Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Real and Complex Stabilization: Stability and Performance

Ming-Tzu Ho, Guillermo J. Silva, Aniruddha Datta and S. P. Bhattacharyya

***Abstract*— This paper presents recent results and algorithms that can be used to generate the entire set of stabilizing PID controllers for single-input single-output 1) continuous-time rational plants of arbitrary order, 2) discrete-time rational plants of arbitrary order, and 3) continuous-time first order plants with time delay. These algorithms follow from substantial theoretical advances on PID stabilization that have been reported by us in recent years. They display the rich mathematical structure underlying the topology of PID stabilizing sets. By presenting these algorithms without the highly technical details of the underlying theory, the paper seeks to make the results accessible to as wide an engineering audience as possible. Examples are presented throughout the paper to clarify the steps involved in implementing the different algorithms. We believe that these algorithms can significantly complement the current techniques for industrial PID design, many of which are adhoc in nature. In particular, the graphical displays of feasible design regions using 2-D and 3-D graphics should appeal to control designers and are very suitable for computer aided design where several performance objectives have to be overlaid and intersected. Specific design problems where these algorithms can be profitably used are discussed.**

I. Introduction

PID Controllers are widely used in applications [1]. However, in most cases, their designs are carried out using *adhoc* tuning rules. These rules have been developed over the years based primarily on empirical observations and industrial experience. In part this state of affairs is due to the fact that the state feedback observer based theory of modern and post-modern control theory including H_2, H_∞, μ and l_1 optimal control cannot be applied to PID control. Indeed, until the results to be described here appeared, it was not known how to even determine whether stabilization of a nominal system was possible using PID controllers.

In recent years, we have obtained a number of significant results on PID stabilization [2], [3], [4] which, we believe, could assist the industrial practitioner to carry out computer aided designs of PID controllers with guaranteed stability and performance. Given that PID controllers are used in such diverse applications as process control, rolling mills, aerospace, motion control, pneumatic, hydraulic, electrical and mechanical systems, disc drives and digital cameras, the impact of these results could be enormous.

The theoretical development of these results is quite involved and technical, and this could make them inaccessible to practicing engineers. However, the algorithms that result are straightforward, and can be easily programmed on a computer. The objective of this paper is to present these PID stabilization and design algorithms, devoid of detailed mathematical proofs, and show via examples how these algorithms can be used by the industrial practitioner to carry out computer-aided designs.

The paper is organized as follows. In Section II, we present an algorithm for determining the set of all stabilizing PID controllers for a continuous-time delay free plant of arbitrary order. An example is included to illustrate the detailed calculations involved. In Section III, we show how all PID stabilizers for a discrete-time plant of arbitrary order can be determined by suitably modifying the algorithm of Section II. Once again, an illustrative example is included. In Section IV, we present an algorithm to determine the set of all stabilizing PID controllers for a continuous-time first-order plant with time delay. Since plants of this type are widely encountered in process control, this algorithm should be of particular interest to industrial practitioners. An example is included to demonstrate the use of this algorithm. Section V discusses some PID controller *design* problems involving frequency domain performance specifications, which can be solved by using a complex version of the algorithm of Section II. The modifications required are indicated. Finally, Section VI contains some concluding remarks.

II. PID Controllers for Linear Time-invariant Continuous-time Systems

Consider the general feedback system shown in Fig. 1. Here r is the command signal, y is the output, $G(s)$ is the plant to be controlled, and $C(s)$ is the controller to be

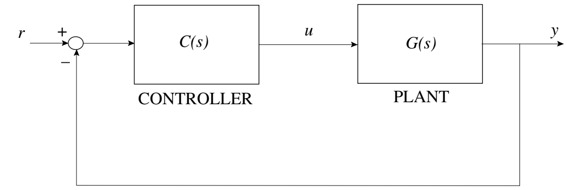

Fig. 1. Feedback control system.

The work of M. T. Ho was supported by the National Science Council of Taiwan under Grant NSC 92-2213-E-006-070. The work of A. Datta was supported by the U. S. National Science Foundation under Grant ECS-9903488, the Texas Advanced Technology Program under Grant No. 000512-0099-1999 and the U. S. National Cancer Institute under Grant CA90301. The work of S. P. Bhattacharyya was supported by a grant from National Instruments, the U. S. National Science Foundation under Grant ECS-9903488 and the Texas Advanced Technology Program under Grant No. 000512-0099-1999.

M. T. Ho is with the Engineering Science Department, National Cheng Kung University, 1 University Road, Tainan 701, Taiwan, e-mail: bruceho@mail.ncku.edu.tw

G. J. Silva is with the IBM Server Group, 11400 Burnet Road, Austin, TX 78758, U.S.A., e-mail: guilsilv@us.ibm.com

A. Datta and S. P. Bhattacharyya are with the Department of Electrical Engineering, Texas A & M University, College Station, TX 77843-3128, U.S.A., e-mail: datta@ee.tamu.edu; bhatt@ee.tamu.edu

designed. The controller $C(s)$ will be assumed to be of the PID type so that

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

where k_p, k_i and k_d are the proportional, integral and derivative gains respectively. For this section, the plant transfer function $G(s)$ will be assumed to be rational so that

$$G(s) = \frac{N(s)}{D(s)}$$

where $N(s)$, $D(s)$ are polynomials in the Laplace variable s. With this plant description, the closed-loop characteristic polynomial becomes

$$\delta(s, k_p, k_i, k_d) = sD(s) + (k_i + k_d s^2)N(s) + k_p sN(s). \quad (1)$$

The problem of stabilization using a PID controller is to determine the values of k_p, k_i and k_d for which the closed-loop characteristic polynomial $\delta(s, k_p, k_i, k_d)$ is Hurwitz, that is, has all its roots in the open left half plane. Since plants with a zero at the origin cannot be stabilized by PID controllers we exclude such plants at the outset. In this section, we simply present an algorithm for computationally characterizing all stabilizing PID controllers for a given plant with $N(0) \neq 0$. For a proof of the derivation, the reader is referred to [2].

We first introduce some definitions and notation.

Definition 2.1: The standard signum function $sgn : \mathcal{R} \rightarrow \{-1, 0, 1\}$ is defined by

$$sgn[x] = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases}$$

Definition 2.2: Let $a(s) = a_0 + a_1 s + \cdots + a_n s^n$ be a given real polynomial of degree n. Let $\mathcal{C}^-$ denote the open left-half plane (LHP) and $\mathcal{C}^+$ the open right-half plane (RHP). Then $l(a(s))$ and $r(a(s))$ denote the numbers of roots of $a(s)$ in $\mathcal{C}^-$ and $\mathcal{C}^+$ respectively.

Definition 2.3: Given a real polynomial $a(s)$ of degree n, the even-odd decomposition of $a(s)$ is defined as

$$a(s) = a_e(s^2) + s a_o(s^2)$$

where $a_e(s^2)$ and $s a_o(s^2)$ are the components of $a(s)$ made up of even and odd powers of s respectively.

To motivate the manipulations to follow we observe first that for a given real polynomial $a(s)$, the real and imaginary parts of $a(j\omega)$ are given by $a_e(-\omega^2)$ and $\omega a_o(-\omega^2)$ respectively. It will turn out that the root distribution (numbers of left and right half plane roots) of $a(s)$ can be determined from the zeros of its imaginary part and the signs of the real parts at these zeros. Finally, if $a(s)$ has unknown design parameters this approach to determining the root distribution is most conveniently applied when the unknown parameter sets appearing in the real and imaginary parts are separated, that is have no common elements. These ideas were used in [2] to develop an algorithm to determine the complete set of parameters k_p, k_i, k_d resulting in Hurwitz stability of (1). In the following we describe the essentials of this algorithm without mathematical proofs.

Using the even-odd decomposition of $N(s)$, define

$$N^*(s) = N(-s) = N_e(s^2) - sN_o(s^2).$$

Also let n, m be the degrees of $\delta(s, k_p, k_i, k_d)$ and $N(s)$ respectively. To achieve the parameter separation mentioned earlier, multiply $\delta(s, k_p, k_i, k_d)$ by $N^*(s)$ and rewrite $N(s)$, $D(s)$ in terms of their even-odd decompositions, to obtain

$$\begin{aligned} \nu(s) &:= \delta(s, k_p, k_i, k_d)N^*(s) \\ &= [s^2(N_e(s^2)D_o(s^2) - D_e(s^2)N_o(s^2)) \\ &\quad +(k_i + k_d s^2)(N_e(s^2)N_e(s^2) \\ &\quad -s^2 N_o(s^2)N_o(s^2))] + s[D_e(s^2)N_e(s^2) \\ &\quad -s^2 D_o(s^2)N_o(s^2) + k_p(N_e(s^2)N_e(s^2) \\ &\quad -s^2 N_o(s^2)N_o(s^2))]. \end{aligned} \quad (2)$$

Note that the polynomial $\nu(s)$ has degree $n + m$ and $\delta(s, k_p, k_i, k_d)$ is Hurwitz if and only if $\nu(s)$ has exactly the same number of closed RHP zeros as $N^*(s)$ and this is the condition we will use for stability. Note also that while the characteristic polynomial $\delta(s, k_p, k_i, k_d)$ has all three parameters appearing in both the even and odd parts, the test polynomial $\nu(s)$ exhibits parameter separation, that is, k_p appears in the odd part only while k_i and k_d appear in the even part only. This will facilitate the application of root counting formulas to $\nu(s)$.

