\sigma_1 - x_1)^2 + \omega_1^2 \leq R^2 & \quad \text{for root } s_1 \\ D_2 \cup \partial D_2: & \quad (\sigma_2 - x_2)^2 + (\omega_2 + y_2)^2 \leq R^2 & \quad \text{for root } s_2 \\ D_3 \cup \partial D_3: & \quad (\sigma_3 - x_2)^2 + (\omega_3 - y_2)^2 \leq R^2 & \quad \text{for root } s_3 \end{aligned} \quad (33)$$

Here the second and third circles are assumed to be symmetrical due to the property that the complex root pair s_2 and s_3 must be conjugate. The location and size of the D regions may be changed by specifying different values for x_1, x_2, y_2 , and R . For the same plant and PI controller as those used in case 1, the simulation results for K_p , K_i

are given in Table 2. The desired D regions and the corresponding characteristic roots are shown in Fig. 5. From Fig. 4 and 5 we can observe that **roots can be changed by different D specifications.**

Example 2: This example is to make a comparative study. Consider the plant

$$G_p(s) = Ke^{-\tau s} / (T_1 s + 1) \quad (34)$$

find the design parameters $\mathbf{u} = [K_p, K_i, T_d, \delta]$ for a modified PID controller

$$G_c(s) = \left(K_p + \frac{K_i}{s} \right) \frac{T_d s + 1}{\delta T_d s + 1} \quad (35)$$

where $0.05 \leq \delta \leq 0.2$.

A number of PID and PID-like tuning formulas for this example have been developed and seven of them are reproduced in Table 3. According to those given in Table 3 and the use of the second-order Pade approximation, the resulting characteristic polynomial is of the fifth order. The four dominant roots are plotted in Fig. 6 and the fifth one, a real root far from the imaginary axis is omitted. From the root distributions shown in Fig. 6 it can be seen that for the Internal model control (IMC), all the roots are within the critical damping line (damping ratio=0.707). For IAE-setpoint and ITAE-setpoint, each of them has a pair of complex roots just outside the critical damping line. For the rest of the four formulas, each of them has a pair of complex roots that are far outside the critical damping line (damping is too small), which may lead to a large overshoot and oscillatory response. The following simulations for step responses prove this point.

The step responses for the seven formulas are shown in Fig. 7. It is clear that the responses for both IAE-load and ITAE-load suffer from the too high overshoots (70%). The response for ISE-load has an overshoot of 24.5%, is oscillatory and has steady state error (not very good). The ISE-setpoint response has an overshoot of 16.5%, and is oscillatory. The IAE-setpoint and ITAE-setpoint responses are quite similar, with very fast rising time and small overshoot, but they all suffer from the high unexpected undershooting at the initial stage. For the IMC its step response is sluggish although it has no overshoot and no unexpected undershooting at the initial stage.

In order to overcome the high overshoots, the desired D stability region is chosen as a wedge with the critical damping line (damping ratio=0.707) (see Fig. 8). Table 4 gives the generated controller parameters $\mathbf{u}=[K_p, K_i, T_d, \delta]$ for different e values varying from 0 to 5, and the desired D stability regions and the characteristic roots are shown in Fig. 8. The step responses for different e values are shown in Fig. 9. In general they all give very smooth good step responses with small initial undershooting. When $e=2$ the step response has an overshoot of less than 7%. These results clearly demonstrate the flexibility of the design method by D stability regions. As expected the step responses given in Fig. 9 are a little bit slower than those of the IAE, ITAE and ISE, and this is a trade-off between the fast response and the high overshoot. We stress that if needed the damping line of the specified D region may be changed, and the approach presented in this paper can also be used to produce a fast response.

Example 3: Consider the high-order model with a large delay

$$G_p(s)=e^{-50s}/(10s+1)(s+1)(s+5) \quad (36)$$

find a PID controller with the design parameters $\mathbf{u}=[K_p, K_i, T_d]$

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (37)$$

Using the second-order Pade approximation which leads to a six-order closed-loop system, we can find that $\mathbf{u} = [K_p, K_i, T_d] = [1.5, 25.6, 4.75]$. The step responses and the dominant roots corresponding to the first and second order Pade approximation are shown in Fig. 10. They are both smooth and good, and the only difference is at the initial undershooting. For this example the computational time used for a Personal Computer is less than one second.