To proceed, substitute $s = j\omega$ into (2) to obtain

$$\delta(j\omega, k_p, k_i, k_d)N^*(j\omega) = p(\omega,\ k_i,\ k_d) + jq(\omega,\ k_p) \quad (3)$$

where

$$\begin{aligned} p(\omega,\ k_i,\ k_d) &= p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega) && (4) \\ q(\omega,\ k_p) &= q_1(\omega) + k_p\ q_2(\omega) && (5) \\ p_1(\omega) &= -\omega^2(N_e(-\omega^2)D_o(-\omega^2) \\ &\quad -D_e(-\omega^2)N_o(-\omega^2)) && (6) \\ p_2(\omega) &= N_e(-\omega^2)N_e(-\omega^2) \\ &\quad +\omega^2 N_o(-\omega^2)N_o(-\omega^2)) && (7) \\ q_1(\omega) &= \omega(D_e(-\omega^2)N_e(-\omega^2) \\ &\quad +\omega^2 D_o(-\omega^2)N_o(-\omega^2)) && (8) \\ q_2(\omega) &= \omega(N_e(-\omega^2)N_e(-\omega^2) \\ &\quad +\omega^2 N_o(-\omega^2)N_o(-\omega^2)). && (9) \end{aligned}$$

The PID stabilization algorithm to be presented below is based on a fundamental and new result generalizing the classical Hermite-Biehler Theorem [5] to the case of root distribution determination of real polynomials which are not necessarily Hurwitz. This generalization reported in [7], [2], [6] provides an analytical expression for the difference between the numbers of roots of a real polynomial in the open left-half and open right-half planes. In our case we will exploit these results to impose the stability condition that $\nu(s)$ has exactly the same number of RHP roots as $N(-s)$. In order for this to happen a necessary condition is that $q(\omega, k_p)$ has at least

$$\begin{cases} \frac{|n-(l(N(s))-r(N(s)))|}{2} & \text{for } m+n \text{ even} \\ \\ \frac{|n-(l(N(s))-r(N(s)))|+1}{2} & \text{for } m+n \text{ odd} \end{cases} \quad (10)$$

real, nonnegative, distinct roots of odd multiplicity. The ranges of k_p satisfying (10) are called *allowable*. Let

$$0 = \omega_0 < \omega_1 < \ldots < \omega_{l-1} \quad (11)$$

denote the real, nonnegative distinct roots of $q(\omega, k_p)$ of odd multiplicity, and with $\omega_l := \infty$ write

$$sgn[p(\omega_j)] = i_j, j = 0, 1, \ldots .l. \quad (12)$$

It can be shown using the root counting results mentioned earlier, that the stability condition reduces to:

$$n - (l(N(s)) - r(N(s))) =$$

$$\begin{cases} \{i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2 i_{l-1} \\ +(-1)^l i_l\} \cdot (-1)^{l-1} \text{sgn}[q(\infty, k_p)] \\ \qquad \text{for } m+n \text{ even} \\ \\ \{i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2 i_{l-1}\} \\ \cdot (-1)^{l-1} \text{sgn}[q(\infty, k_p)] \\ \qquad \text{for } m+n \text{ odd} \end{cases} \quad (13)$$

and therefore the string of integers $\{i_0, i_1, \ldots, i_l\}$ will be called *admissible* if it satisfies (13).

Using the above we can present the following algorithm for determining all stabilizing $(k_p,\ k_i,\ k_d)$ values for the given plant. The reader is referred to [2] for a more complete development.

PID Stabilization Algorithm For LTI Plants:

Step 1: For the given $N(s)$ and $D(s)$, compute the corresponding $p_1(\omega)$, $p_2(\omega)$, $q_1(\omega)$, and $q_2(\omega)$ from (6)-(9);

Step 2: Determine the allowable ranges $P_i, i = 1, 2, \ldots, d$ of k_p from (10). The resulting ranges of k_p are the only ranges of k_p for which stabilizing $(k_i,\ k_d)$ values may exist;

Step 3: If there is no k_p satisfying **Step 2** then output **NO SOLUTION** and **EXIT**;

Step 4: Initialize $j = 1$ and $P = P_j$;

Step 5: Pick a range $[k_{low}, k_{upp}]$ in P and initialize $k_p = k_{low}$;

Step 6: Pick the number of grid points N and set $step = \frac{1}{N+1}[k_{upp} - k_{low}]$;

Step 7: Increase k_p as follows: $k_p = k_p + step$. If $k_p > k_{upp}$ then **GOTO Step 14**;

Step 8: For fixed k_p in **Step 7**, solve for the real, nonnegative, distinct finite zeros of $q(\omega, k_p)$ with odd multiplicities and denote them by $0 = \omega_0 < \omega_1 < \omega_2 < \cdots < \omega_{l-1}$. Also define $\omega_l = \infty$;

Step 9: Construct sequences of numbers $i_0, i_1, i_2, \cdots, i_l$ as follows:

(i) If $N^*(j\omega_t) = 0$ for some $t = 1, 2, \cdots, l-1$, then define

$$i_t = 0;$$

(ii) For all other $t = 0, 1, 2, \cdots, l$,

$$i_t \in \{-1, 1\}.$$

With $i_0, i_1, \cdots$ defined in this way, define the set $A_{(k_p)}$ as

$$A_{(k_p)} := \begin{cases} \{\{i_0, i_1, \cdots, i_l\}\} & \text{if } m+n \text{ is even} \\ \{\{i_0, i_1, \cdots, i_{l-1}\}\} & \text{if } m+n \text{ is odd;} \end{cases}$$

Step 10: Determine the admissible strings $\mathcal{I} = \{i_0, i_1, \cdots\}$ in $A_{(k_p)}$ from (13). If there is no admissible string then **GOTO Step 7**;

Step 11: For an admissible string $\mathcal{I} = \{i_0, i_1, \cdots\}$, determine the set of $(k_i,\ k_d)$ values that simultaneously satisfy the following string of linear inequalities:

$$[p_1(\omega_t) + (k_i - k_d\omega_t^2)p_2(\omega_t)]i_t > 0,$$
$$\forall\ t = 0, 1, 2, \cdots \text{ for which } N^*(j\omega_t) \neq 0;$$

Step 12: Repeat **Step 11** for all admissible strings $\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_v$ to obtain the corresponding admissible (k_i, k_d) sets $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_v$. The set of all stabilizing (k_i, k_d) values corresponding to the fixed k_p is then given by

$$\mathcal{S}_{(k_p)} = \cup_{x=1,\,2,\cdots,\,v} \mathcal{S}_x;$$

Step 13: **GOTO Step 7**

Step 14: Set $j = j + 1$ and $P = P_j$. If $j \leq d$ **GOTO STEP 5**; else, terminate the algorithm.

We now present an example to illustrate the detailed calculations involved in determining the stabilizing $(k_p,\ k_i,\ k_d)$ gain values.

Example 2.1: Consider the problem of determining stabilizing PID gains for the plant $G(s) = \frac{N(s)}{D(s)}$ where

$$\begin{aligned} N(s) &= s^3 - 2s^2 - s - 1 \\ D(s) &= s^6 + 2s^5 + 32s^4 + 26s^3 + 65s^2 - 8s + 1. \end{aligned}$$

The closed-loop characteristic polynomial is

$$\delta(s, k_p, k_i, k_d) = sD(s) + (k_i + k_d s^2)N(s) + k_p s N(s).$$

Thus $n = 7$ and $m = 3$. Also

$$\begin{aligned} N_e(s^2) &= -2s^2 - 1\ , \\ N_o(s^2) &= s^2 - 1\ , \\ D_e(s^2) &= s^6 + 32s^4 + 65s^2 + 1\ , \\ D_o(s^2) &= 2s^4 + 26s^2 - 8\ , \end{aligned}$$

and

$$N^*(s) = (-2s^2 - 1) - s(s^2 - 1).$$

Therefore, from (2) we obtain

$$\begin{aligned} \delta(s, k_p, k_i, k_d)N^*(s) = {} & [s^2(-s^8 - 35s^6 - 87s^4 + 54s^2 \\ & +9) + (k_i + k_d s^2)(-s^6 + 6s^4 \\ & +3s^2 + 1)] + s[(-4s^8 - 89s^6 \\ & -128s^4 - 75s^2 - 1) \\ & +k_p(-s^6 + 6s^4 + 3s^2 + 1)] \end{aligned}$$

so that

$$\begin{aligned}\delta(j\omega, k_p, k_i, k_d)\ N^*(j\omega) = &\\ &[p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega)]\\ &+j[q_1(\omega)L + k_p\ q_2(\omega)]\end{aligned}$$

where

$$\begin{aligned}p_1(\omega) &= \omega^{10} - 35\omega^8 + 87\omega^6 + 54\omega^4 - 9\omega^2\\ p_2(\omega) &= \omega^6 + 6\omega^4 - 3\omega^2 + 1\\ q_1(\omega) &= -4\omega^9 + 89\omega^7 - 128\omega^5 + 75\omega^3 - \omega\\ q_2(\omega) &= \omega^7 + 6\omega^5 - 3\omega^3 + \omega.\end{aligned}$$

In **Step 2**, the range of k_p such that $q_f(\omega,\ k_p)$ has at least 3 real, non-negative, distinct, finite zeros with odd multiplicities was determined to be $(-24.7513,\ 1)$ which is the *allowable* range. Now for a fixed $k_p \in (-24.7513,\ 1)$, for instance $k_p = -18$,we have

$$\begin{aligned}q(\omega, -18) &= q_1(\omega) - 18q_2(\omega)\\ &= -4\omega^9 + 71\omega^7 - 236\omega^5 + 129\omega^3 - 19\omega.\end{aligned}$$