Now let us have a better understanding of the computing efficiency. Considering the fact that an optimisation problem with 1000 variables can be easily solved using a PC, there is no problem for the PID and PID-like controller design for a high-order system, for example tenth-order system. Actually we have used a similar approach for the Simultaneous Stabilisation Problem with D-stability. For more details see [19].

7. Conclusions

In this paper a new technique for designing a PID and PID-like controller such that the roots of the closed-loop system are controlled to be within a set of specified regions has been presented. The main contributions of the paper include:

- A general PID and PID-like controller design formulation with D stable regions is proposed
- The link between the PID controller design and the well-developed FP and optimisation is established
- The flexibility of the D stability regions enables us to deal with both the continuous and discrete system design in a unified way
- The classical root locus approach is extended to the multiple design variables.

Acknowledgement

This work was supported by the European Commission through the Long Term Research Project 28104, H²C (Heterogeneous Hybrid Control).

References

- [1] Astrom, K. J., T. Haggglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers- a survey," *Control Engineering Practice*, Vol. 1, pp. 699-714(1993).
- [2] Chang Y. H., and G. L. Wise, "Robust gamma stability of highly perturbed systems," *IEE Proc. Control theory Appl.*, Vol. 145, No. 2, pp. 165-175(1998).
- [3] Dennies JR, J. E., M. El-alem, and K. Williamson, "A trust-region approach to nonlinear systems of equalities and inequalities," *SIAM J. Optimisation*, Vol. 9, No. 2, pp. 291-315(1999).
- [4] Dennies JR, J. E., and R. B. Schanabel, *Numerical methods for unconstrained optimisation and nonlinear equations*, Prentice-Hall, Amsterdam, (1989).
- [5] De Paor A. M., and M. O'malley, "Controller of Zieger-Nichols type for unstable process with time delay," *Int. J. of Control*, Vol. 49, pp. 1273-1284(1989).
- [6] Eitelberg, E, "A regulating and tracking PI(D) controller," *Int. J. of Control*, Vol. 45, pp. 91-95(1987).
- [7] Hang, A. C., A. P. Loh, and V. U. Vasnani, "Relay feedback autotuning of cascade controller," *IEEE Trans. on Control System Technology*, Vol. 2, No. 1, pp.42-45(1994).
- [8] Ho W. K., O. P. Gan, E. B. tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas," *IEEE Trans. on Control Systems Technology*, Vol. 4, No. 4, pp.473-477(1996).
- [9] Juang, Y., Z. Hong, and Y. Wang, "Robustness of pole assignment in a specified region," *IEEE Trans. on Automatic Control*, Vol. 34, No. 7, pp. 758-760(1989).
- [10] Khan B. Z., and Brad Lehman, "Setpoint PI controllers for systems with large normalized dead time," *IEEE Trans. on Control System Technology*, Vol. 4, No. 4, pp.459-252(1996).

- [11] Levenberg, K., "A method for the solution of certain non-linear problem in least squares," *Quart. Appl. Math.*, Vol. 2, pp. 164-168(1944).
- [12] Leva L., and A. M. Colombo, "Method for optimising set-point weights in ISA-PID autotuners," *IEE Proc. – Control Theory Appl.* Vol. 146, No. 2, pp. 137-146(1999).
- [13] Loron, L., "Tuning PID controllers by the non-symmetrical optimum method," *Automatica*, Vol. 33, No. 1, pp. 103-107(1997).
- [14] Marquardt, W. "An algorithm for least-squares estimation of non-linear parameters," *SIAM J. Appl. Math.*, Vol. 11, pp. 431-441(1963).
- [15] Poulin A., and A. Pomerleau, "Unified PID design method based on a maximum peak resonance specification," *IEE Proc. Control theory Appl.*, Vol. 144, No. 6, pp. 566-574(1997).
- [16] Rivera, A. E., M. Morari, and S. Skogestad, "Internal model control. 4, PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, Vol. 25, pp. 252-265(1986).
- [17] Shafiei, Z., 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).
- [18] Tan, W., J. Liu and P. K. S. Tam, "PID tuning based on loop-shaping H_∞ control," *IEE Proc. Control theory Appl.*, Vol. 145, No. 6, pp. 485-490(1998).
- [19] Wang; Y., T. Schmitt-Hartmann; M. Schinkel; K. J. Hunt, "A new approach to simultaneous stabilisation with D stability and its application to control of antilock braking systems," In Proceedings of European Control Conference (ECC), Portugal, pp. 612-617(2001).
- [20] Ziegler, J. G. and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, Vol. 64, pp. 759-768(1942).