Then the real, non-negative, distinct finite zeros of $q_f(\omega,\ -18)$ with odd multiplicities are

$$\omega_0 = 0,\ \omega_1 = 0.5195,\ \omega_2 = 0.6055,$$

$$\omega_3 = 1.8804,\ \omega_4 = 3.6848.$$

Also define $\omega_5 = \infty$. Since $m+n = 10$ which is even, and $N^*(s)$ has no $j\omega$-axis roots, from **Step 9**, the set $A_{(-18)}$ becomes

$$\left\{\begin{array}{cc}
\{-1,-1,-1,-1,-1,-1\} & \{1,-1,-1,-1,-1,-1\}\\
\{-1,1,-1,-1,-1,-1\} & \{1,1,-1,-1,-1,-1\}\\
\{-1,-1,1,-1,-1,-1\} & \{1,-1,1,-1,-1,-1\}\\
\{-1,1,1,-1,-1-1\} & \{1,1,1,-1,-1,-1\}\\
\{-1,-1,-1,1,-1,-1\} & \{1,-1,-1,1,-1,-1\}\\
\{-1,1,-1,1,-1,-1\} & \{1,1,-1,1,-1,-1\}\\
\{-1,-1,1,1,-1,-1\} & \{1,-1,1,1,-1,-1\}\\
\{-1,1,1,1,-1,-1\} & \{1,1,1,1,-1,-1\}\\
\{-1,-1,-1,-1,1,-1\} & \{1,-1,-1,-1,1,-1\}\\
\{-1,1,-1,-1,1,-1\} & \{1,1,-1,-1,1,-1\}\\
\{-1,-1,1,-1,1,-1\} & \{1,-1,1,-1,1,-1\}\\
\{-1,1,1,-1,1,-1\} & \{1,1,1,-1,1,-1\}\\
\{-1,-1,-1,1,1,-1\} & \{1,-1,-1,1,1,-1\}\\
\{-1,1,-1,1,1,-1\} & \{1,1,-1,1,1,-1\}\\
\{-1,-1,1,1,1,-1\} & \{1,-1,1,1,1,-1\}\\
\{-1,1,1,1,1,-1\} & \{1,1,1,1,1,-1\}\\
\{-1,-1,-1,-1,-1,1\} & \{1,-1,-1,-1,-1,1\}\\
\{-1,1,-1,-1,-1,1\} & \{1,1,-1,-1,-1,1\}\\
\{-1,-1,1,-1,-1,1\} & \{1,-1,1,-1,-1,1\}\\
\{-1,1,1,-1,-1,1\} & \{1,1,1,-1,-1,1\}\\
\{-1,-1,-1,1,-1,1\} & \{1,-1,-1,1,-1,1\}\\
\{-1,1,-1,1,-1,1\} & \{1,1,-1,1,-1,1\}\\
\{-1,-1,1,1,-1,1\} & \{1,-1,1,1,-1,1\}\\
\{-1,1,1,1,-1,1\} & \{1,1,1,1,-1,1\}\\
\{-1,-1,-1,-1,1,1\} & \{1,-1,-1,-1,1,1\}\\
\{-1,1,-1,-1,1,1\} & \{1,1,-1,-1,1,1\}\\
\{-1,-1,1,-1,1,1\} & \{1,-1,1,-1,1,1\}\\
\{-1,1,1,-1,1,1\} & \{1,1,1,-1,1,1\}\\
\{-1,-1,-1,1,1,1\} & \{1,-1,-1,1,1,1\}\\
\{-1,1,-1,1,1,1\} & \{1,1,-1,1,1,1\}\\
\{-1,-1,1,1,1,1\} & \{1,-1,1,1,1,1\}\\
\{-1,1,1,1,1,1\} & \{1,1,1,1,1,1\}
\end{array}\right\}$$

Since $l(N(s)) = 2$ and $r(N(s)) = 1$,

$$l(N(s)) - r(N(s)) = 1$$

and

$$(-1)^{l-1}sgn[q(\infty,\ -18)] = -1,$$

it follows from **Step 10** that every *admissible* string

$$\mathcal{I} = \{i_0,\ i_1,\ i_2,\ i_3,\ i_4,\ i_5\}$$

must satisfy

$$\{i_0 - 2i_1 + 2i_2 - 2i_3 + 2i_4 - i_5\} \cdot (-1) = 6.$$

Hence the admissible strings are

$$\begin{aligned}\mathcal{I}_1 &= \{-1,\ -1,\ -1,\ 1,\ -1,\ 1\}\\ \mathcal{I}_2 &= \{-1,\ 1,\ 1,\ 1,\ -1,\ 1\}\\ \mathcal{I}_3 &= \{-1,\ 1,\ -1,\ -1,\ -1,\ 1\}\\ \mathcal{I}_4 &= \{-1,\ 1,\ -1,\ 1,\ 1,\ 1\}\\ \mathcal{I}_5 &= \{1,\ 1,\ -1,\ 1,\ -1,\ -1\}.\end{aligned}$$

From **Step 11**, for $\mathcal{I}_1$ it follows that the stabilizing (k_i, k_d) values corresponding to $k_p = -18$ must satisfy the string of inequalities:

$$\left\{\begin{array}{l}p_1(\omega_0) + (k_i - k_d\omega_0^2)p_2(\omega_0) < 0\\ p_1(\omega_1) + (k_i - k_d\omega_1^2)p_2(\omega_1) < 0\\ p_1(\omega_2) + (k_i - k_d\omega_2^2)p_2(\omega_2) < 0\\ p_1(\omega_3) + (k_i - k_d\omega_3^2)p_2(\omega_3) > 0\\ p_1(\omega_4) + (k_i - k_d\omega_4^2)p_2(\omega_4) < 0\\ p_1(\omega_5) + (k_i - k_d\omega_5^2)p_2(\omega_5) > 0\end{array}\right.$$

Substituting for ω_0, ω_1, ω_2, ω_3, ω_4 and ω_5 in the above expressions, we obtain

$$\left\{\begin{array}{l}k_i < 0\\ k_i - 0.2699k_d < -4.6836\\ k_i - 0.3666k_d < -10.0797\\ k_i - 3.5358k_d > 3.912\\ k_i - 13.5777k_d < 140.2055\end{array}\right. \tag{14}$$

The set of values of $(k_i,\ k_d)$ for which (14) holds can be solved by linear programming and is denoted by $\mathcal{S}_1$. For $\mathcal{I}_2$, we have

$$\left\{\begin{array}{l}k_i < 0\\ k_i - 0.2699k_d > -4.6836\\ k_i - 0.3666k_d > -10.0797\\ k_i - 3.5358k_d > 3.912\\ k_i - 13.5777k_d < 140.2055\end{array}\right. \tag{15}$$

The set of values of $(k_i,\ k_d)$ for which (15) holds can also be solved by linear programming and is denoted by $\mathcal{S}_2$. Similarly, we obtain

$$\left\{\begin{array}{l}\mathcal{S}_3 = \emptyset \text{ for } \mathcal{I}_3\\ \mathcal{S}_4 = \emptyset \text{ for } \mathcal{I}_4\\ \mathcal{S}_5 = \emptyset \text{ for } \mathcal{I}_5\end{array}\right.$$

Then, the stabilizing set of $(k_i,\ k_d)$ values when $k_p = -18$ is given by

$$\begin{aligned}\mathcal{S}_{(-18)} &= \cup_{x=1,\,2,\cdots,\,5}\mathcal{S}_x\\ &= \mathcal{S}_1 \cup \mathcal{S}_2\end{aligned}$$

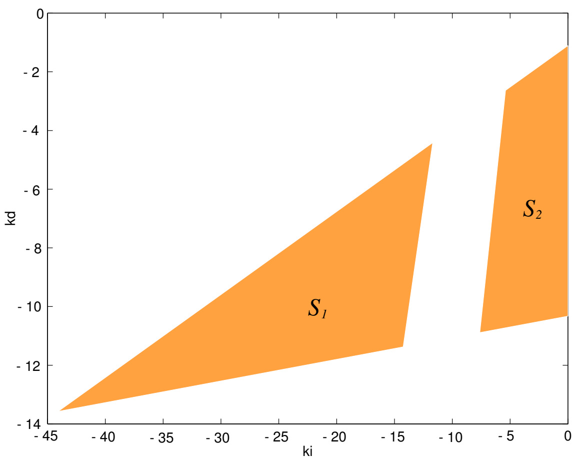

Fig. 2. The stabilizing set of $(k_i,\ k_d)$ values when $k_p = -18$.

The set $\mathcal{S}_{(-18)}$ and the corresponding $\mathcal{S}_1$ and $\mathcal{S}_2$ are shown in Fig. 2. By sweeping over different k_p values within the interval $(-24.7513,\ 1)$ and repeating the above procedure at each stage, we can generate the set of stabilizing $(k_p,\ k_i,\ k_d)$ values. This set is shown in Fig. 3.

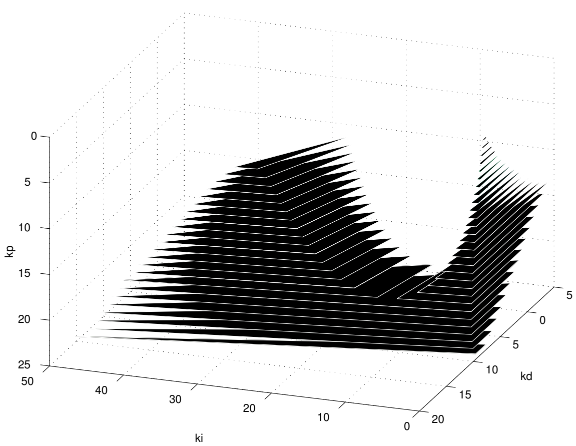

Fig. 3. The stabilizing set of $(k_p,\ k_i,\ k_d)$ values.

III. PID Controllers for Discrete-time Systems

In the case of a discrete-time system, the plant is given by

$$G_z(z) = \frac{N_z(z)}{D_z(z)}$$

where $N_z(z)$ and $D_z(z)$ are polynomials in the forward shift operator z. The discrete-time PID controller is given by:

$$\begin{aligned} C_z(z) &= k_p + k_i\frac{1}{1-z^{-1}} + k_d\frac{1-2z^{-1}+z^{-2}}{1-z^{-1}} \\ &= \frac{(k_p+k_i+k_d)z^2 - (k_p+2k_d)z + k_d}{z^2 - z} \end{aligned}$$

where k_p, k_i and k_d are the proportional, integral and derivative gains respectively. Plants with a zero at $z = 1$ cannot be stabilized by PID controllers because of the unstable pole-zero cancellation implied and are excluded at the outset. Using the bilinear transformation $z = \frac{w+1}{w-1}$, we obtain the w domain plant:

$$\frac{N(w)}{D(w)} = G_z(z)|_{z=\frac{w+1}{w-1}}$$

and the w-domain PID controller

$$\frac{B(w)}{A(w)} = \frac{k_i w^2 + 2(k_p+k_i)w + 2k_p + k_i + 4k_d}{2w+2}.$$

The corresponding w-domain closed loop characteristic polynomial becomes:

$$\begin{aligned} \delta(w,k_p,k_i,k_d) &= (2w+2)D(w) + (k_i w^2 \\ &\quad +2(k_p+k_i)w + 2k_p + k_i \\ &\quad +4k_d)N(w) \end{aligned} \tag{16}$$

and Hurwitz stability of this polynomial is equivalent to stability of the original discrete time system.