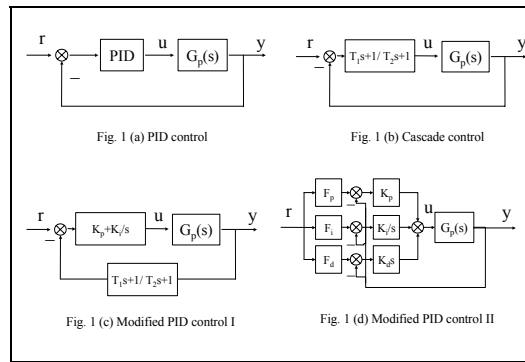


Fig. 1 (a)-(d) Diagram of PID and PID-like control systems

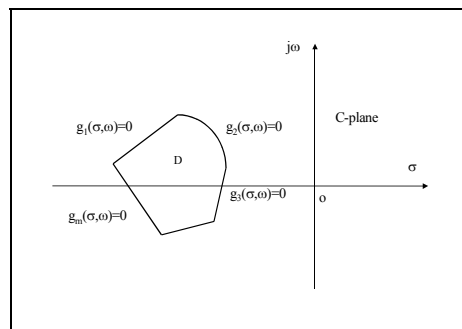


Fig. 2 A general D -stable region bounded by a set of curves

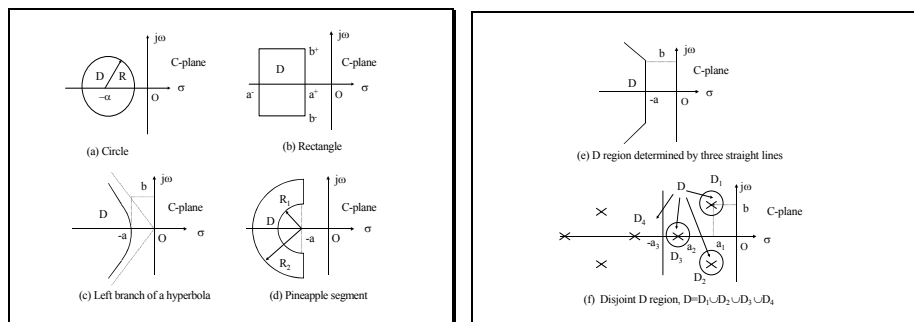


Fig. 3 (a)-(f) Examples of widely-used D -stable regions

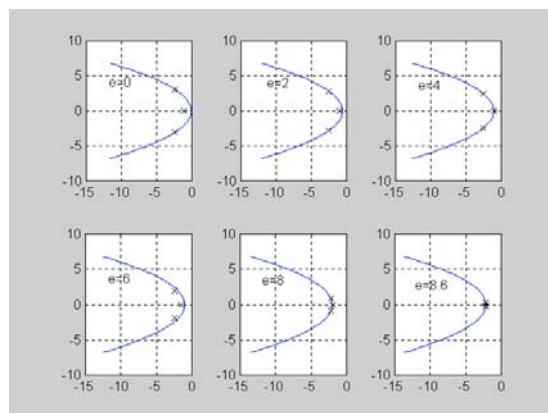


Fig. 4 Parabola with different e and the corresponding characteristic roots generated