Following [3] we proceed as in the last section and multiply (16) by the factor $N(-w)$ to obtain

$$\delta^*(w,k_p,k_i,k_d) = N(-w)\delta(w,k_p,k_i,k_d).$$

By using the substitution

$$k_i = k_s - k_p \tag{17}$$

we can write

$$\begin{aligned} \delta^*(w,k_p,k_d,k_s) &= \delta'_e(w^2,k_p,k_d,k_s) + w\delta'_o(w^2,k_s) \\ &= [k_p\delta'_{ep}(w^2) + k_s\delta'_{es}(w^2) \\ &\quad +k_d\delta'_{ed}(w^2) + \delta'_{ec}(w^2)] \\ &\quad +w[k_s\delta'_{os}(w^2) + \delta'_{oc}(w^2)] \end{aligned} \tag{18}$$

where,

$$\begin{aligned} \delta'_{ep}(w^2) &= (1-w^2)(N_e^2 - w^2N_o^2) \\ \delta'_{es}(w^2) &= (1+w^2)(N_e^2 - w^2N_o^2) \\ \delta'_{ed}(w^2) &= 4(N_e^2 - w^2N_o^2) \\ \delta'_{ec}(w^2) &= 2(N_eD_e + w^2N_eD_o - w^2N_oD_e \\ &\quad -w^2N_oD_o) \\ \delta'_{os}(w^2) &= 2(N_e^2 - w^2N_o^2) \\ \delta'_{oc}(w^2) &= 2(N_eD_e + N_eD_o - N_oD_e \\ &\quad -w^2N_oD_o) \end{aligned} \tag{19}$$

From (18), it is clear that we can now proceed as in the previous section, i.e. fix k_s, then use linear programming to solve for the stabilizing values of k_p and k_d. In other words, the entire development in the last section can be repeated by replacing $\delta(s,k_p,k_i,k_d)N^*(s)$ in (2) by $\delta^*(w,k_p,k_d,k_s)$ and proceeding as before. However, this procedure will yield the stabilizing parameters only in the space of (k_p,k_d,k_s). In order to recover the stabilizing parameters in the original (k_p, k_i, k_d) space, we need to go through the inverse linear transformation.

Example 3.1: Consider a PID controller to stabilize the discrete-time system $\frac{N_z(z)}{D_z(z)}$ where

$$\begin{aligned} N_z(z) &= z + 1 \\ D_z(z) &= z^2 - 1.5z + 0.5. \end{aligned} \tag{20}$$

Using the bilinear transformation, we obtain the w-domain plant $\frac{N(w)}{D(w)}$ where

$$\begin{aligned} N(w) &= 2w^2 - 2w \\ D(w) &= w + 3. \end{aligned} \tag{21}$$

Fig. 4 shows the stabilizing regions in the space of (k_p,k_d,k_s) determined using the procedure outlined above. After going through the inverse linear transformation we

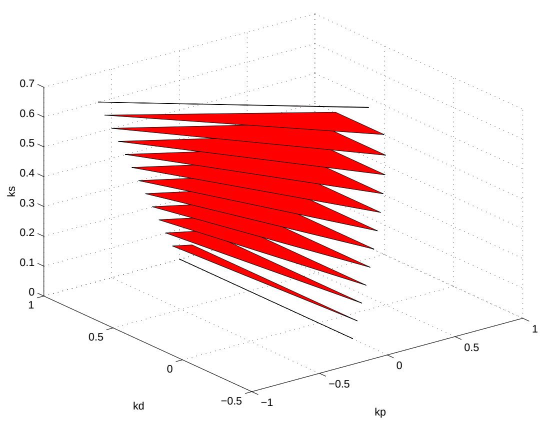

Fig. 4. The stabilizing region in the space of (k_p,k_d,k_s).

obtained the stabilizing regions in the space of (k_p,k_i,k_d). This region is shown in Fig. 5.

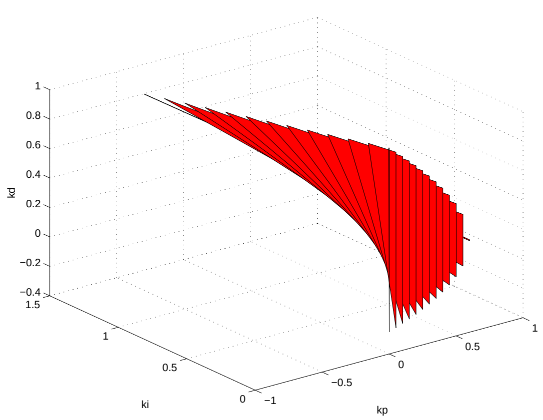

Fig. 5. The stabilizing region in the space of (k_p,k_i,k_d).

IV. PID Controllers for Continuous-time First Order Systems with Time Delay

In this section, we consider the feedback system of Fig. 1 where the plant $G(s)$ is described by

$$G(s) = \frac{k}{1 + Ts} e^{-Ls}. \tag{22}$$

Here k represents the steady-state gain of the plant, L the time delay, and T the time constant of the plant. As before, the controller is of the PID type, *i.e.*,

$$C(s) = k_p + \frac{k_i}{s} + k_d s \ .$$

The objective is to determine the set of controller parameters (k_p,k_i,k_d) for which the closed-loop system is stable. A complete solution to this problem has been presented in [4]. We provide a brief summary of these results.

[A] Open-loop Stable Plant

In this case $T > 0$. Furthermore, we make the standing assumption that $k > 0$ and $L > 0$. The next theorem presents the complete set of stabilizing PID controllers for an open-loop stable plant described by (22).

Theorem 4.1: The range of k_p values for which a given open-loop stable plant, with transfer function $G(s)$ as in (22), continues to have closed loop stability with a PID controller in the loop is given by

$$-\frac{1}{k} < k_p < \frac{1}{k}\left[\frac{T}{L}\alpha_1 sin(\alpha_1) - cos(\alpha_1)\right] \tag{23}$$

where α_1 is the solution of the equation

$$tan(\alpha) = -\frac{T}{T + L}\alpha \tag{24}$$

in the interval $(0, \pi)$. For k_p values outside this range, there are no stabilizing PID controllers. The complete stabilizing region is given by: (see Fig. 6)

1) For each $k_p \in (-\frac{1}{k}, \frac{1}{k})$, the cross-section of the stabilizing region in the (k_i, k_d) space is the trapezoid T.
2) For $k_p = \frac{1}{k}$, the cross-section of the stabilizing region in the (k_i, k_d) space is the triangle Δ.
3) For each $k_p \in \left(\frac{1}{k}, k_u := \frac{1}{k}\left[\frac{T}{L}\alpha_1 sin(\alpha_1) - cos(\alpha_1)\right]\right)$ the cross-section of the stabilizing region in the (k_i, k_d) space is the quadrilateral Q.

The parameters $m_j, b_j, w_j, j = 1, 2$ necessary for determining the boundaries of T, Δ and Q can be determined using the following equations:

$$m_j = \frac{L^2}{z_j^2} \tag{25}$$

$$b_j = -\frac{L}{kz_j}\left[\sin(z_j) + \frac{T}{L} z_j \cos(z_j)\right] \tag{26}$$

$$w_j = \frac{z_j}{kL}\left[\sin(z_j) + \frac{T}{L} z_j (\cos(z_j) + 1)\right] \tag{27}$$

where z_j, $j = 1, 2, \cdots$ are the real, positive solutions of

$$kk_p + cos(z) - \frac{T}{L}zsin(z) = 0 \quad (28)$$

arranged in ascending order of magnitude.

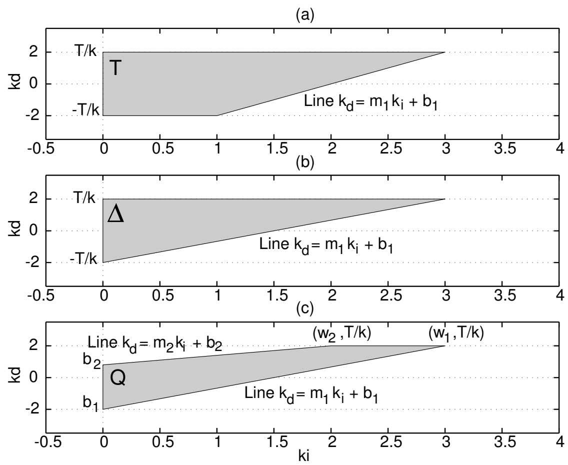

Fig. 6. The stabilizing region of (k_i,k_d) for: (a) $-\frac{1}{k} < k_p < \frac{1}{k}$; (b) $k_p = \frac{1}{k}$; (c) $\frac{1}{k} < k_p < k_u$.

[B] Open-Loop Unstable Plant

In this case $T < 0$ in (22). Furthermore, let us assume that $k > 0$ and $L > 0$.

Theorem 4.2: A necessary and sufficient condition for the existence of a stabilizing PID controller for the open-loop unstable plant (22) is $|\frac{T}{L}| > 0.5$. If this condition is satisfied, then the range of k_p values for which a given open-loop unstable plant, with transfer function $G(s)$ as in (22), can be stabilized using a PID controller is given by

$$\frac{1}{k}\left[\frac{T}{L}\alpha_1 sin(\alpha_1) - cos(\alpha_1)\right] < k_p < -\frac{1}{k} \quad (29)$$

where α_1 is the solution of the equation

$$tan(\alpha) = -\frac{T}{T+L}\alpha \quad (30)$$

in the interval $(0, \pi)$. In the special case of $|\frac{T}{L}| = 1$, we have $\alpha_1 = \frac{\pi}{2}$. For k_p values outside this range, there are no stabilizing PID controllers. Moreover, the complete stabilizing region is characterized by: (see Fig. 7)

For each $k_p \in \left(k_l := \frac{1}{k}\left[\frac{T}{L}\alpha_1 \sin(\alpha_1) - \cos(\alpha_1)\right], -\frac{1}{k}\right)$, the cross-section of the stabilizing region in the (k_i, k_d) space is the quadrilateral Q.

The parameters m_j, b_j and w_j, $j = 1, 2$ necessary for determining the boundary of Q are as defined in the statement of Theorem 4.1.

In view of Theorem 4.1, we now propose an algorithm to determine the set of stabilizing parameters for the plant (22) with $T > 0$.

PID Stabilization Algorithm for Time-Delay Plants:

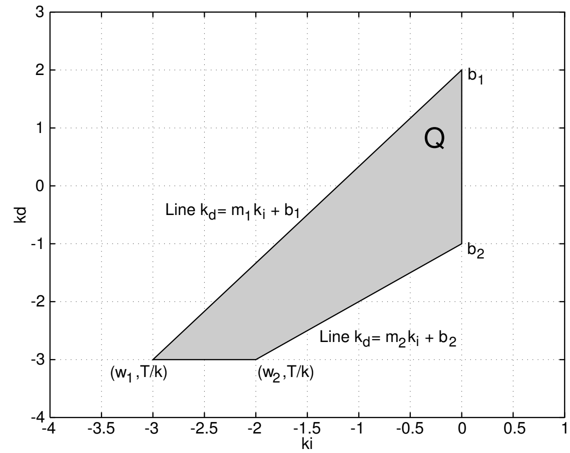

Fig. 7. The stabilizing region of (k_i,k_d) for $k_l < k_p < -\frac{1}{k}$.

(1) Initialize $k_p = -\frac{1}{k}$ and $step = \frac{1}{N+1}\left(k_u + \frac{1}{k}\right)$, where N is the desired number of grid points.
(2) Increase k_p as follows: $k_p = k_p + step$.
(3) If $k_p < k_u$ then go to Step 4. Else, terminate the algorithm.
(4) Find the roots z_1 and z_2 of (28).
(5) Compute the parameters m_j and b_j, $j = 1, 2$ associated with the previously found z_j by using (25) and (26).
(6) Determine the stabilizing region in the k_i-k_d space using Fig. 6.
(7) Go to Step 2.

A similar algorithm can be written down for the case of an open-loop unstable plant by using Theorem 4.2.

We next present an example that illustrates the use of the above algorithm to determine stabilizing PID parameters.

Example 4.1: Consider the PID stabilization problem for a plant described by the differential equation

$$\frac{dy(t)}{dt} = -0.5y(t) + 0.5u(t-4) \ .$$

This process can also be described by the transfer function $G(s)$ in (22) with the following parameters: $k = 1$, $T = 2$ sec, and $L = 4$ sec. Since the system is open-loop stable we use Theorem 4.1 to find the range of k_p values for which a solution to the PID stabilization problem exists. We first compute the parameter $\alpha_1 \in (0, \pi)$ satisfying the following equation

$$\tan(\alpha) = -0.3333\alpha \ .$$

Solving this equation we obtain $\alpha_1 = 2.4557$. Thus, from (23) the range of k_p values is given by

$$-1 < k_p < 1.5515 \ .$$

We now sweep over the above range of k_p values and determine the stabilizing set of (k_i, k_d) values at each stage using the previous algorithm. These regions are sketched in Fig. 8.

Any PID gains selected from these regions will result in closed-loop stability and any gains outside will result in instability. Now, consider the following performance specifications:

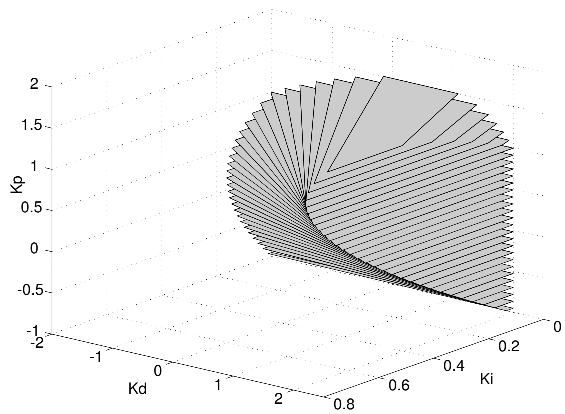

Fig. 8. The stabilizing region of (k_p,k_i,k_d) values for the PID controller in Example 4.1.

1) Settling time ≤ 60 secs;
2) Overshoot $\leq 20\%$.

We can obtain the transient responses of the closed-loop system for the (k_p,k_i,k_d) values inside the regions depicted in Fig.8. In general we also need some tolerance around the controller parameters, that is we want the controller to be controller-robust or non-fragile [8]. Thus we only consider PID gains lying inside the following box defined in the parameter space:

$$0.1 \leq k_p \leq 1 \ \ , \ \ 0.1 \leq k_i \leq 0.3 \ \ \text{and} \ \ 0.5 \leq k_d \leq 1.5 \ .$$

By searching over this box, several (k_p,k_i,k_d) values are found to meet the desired performance specifications. We arbitrarily set the controller parameters to: $k_p = 0.3444$, $k_i = 0.1667$, $k_d = 0.8333$. Fig. 9 shows the step response of the resulting closed-loop system. It is clear from the figure that the closed-loop system is stable, the output $y(t)$ tracks the step input signal and the performance specifications are met. The figure also shows the responses of the closed-loop

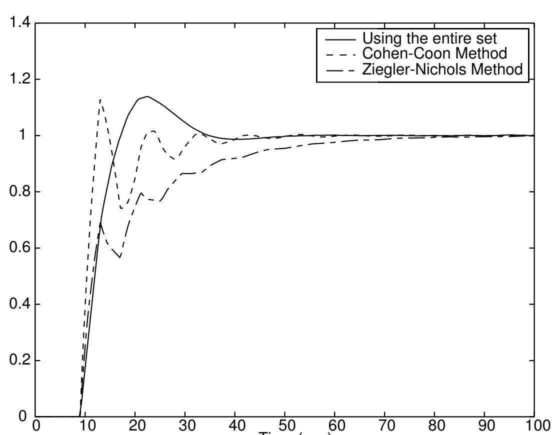

Fig. 9. Time response of the closed-loop system for Example 4.1.

systems for the case of a PID controller designed using the Cohen-Coon method ($k_p = 0.9180, k_i = 0.1456, k_d = 0.9845$) and the Ziegler-Nichols method ($k_p = 0.6, k_i = 0.075, k_d = 1.2$) Notice that in these cases also the system is stable and achieves setpoint following. However, the responses are much more oscillatory. $\triangle$

Although the design presented above is essentially an optimization search by gridding, nevertheless the fact that the algorithm of this section can be used to confine the search to the stabilizing set makes the design problem orders of magnitude easier.

V. PID Controller Design

The PID stabilization algorithms presented in the last three sections can be used to determine the entire set of stabilizing PID controllers. Hence, in principle, they can be used to facilitate PID design; indeed, by confining the search for the PID parameters to the stabilizing regions, it is possible to optimize different performance indices while ensuring that the stability constraint is always satisfied. This, however, constitutes a numerical design approach and Example 4.1 does illustrate this point. In certain situations, nevertheless, it is possible to do better than mere numerical optimization and this section is devoted to a discussion of such situations that have arisen so far in our research.

In many situations control system performance can be specified by a frequency domain inequality or equivalently an H_∞ norm constraint on a closed loop transfer function $G(s) = \frac{N(s)}{D(s)}$:

$$\|G(s)\|_\infty < \gamma.$$

It has been shown in [10] that the above condition is equivalent to Hurwitz stability of the complex polynomial family:

$$\gamma D(s) + e^{j\theta} N(s), \theta \in [0, 2\pi].$$

In our PID design problem the polynomials $D(s), N(s)$ will have the PID gains embedded in them and the set of parameters achieving specifications is given by those achieving simultaneously the stabilization of the complex polynomial family as well as the real closed loop characteristic polynomial. It turns out that the set of PID gains achieving stabilization of a complex polynomial family and therefore attaining the specifications can be found by an extension of the algorithm given for the real case. Towards this end, consider a complex polynomial of the form:

$$c(s,\ k_p,\ k_i,\ k_d) = L(s) + (k_d s^2 + k_p s + k_i) M(s) \tag{31}$$

where $L(s)$ and $M(s)$ are given complex polynomials. In [11], the results on PID stabilization presented in Section II were extended to the stabilization of (31). The algorithm, described below, is similar to the stabilization algorithm given for the real case and we will therefore not write the algorithm in detail but only point out the differences in the formulas and steps from that of the real case. We then show through examples how, many PID *performance or design* problems can be converted into stabilization problems of complex polynomial families of the form of (31) and solved using this algorithm.

A. Complex PID Stabilization Algorithm

The complex PID Stabilization Algorithm is similar to the algorithm given for the real case in Section 2 and we need only to point out the differences in some formulas and steps. To do that we first introduce some definitions and notation.