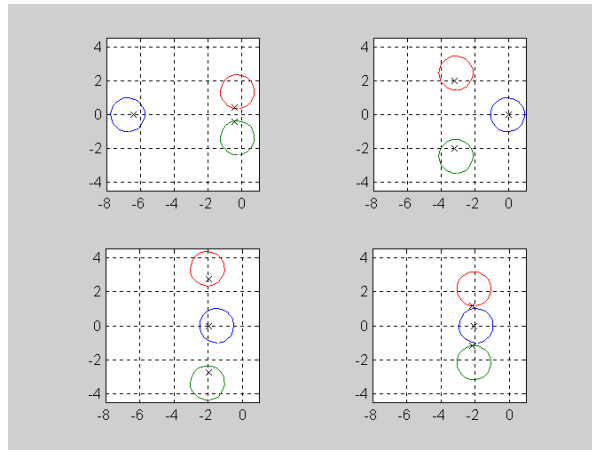


Fig. 5 Three disjoint circles and the corresponding characteristic roots generated

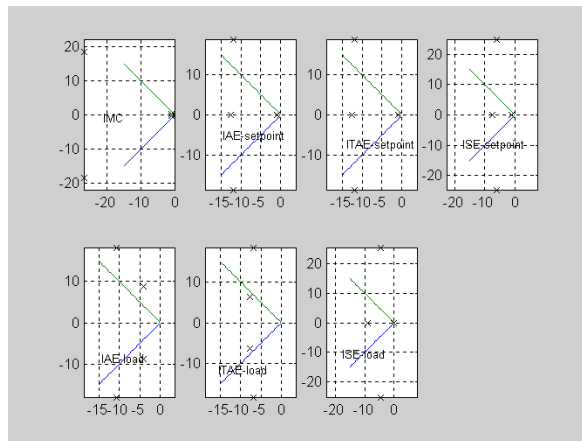


Fig. 6 Root distributions from the formulas in table 3

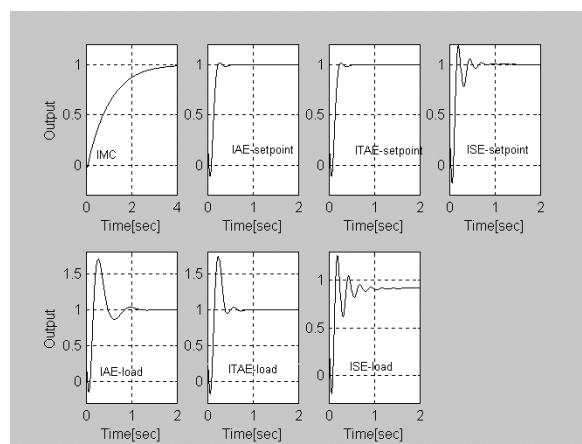


Fig. 7 Step responses for IMC, IAE-setpoint, ITAE-setpoint, ISE-load, IAE-setpoint, ITAE-load, and ISE-load