Definition 5.1: Let $a(s)$ be a given complex polynomial of degree n:

$$\begin{aligned} a(s) &= (a_0 + jb_0) + (a_1 + jb_1)s + \cdots \\ &\quad + (a_{n-1} + jb_{n-1})s^{n-1} + (a_n + jb_n)s^n, \end{aligned}$$

$a_n + jb_n \neq 0$. The real-imaginary decomposition of $a(s)$ is defined as

$$a(s) = a_R(s) + a_I(s)$$

where

$$\begin{aligned} a_R(s) &= a_0 + jb_1 s + a_2 s^2 + jb_3 s^3 + \cdots \\ a_I(s) &= jb_0 + a_1 s + jb_2 s^2 + a_3 s^3 + \cdots . \end{aligned}$$

Now we consider a complex polynomial of the form:

$$\begin{aligned} c(s,\ k_p,\ k_i,\ k_d) &= L(s) \\ &\quad + (k_d s^2 + k_p s + k_i)M(s) \end{aligned} \tag{32}$$

where $L(s)$ and $M(s)$ are two given complex polynomials. Write $L(s)$ and $M(s)$ in terms of their real-imaginary decompositions:

$$\begin{aligned} L(s) &= L_R(s) + L_I(s) \\ M(s) &= M_R(s) + M_I(s) \end{aligned}$$

and define

$$M^*(s) = M_R(s) - M_I(s)$$

and

$$\nu(s) = c(s, k_p, k_i, k_d)M^*(s)$$

Also let n, m be the degrees of $c(s,\ k_p,\ k_i,\ k_d)$ and $M(s)$ respectively. Evaluating the polynomial $\nu(s)$ at $s = j\omega$, we obtain

$$\begin{aligned} \nu(j\omega) &= c(j\omega,\ k_p,\ k_i,\ k_d)M^*(j\omega) \\ &= p(\omega,\ k_i,\ k_d) + jq(\omega,\ k_p) \end{aligned}$$

where

$$p(\omega,\ k_i,\ k_d) = p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega) \tag{33}$$

$$q(\omega,\ k_p) = q_1(\omega) + k_p q_2(\omega) \tag{34}$$

$$p_1(\omega) = L_R(j\omega)M_R(j\omega) - L_I(j\omega)M_I(j\omega) \tag{35}$$

$$p_2(\omega) = M_R^2(j\omega) - M_I^2(j\omega) \tag{36}$$

$$q_1(\omega) = \frac{1}{j}[L_I(j\omega)M_R(j\omega) - L_R(j\omega)M_I(j\omega)] \tag{37}$$

$$q_2(\omega) = \omega[M_R^2(j\omega) - M_I^2(j\omega)]. \tag{38}$$

Let ξ denote the leading coefficient of $c(s,\ k_p,\ k_i,\ k_d)M^*(s)$. The procedure for determining all stabilizing $(k_p,\ k_i,\ k_d)$ for which $c(s,\ k_p,\ k_i,\ k_d)$ is Hurwitz for the given $L(s)$ and $M(s)$ is identical to the stabilization algorithm of Section 2 except for the following steps below, labelled Step ic, in the computation of the *allowable* range and *admissible* strings.

Differences between real and complex PID Stabilization Algorithms:

- **Step 1c** Compute $p_1(\omega), p_2(\omega), q_1(\omega), q_2(\omega)$ from (35)-(38).
- **Step 2c** The *allowable* ranges of k_p are such that $q(\omega,\ k_p)$ has at least

$$\begin{cases} |n - (l(M(s)) - r(M(s)))| - 1\ , \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is purely real,} \\ \quad \text{or } m+n \text{ is odd and } \xi \text{ is purely imaginary} \\ |n - (l(M(s)) - r(M(s)))|\ , \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is not purely real,} \\ \quad \text{or } m+n \text{ is odd and } \xi \text{ is not purely imaginary} \end{cases}$$

  real, distinct finite zeros with odd multiplicities. The resulting ranges of k_p are the only ranges of k_p for which stabilizing $(k_i,\ k_d)$ values may exist;
- **Step 8c** For fixed k_p solve for the real, distinct finite zeros of $q(\omega, k_p)$ with odd multiplicities and denote them by $\omega_1 < \omega_2 < \cdots < \omega_{l-1}$ and let $\omega_0 = -\infty$ and $\omega_l = \infty$;
- **Step 9c** The construction of the sequences of numbers $i_0, i_i, i_2, \cdots, i_l$ is as follows:
  If $M^*(j\omega_t) = 0$ for some $t = 0, 1, \cdots, l$, then define

$$i_t = 0;$$

  else

$$i_t \in \{-1, 1\}, \text{ for all other } t = 0, 1, \cdots, l.$$

  With $i_0, i_1, \cdots$ defined in this way, define the set $A_{(k_p)}$ as

$$A_{(k_p)} = \begin{cases} \{\{i_0, i_1, \cdots, i_l\}\}\ , \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is purely real,} \\ \quad \text{or } m+n \text{ is odd and } \xi \text{ is purely imaginary} \\ \{\{i_1, i_2, \cdots, i_{l-1}\}\}\ , \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is not purely real, or } m+n \text{ is odd and } \xi \text{ is not purely imaginary} \end{cases}$$

- **Step 10c** Determine the *admissible* strings $\mathcal{I} \in A_{(k_p)}$ such that the following equality holds:

$$n - (l(M(s)) - r(M(s))) =$$

$$\begin{cases} \frac{1}{2}\{i_0 \cdot (-1)^{l-1} + 2\sum_{r=1}^{l-1} i_r \cdot (-1)^{l-1-r} \\ -i_l\} \cdot \text{sgn}[q(\infty,\ k_p)] \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is purely real,} \\ \quad \text{or } m+n \text{ is odd and } \xi \text{ is purely imaginary} \\ \frac{1}{2}\{2\sum_{r=1}^{l-1} i_r \cdot (-1)^{l-1-r}\} \cdot \text{sgn}[q(\infty,\ k_p)] \\ \quad \text{if } m+n \text{ is even and } \xi \text{ is not purely real,} \\ \quad \text{or } m+n \text{ is odd and } \xi \text{ is not purely imaginary} \end{cases} \tag{39}$$

We now give some application examples of PID performance using the complex stabilization algorithm.

B. Synthesis of H_∞ PID controllers

First, let us consider the problem of synthesizing PID controllers for which the closed-loop system is internally stable and the H_∞-norm of a certain closed loop transfer function is less than a prescribed level. In particular, the following closed-loop transfer functions are considered:

- The sensitivity function:

$$S(s) = \frac{1}{1+C(s)G(s)}. \quad (40)$$

- The complementary sensitivity function:

$$T(s) = \frac{C(s)G(s)}{1+C(s)G(s)}. \quad (41)$$

- The input sensitivity function:

$$U(s) = \frac{C(s)}{1+C(s)G(s)}. \quad (42)$$

As shown in [9], various performance and robustness specifications can be captured by using the H_∞-norm of weighted versions of the transfer functions (40)-(42). It can be verified that when $C(s)$ is a PID controller, the transfer functions (40)-(42) can all be represented in the following general form:

$$T_{cl}(s,\ k_p,\ k_i,\ k_d) \ = \ \frac{A(s)+(k_d s^2+k_p s+k_i)B(s)}{sD(s)+(k_d s^2+k_p s+k_i)N(s)} \quad (43)$$

where $A(s)$ and $B(s)$ are some real polynomials. For the transfer function $T_{cl}(s,\ k_p,\ k_i,\ k_d)$ and a given number $\gamma > 0$, the standard H_∞ performance specification usually takes the form:

$$\|W(s)T_{cl}(s,\ k_p,\ k_i,\ k_d)\|_\infty < \gamma \quad (44)$$

where $W(s)$ is a stable frequency-dependent weighting function that is selected to capture the desired design objectives at hand. Suppose the weighting function $W(s) = \frac{W_n(s)}{W_d(s)}$, where $W_n(s)$ and $W_d(s)$ are coprime polynomials and $W_d(s)$ is Hurwitz. Define the polynomials $\delta(s,k_p,k_i,k_d)$ and $\phi(s,k_p,k_i,k_d,\gamma,\theta)$ as follows:

$$\delta(s,\ k_p,\ k_i,\ k_d) \ \triangleq \ sD(s)+(k_i+k_p s+k_d s^2)N(s)$$

and

$$\begin{aligned}\phi(s,\ k_p,\ k_i,\ k_d,\ \gamma,\ \theta) \triangleq \\ & [sW_d(s)D(s)+\tfrac{1}{\gamma}e^{j\theta}W_n(s)A(s)] \\ & +(k_d s^2+k_p s+k_i)[W_d(s)N(s) \\ & +\tfrac{1}{\gamma}e^{j\theta}W_n(s)B(s)].\end{aligned}$$

Then as shown in [11], we can establish the following relationship between H_∞ synthesis using PID controllers and simultaneous stabilization of a complex polynomial family:

For a given $\gamma > 0$, there exist PID gain values $(k_d,\ k_p,\ k_i)$ such that $\|W(s)T_{cl}(s,\ k_p,\ k_i,\ k_d)\|_\infty < \gamma$ if and only if the following conditions hold:

(1) $\delta(s,\ k_p,\ k_i,\ k_d)$ is Hurwitz;
(2) $\phi(s,\ k_p,\ k_i,\ k_d,\ \gamma,\ \theta)$ is Hurwitz for all θ in $[0,\ 2\pi)$;
(3) $|W(\infty)T_{cl}(\infty,\ k_p,\ k_i,\ k_d)| < \gamma$.

The above equivalence can be used to determine stabilizing $(k_p,\ k_i,\ k_d)$ values such that the H_∞-norm of a certain closed-loop transfer function is less than a prescribed level. This is illustrated using the following example.

Example 5.1: Consider the plant $G(s) = \frac{N(s)}{D(s)}$ where

$$\begin{aligned} N(s) &= s-1 \\ D(s) &= s^2+0.8s-0.2 \end{aligned}$$

and the PID controller

$$C(s) = \frac{k_d s^2+k_p s+k_i}{s}.$$

In this example, we consider the problem of determining all stabilizing PID gain values for which $\|W(s)T(s,\ k_p,\ k_i,\ k_d)\|_\infty < 1$, where $T(s,\ k_p,\ k_i,\ k_d)$ is the complementary sensitivity function:

$$T(s,\ k_p,\ k_i,\ k_d) = \frac{(k_d s^2+k_p s+k_i)(s-1)}{s(s^2+0.8s-0.2)+(k_d s^2+k_p s+k_i)(s-1)}$$

and the weight $W(s)$ is chosen as a high pass transfer function:

$$W(s) \ = \ \frac{s+0.1}{s+1}.$$

We know that $(k_p,\ k_i,\ k_d)$ values meeting the H_∞ performance constraint exist if and only if the following conditions hold:

(1) $\delta(s,\ k_p,\ k_i,\ k_d) = s(s^2+0.8s-0.2)+(k_d s^2+k_p s+k_i)(s-1)$ is Hurwitz;
(2) $\phi(s,\ k_p,\ k_i,\ k_d,\ 1,\ \theta) = s(s+1)(s^2+0.8s-0.2)+(k_d s^2+k_p s+k_i)[(s+1)(s-1)+e^{j\theta}(s+0.1)(s-1)]$ is Hurwitz for all θ in $[0,\ 2\pi)$;
(3) $|W(\infty)T(\infty,\ k_p,\ k_i,\ k_d)| = |\frac{k_d}{k_d+1}| < 1$.

The set of all (k_p,k_i,k_d) values for which the H_∞ performance specifications are met are precisely the values of k_p,k_i,k_d for which conditions (1),(2) and (3) are satisfied. To search for such values of (k_p,k_i,k_d), we fix k_p and determine all the values of (k_i,k_d) for which conditions (1),(2) and (3) hold.

For the condition (1), with a fixed k_p, for instance $k_p = -0.35$, by setting $L(s) = s(s^2+0.8s-0.2)$ and $M(s) = s-1$, and using the algorithm of Section 2, we obtain the set of (k_i,k_d) values for which the closed-loop system is stable. This set is denoted by $\mathcal{S}_{(1,\ -0.35)}$ and is sketched in Fig. 10. Now fixing $k_p = -0.35$ and any fixed $\theta \in [0,2\pi)$, by setting $L(s) = s(s+1)(s^2+0.8s-0.2)$ and $M(s,\ \theta) = (s+1)(s-1)+e^{j\theta}(s+0.1)(s-1)$ and using the complex stabilization algorithm of Section V-A again we can solve a linear programming problem to determine the set of (k_i,k_d) values. Let this set be denoted by $\mathcal{S}_{(2,\ -0.35,\ \theta)}$. By keeping k_p fixed, sweeping over $\theta \in [0,2\pi)$, and using the complex

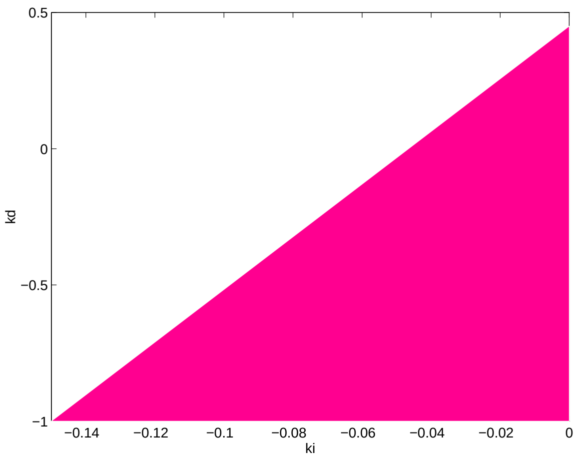

Fig. 10. The set $\mathcal{S}_{(1,\ -0.35)}$.

stabilization algorithm of Section V-A at each stage, we can determine the set of (k_i, k_d) values for which condition (2) is satisfied. This set is denoted by $\mathcal{S}_{(2,\ -0.35)}$ and is given by

$$\mathcal{S}_{(2,\ -0.35)} = \cap_{\theta\in[0,\ 2\pi)}\mathcal{S}_{(2,\ -0.35,\ \theta)}.$$

The set $\mathcal{S}_{(2,\ -0.35)}$ is sketched in Fig. 11. Let $\mathcal{S}_{(3,\ -0.35)}$ be the set of $(k_i,\ k_d)$ values satisfying condition (3) and this set is given by

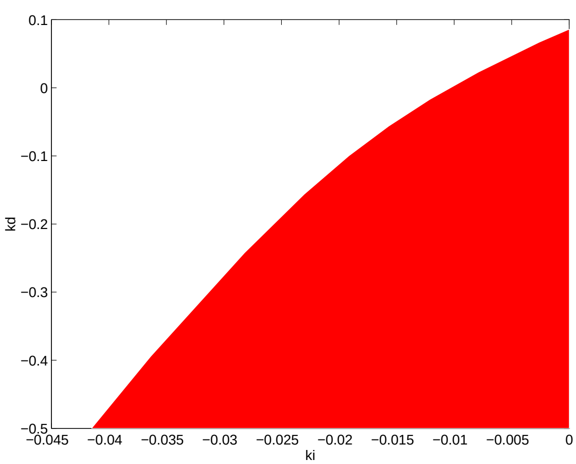

Fig. 11. The set $\mathcal{S}_{(2,\ -0.35)} = \cap_{\theta\in[0,\ 2\pi)}\mathcal{S}_{(2,\ -0.35,\ \theta)}$.

$$\mathcal{S}_{(3,\ -0.35)} = \{(k_i,\ k_d)|\ k_i \in \mathcal{R},\ k_d > -0.5\}.$$

Then for $k_p = -0.35$, the set of $(k_i,\ k_d)$ values for which $\|W(s)T(s,\ k_p,\ k_i,\ k_d)\|_\infty < 1$ is denoted by $\mathcal{S}_{(-0.35)}$ and is given by

$$\mathcal{S}_{(-0.35)} = \cap_{i=1,2,3}\mathcal{S}_{(i,\ -0.35)}.$$

In this case, we have $\mathcal{S}_{(-0.35)} = \mathcal{S}_{(2,\ -0.35)}$. Now, using root loci [2], it was determined that a necessary condition for the existence of stabilizing (k_i, k_d) values is that $k_p \in (-0.5566,\ -0.2197)$. Then, by sweeping over $k_p \in (-0.5566,\ -0.2197)$, and repeating the above procedure, we obtained the stabilizing set of $(k_p,\ k_i,\ k_d)$ values for which $\|W(s)T(s,\ k_p,\ k_i,\ k_d)\|_\infty < 1$. This set is sketched in Fig. 12.

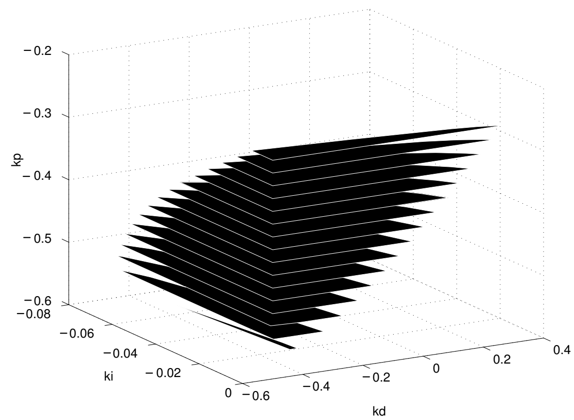

Fig. 12. The set of stabilizing $(k_p,\ k_i,\ k_d)$ values for which $\|W(s)T(s,\ k_p,\ k_i,\ k_d)\|_\infty < 1$.

C. PID Controller Design for Robust Performance

This subsection is devoted to the problem of synthesizing PID controllers for robust performance. In particular, we focus on the following robust performance specification [9]:

$$\| |W_1(s)S(s)| + |W_2(s)T(s)| \|_\infty < 1, \qquad (45)$$

where $W_1(s) = \frac{N_{W1}(s)}{D_{W1}(s)}$ and $W_2(s) = \frac{N_{W2}(s)}{D_{W2}(s)}$ are stable weighting functions, and $S(s)$ and $T(s)$ are the sensitivity and the complementary sensitivity functions respectively. As before, let $\delta(s, k_p, k_i, k_d)$ denote the closed loop characteristic polynomial

$$\delta(s,\ k_p,\ k_i,\ k_d) \ \triangleq\ sD(s) + (k_i + k_p s + k_d s^2)N(s).$$

We define the complex polynomial $\psi(s, k_p, k_i, k_d, \theta, \phi)$ by

$$\begin{aligned}\psi(s,\ k_p,\ k_i,\ k_d,\ \theta,\ \phi) \triangleq\ & \\ & sD_{W1}(s)D_{W2}(s)D(s) + e^{j\theta}sN_{W1}(s)D_{W2}(s)D(s)\\ & +(k_d s^2 + k_p s + k_i)[D_{W1}(s)D_{W2}(s)N(s)\\ & +e^{j\phi}D_{W1}(s)N_{W2}(s)N(s)].\end{aligned}$$

As shown in [12], the problem of synthesizing PID controllers for robust performance can be converted into the problem of determining values of $(k_p,\ k_i,\ k_d)$ for which the following conditions hold:

(1) $\delta(s,\ k_p,\ k_i,\ k_d)$ is Hurwitz;
(2) $\psi(s,\ k_p,\ k_i,\ k_d,\ \theta,\ \phi)$ is Hurwitz for all $\theta \in [0,\ 2\pi)$ and for all $\phi \in [0,\ 2\pi)$;
(3) $|W_1(\infty)S(\infty)| + |W_2(\infty)T(\infty)| < 1$.

The following example shows how the above conditions can be used to determine the set of stabilizing gains (k_p, k_i, k_d) for which the robust performance specification (45) is met.