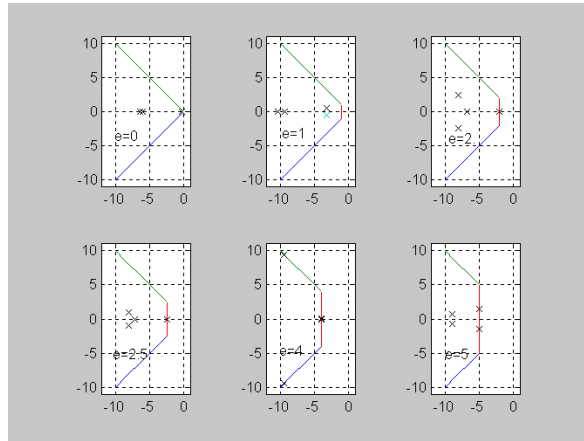


Fig. 8 Root distributions for different e values

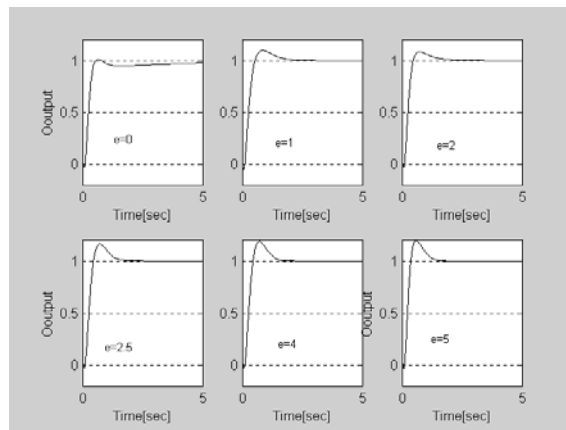


Fig. 9 Step responses for different e values

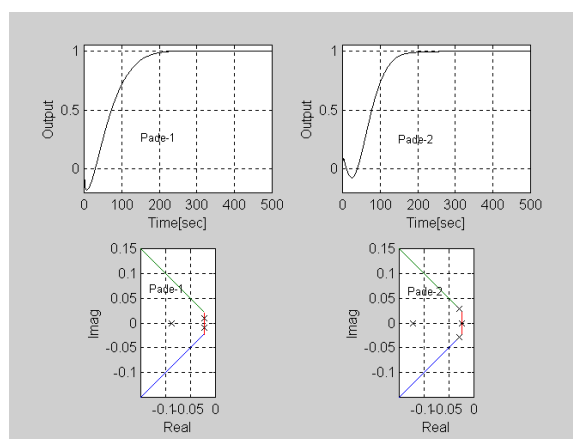


Fig. 10 Step responses and root distributions for example 3

Table 1: PI parameters and the corresponding roots for parabola with different e values.
 $G_p(s)=e^{-0.2s}/(s-1)$

e	0	2	4	6	8	8.6	12
K_p	3.184	3.042	2.905	2.758	2.575	2.513	no
K_i	1.660	1.384	1.268	1.366	1.149	1.027	no

Table 2: PI parameters and the corresponding roots for three disjoint circles. $G_p(s)=e^{-0.2s}/(s-1)$

x_1, R	-6.74, 1.0	-0.08, 1.0	-1.5, 1.0	-1.95, 1.0
x_2, y_2	-0.27, -1.36	-3.13, -2.45	-2.05, -3.34	-2.08, -2.16
K_p, K_i	1.66, 0.25	2.49, 0.07	3.13, 2.23	2.62, 1.24

Table 3: Existing PID tuning formulas for plant model. $G_p(s)=Ke^{-\tau s}/(T_1s+1)$, $\kappa=\tau/T_1$, $\delta=0.1, K=1, T_1=1, \tau=0.1$

Controller	$G_c(s) = (K_p + \frac{K_i}{s}) \frac{T_d s + 1}{\delta T_d s + 1}$		
Parameters	K_p	K_i	T_d
IMC	$\frac{1}{K(1+0.5\kappa)}$	$\frac{1}{T_1}$	0.5τ
IAE-setpoint	$\frac{0.65}{K} \kappa^{-1.04432}$	$\frac{K_p}{T_1} (0.9895 + 0.09539\kappa)$	$0.50814\kappa^{1.08433}$
ITAE-setpoint	$\frac{1.12762}{K} \kappa^{-0.80368}$	$\frac{K_p}{T_1} (0.9978 + 0.02860\kappa)$	$0.42844\kappa^{1.0081}$
ISE-setpoint	$\frac{0.71959}{K} \kappa^{-1.03092}$	$\frac{K_p}{T_1} (1.12666 - 0.18145\kappa)$	$0.54568\kappa^{0.86411}$
IAE-load	$\frac{0.98089}{K} \kappa^{-0.76167}$	$\frac{K_p}{T_1} 0.91032\kappa^{-1.05211}$	$0.59974\kappa^{0.89819}$
ITAE-load	$\frac{0.77902}{K} \kappa^{-1.06401}$	$\frac{K_p}{T_1} 1.1431\kappa^{-0.70949}$	$0.57137\kappa^{1.03826}$
ISE-load	$\frac{1.11907}{K} \kappa^{-0.89711}$	$\frac{K_p}{T_1} 0.7987\kappa^{-0.95480}$	$0.54766\kappa^{0.87798}$

Table 4: Controller parameters generated for wedges with different e values: $G_p(s)=e^{-0.1s}/(s+1)$

e	0	1	2	2.5	4	5
K_p	0.7707	0.6845	0.7814	0.7791	0.7671	0.8793
K_i	2.0975	5.8236	6.1209	6.8014	7.3407	9.6655
T_d	3.2664	0.5824	0.6340	0.5717	0.5224	0.4472
δ	0.1995	0.2000	0.2000	0.2500	0.2000	0.2000