Example 5.2: Consider the plant $G(s) = \frac{N(s)}{D(s)}$ where

$$\begin{aligned} N(s) &= s - 15 \\ D(s) &= s^2 + s - 1. \end{aligned}$$

Then the sensitivity function and complementary sensitivity function are:

$$S(s,\ k_p,\ k_i,\ k_d) = \frac{s(s^2+s-1)}{s(s^2+s-1)+(k_d s^2+k_p s+k_i)(s-15)},$$

$$T(s,\ k_p,\ k_i,\ k_d) = \frac{(k_d s^2+k_p s+k_i)(s-15)}{s(s^2+s-1)+(k_d s^2+k_p s+k_i)(s-15)}.$$

The weighting functions are chosen as: $W_1(s) = \frac{0.2}{s+0.2}$ and $W_2(s) = \frac{s+0.1}{s+1}$. We know that stabilizing $(k_p,\ k_i,\ k_d)$ values meeting the performance specification (45) exist if and only if the following conditions hold:

(1) $\delta(s,\ k_p,\ k_i,\ k_d) = s(s^2+s-1) + (k_d s^2 + k_p s + k_i)(s-15)$ is Hurwitz;

(2) $\psi(s,\ k_p,\ k_i,\ k_d,\ \theta,\ \phi) = s(s+0.2)(s+1)(s^2+s-1) + e^{j\theta}s(0.2)(s+1)(s^2+s-1) + (k_d s^2 + k_p s + k_i)[(s+0.2)(s+1)(s-15) + e^{j\phi}(s+0.2)(s+0.1)(s-15)]$ is Hurwitz for all $\theta \in [0,\ 2\pi)$ and for all $\phi \in [0,\ 2\pi)$;

(3) $|W_1(\infty)S(\infty,\ k_p,\ k_i,\ k_d)| + |W_2(\infty)T(\infty,\ k_p,\ k_i,\ k_d)| = |\frac{k_d}{k_d+1}| < 1.$

The procedure for determining the set of $(k_p,\ k_i,\ k_d)$ values satisfying conditions (1), (2) and (3) is similar to that presented in the previous exampled. First using root loci [2], it was determined that a necessary condition for the existence of stabilizing (k_i, k_d) values is that $k_p \in (-0.5079,\ -0.1155)$. For any fixed $k_p \in (-0.5079,\ -0.1155)$, we use the algorithm of Section V-A to determine the set of $(k_i,\ k_d)$ values satisfying conditions (1) and (2). The condition (3) gives that the admissible set of $(k_i,\ k_d)$ is $\{(k_i,\ k_d)|\ k_i \in \mathcal{R},\ k_d > -0.5\}$. Then for a fixed k_p, we obtain the set of all $(k_i,\ k_d)$ values for which $\||W_1(s)S(s,\ k_p,\ k_i,\ k_d)|+|W_2(s)T(s,\ k_p,\ k_i,\ k_d)|\|_\infty < 1$ by taking the intersection of the set of $(k_i,\ k_d)$ values satisfying conditions (1), (2) and (3). Thus by sweeping over $k_p \in (-0.5079,\ -0.1155)$, and repeating the above procedure, we obtain the set of $(k_p,\ k_i,\ k_d)$ values for which $\||W_1(s)S(s,\ k_p,\ k_i,\ k_d)|+|W_2(s)T(s,\ k_p,\ k_i,\ k_d)|\|_\infty < 1$. This set is sketched in Fig. 13.

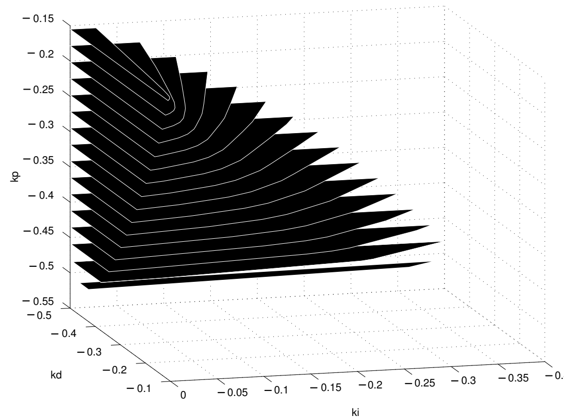

Fig. 13. The set of $(k_p,\ k_i,\ k_d)$ values for which $\||W_1(s)S(s,\ k_p,\ k_i,\ k_d)| + |W_2(s)T(s,\ k_p,\ k_i,\ k_d)|\|_\infty < 1$.

D. PID Controller Design with Guaranteed Gain and Phase Margins

In this subsection, we consider the problem of designing PID controllers that achieve pre-specified gain and phase margins for a given plant. Towards this end, let for example A_m and θ_m denote the desired upper gain and phase margins respectively. From the definitions of the upper gain and phase margins, it follows that the PID gain values $(k_p,\ k_i,\ k_d)$ achieving gain margin A_m and phase margin θ_m must satisfy the following conditions:

(1) $sD(s) + A(k_d s^2 + k_p s + k_i)N(s)$ is Hurwitz for all $A \in [1,\ A_m]$; and

(2) $sD(s) + e^{-j\theta}(k_d s^2 + k_p s + k_i)N(s)$ is Hurwitz for all $\theta \in [0,\ \theta_m]$.

Thus the problem to be solved is reduced to the problem of simultaneous stabilization of two families of polynomials. The algorithm of Section V-A can now be used to solve these simultaneous stabilization problems. The following example illustrates the procedure.

Example 5.3: Consider the plant $G(s) = \frac{N(s)}{D(s)}$ where

$$\begin{aligned} N(s) &= 2s - 1 \\ D(s) &= s^4 + 3s^3 + 4s^2 + 7s + 9. \end{aligned}$$

In this example, we consider the problem of determining all $(k_p,\ k_i,\ k_d)$ gain values that provide a gain margin $A_m \geq 3.0$ and a phase margin $\theta_m \geq 40°$. A given set of (k_p, k_i, k_d) values will meet these specifications if and only if the following conditions hold:

(1) $s(s^4+3s^3+4s^2+7s+9) + A(k_d s^2 + k_p s + k_i)(2s-1)$ is Hurwitz for all $A \in [1,\ 3.0]$;

(2) $s(s^4+3s^3+4s^2+7s+9) + e^{-j\theta}(k_d s^2 + k_p s + k_i)(2s-1)$ is Hurwitz for all $\theta \in [0°,\ 40°]$.

Again, the procedure for determining the set of $(k_p,\ k_i,\ k_d)$ values is similar to that presented in Section V-B and, therefore, a detailed description is omitted. The resulting set is sketched in Fig. 14.

VI. Concluding Remarks

In this paper, we have presented algorithms for determining the set of all PID controllers that stabilize 1) continuous-time rational plants of arbitrary order; 2) discrete-time rational plants of arbitrary order; and 3) continuous-time first order plants with delay. In addition, we showed how some of these algorithms and their extensions can be used to tackle important problems in industrial PID design. The stabilizing sets are neither convex nor even connected in general; nevertheless there is considerable structure available for both computation and design in particular due to the fact

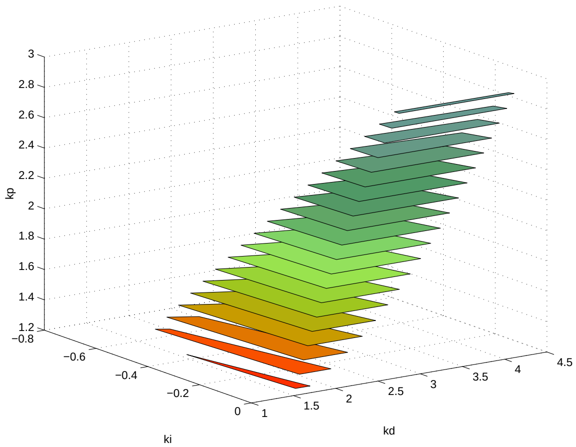

Fig. 14. The set of $(k_p,\ k_i,\ k_d)$ values for which the resulting closed loop system achieves a gain margin $A_m \geq 3.0$ and a phase margin $\theta_m \geq 40°$.

that the regions are bounded by straight lines in the k_i-k_d space for fixed k_p in both, the real and complex cases. This facilitates the complete determination of both stabilizing regions and performance attainment regions and intersecting them, and this feature along with the 2D and 3D graphical displays of these sets should appeal to control designers. In addition to the performance criteria discussed here it is possible to achieve prescribed offset of root locations from the imaginary axis as shown in [13].

The presentation here was motivated by our desire to bring these algorithms to the notice of the industrial control community, whose members, we believe, stand to benefit the most from these results. More motivated development of these algorithms, along with the associated mathematical machinery, can be found in [2], [3], [4], [11], [12], [13]. For an alternative approach digital PID controller design the reader is referred to the recent paper [14].

REFERENCES

[1] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, North Carolina, 1995.

[2] A. Datta, M. T. Ho and S. P. Bhattacharyya, *Structure and Synthesis of PID Controllers*, Springer-Verlag, 2000.

[3] H. Xu, A. Datta and S. P. Bhattacharyya, "Computation of All Stabilizing PID Gains for Digital Control Systems," *IEEE Transactions on Automatic Control*, Vol. AC-46, No. 4, 647-652, April 2001.

[4] 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, 241-252, February 2002.

[5] F. R. Gantmacher, *The Theory of Matrices*, Chelsea Publishing Company, New York, 1959.

[6] L. H. Keel and S. P. Bhattacharyya, "A Generalization of Mikhailov's Criterion with Applications," *Proceedings of American Control Conference,* 4311-4315, Jun. 2000.

[7] A.B.Ozguler and A.A.Kocan, "An Analytic Determination of Stabilizing Feedback Gains" Report 321, Institut fur Dynamische Systeme, Universitat Bremen, Sept. 1994.

[8] L.H.Keel and S.P.Bhattacharyya, "Robust, Fragile or Optimal?" *IEEE Trans. Aut. Contr.*, Vol. AC 42(8), pp.1098-1106,Aug.1997.

[9] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Company, New York, 1992.

[10] S.P.Bhattacharyya, H.Chapellat and L.H.Keel *Robust Control: The Parametric Approach* Prentice Hall, 1995.

[11] M. T. Ho, "Synthesis of H_∞ PID Controllers," *Proceedings of the 40th IEEE Conference on Decision and Control,* 255-260, Dec. 2001.

[12] M. T. Ho and C. Y. Lin, "PID Controller Design for Robust Performance," *Proceedings of the 41st IEEE Conference on Decision and Control,* 1063-1067, Dec. 2002.

[13] J.Ackermann and D.Kaesbauer, "Stable Polyhedra in Parameter Space", *Automatica* Vol. 39(5), pp.937-943, May 2003.

[14] L.H.Keel, J.I.Rego and S.P.Bhattacharyya, "A New Approach to Digital PID Controller Design" *IEEE Trans. Aut. Contr.* Vol. AC-48(4), pp.687-692, April 2003.