Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

Interpolation by polynomials with nonnegative coefficients

Murk J. Bottema
Department of Mathematics and Statistics, IST Building, Flinders University
GPO Box 2100, Adelaide, SA 5001, Australia
murkb@ist.flinders.edu.au

Mark E. Halpern
Department of Electrical and Electronic Engineering, University of Melbourne
Grattan St., Parkville, Vic. 3052, Australia
m.halpern@ee.mu.oz.au

William Moran
Department of Mathematics and Statistics, IST Building, Flinders University
GPO Box 2100, Adelaide, SA 5001, Australia
bill@ist.flinders.edu.au

The authors are participants in the Cooperative Research Centre for
Sensor Signal and Information Processing (CSSIP)

Abstract

We consider the problem of determining whether a polynomial of a given order and having only nonnegative coefficients can be found to interpolate a given set of positive data. This problem arises in the design of maximally robust integrating feedback controllers for linear discrete-time plants and is also relevant to the design of nonovershooting control systems. We present an algorithm for determining whether such a polynomial exists for given interpolation data.

Keywords: polynomial interpolation, extending positive data, feedback control systems, robust stability

1 Introduction

Interpolation problems underly much of linear systems theory. Necessary and sufficient conditions for stability of feedback control systems can be formulated as problems of interpolation by stable rational functions. Designs which are optimal in some sense can be obtained by minimizing an appropriate cost function subject to a finite set of interpolation constraints. For example, in order to obtain stability robustness against norm bounded plant uncertainty, one can separate the system into a feedback interconnection of the plant uncertainty and a closed loop map, which reflects the way the uncertainty enters the plant model. The Small Gain Theorem then motivates the minimization of a norm of the closed loop map in order to allow the uncertainty to be large without loss of stability. This minimization is subject to the interpolation constraints described above. This has been considered in an l_1 norm optimization framework by Dahleh and Pearson [1, 2]. A second example is the minimization of a norm or other functional of the regulated output of a feedback system in order to obtain a design giving optimal performance. This optimization is also subject to these constraints ensuring internal stability.

In this paper we consider the interpolation of a special class of stable rational functions, namely polynomials with only nonnegative coefficients. These have application to both the robust stabilization and optimal performance problems outlined above. For the robust stability problem, Halpern and Evans [3, 4] considered the problem of designing a stabilizing feedback controller with integrating action for linear discrete-time SISO plants to obtain maximal stability robustness against l_1 norm bounded plant numerator coefficient uncertainty. For this problem there is a simply computed upper bound on the stability margin and a class of nominal plants for which a controller can be found to obtain this bound on stability robustness is distinguished by the existence of a nonnegative solution to a set of interpolation equations. The problem of designing feedback systems having no overshoot when tracking a specified command is an important practical one and has been considered in an optimization framework by Deodhare and Vidyasagar [5] and by Hill and Halpern [6]. For a feedback system, zero overshoot corresponds with a nonnegative solution to an interpolation problem.

In section 2, criteria are found for the existence of a polynomial having nonnegative coefficients which interpolates a finite set of data. Leenaerts [7] and Zeheb [8] have used numerical approaches to characterize positive solutions to linear equations, but the problem we consider has interpolation structure which can be exploited. Our problem is equivalent to finding conditions on a vector β such that there is a nonnegative vector x with $Mx = \beta$ where M has a Vandermonde structure. The collection of such vectors β forms a convex set and conditions on β are found by characterizing the hyperplanes that make up the surface of this set.

Conditions on β in the form of inequalities can be found, but if the number of data points is more than three or so, these conditions become numerous and cumbersome. In section 3 an algorithm is described which will settle the question of existence of an interpolating polynomial for any given β.

The geometric view of this interpolation problem is well known (see [10] for example), but has not previously been used (to the authors' knowledge) to construct an algorithm such as presented in section 3.

2 Clams

We consider the following interpolation problem. Given a data set $\{(t_j, \beta_j)\}_{j=0}^{m}$, where the t_j's are distinct and positive, and integer $n \geq m$, decide if there exists a positive vector x such that

$$\sum_{k=0}^{n} x_k t_j^k = \beta_j, \qquad j = 0, 1, \ldots m. \tag{2.1}$$

Remarks

a) By reordering and rescaling if necessary, it suffices to consider data sets with $1 = t_0 > t_1 > \ldots > t_m > 0$ and $\beta_0 = 1$. This structure will be assumed throughout.

b) There is little extra difficulty if the sum in equation (2.1) is infinite instead of finite. Hence the value $n = \infty$ is allowed in what follows.

c) In dimension $m = 2, 3$, the collection of vectors for which (2.1) has nonnegative solution, viewed as a set in R^m, looks somewhat like the shell of a clam (see figure 1). This observation has inspired the following terminology.

Definition 1. Given a set $\{t_j\}_{j=0}^{m}$ as in remark (a) and a positive integer n (or $n = \infty$), the *clam* is the set of vectors $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ such that (2.1) holds for $\tilde{\beta} = (1, \beta_1, \beta_2, \ldots, \beta_m)$ and some positive vector $x \in R^{n+1}$.

In the language of geometrical moment theory, the *clam* is a section (corresponding to $\beta_0 = 1$) of the moment space induced by the Tchebycheff system $\{1, t, t^2, \ldots, t^n\}$ (see [10] page 40). Sketches of the *clam* for $m = 1, 2, 3$ appear in figure 1.

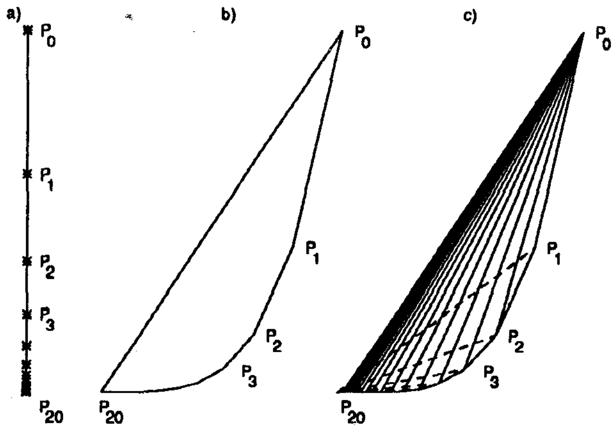

Figure 1: Parts (a), (b) and (c) show sketches of the *clam* in dimensions $m = 1, 2, 3$ respectively. In each case the value $n = 20$ was used. In part (a), $t_1 = .6$. The *clam* is the interval $[P_n, P_0]$. The points P_i are indicated by *. In part (b), $t_1 = .8$, $t_2 = .4$. *TOP* has one element, namely the line segment between P_0 and P_n. The elements of *BOT* are the line segments between P_{i-1} and P_i for $i = 1, 2, \ldots, n$. In part (c), $t_1 = .8$, $t_2 = .6$, and $t_3 = .2$. The hyperplanes in *TOP* are determined by vertices P_0, P_i, and P_{i+1}, for $i = 1, 2 \ldots, n-1$. The hyperplanes in *BOT* are determined by vertices P_n, P_i, and P_{i+1}, for $i = 0, 1 \ldots, n-2$. In higher dimensions (larger m), the surface of the *clam* becomes more and more complex, like the faces of a diamond.

The existence of an interpolation polynomial is now reduced to deciding if, for a data set normalized as in remark (a), the vector $\beta \in R^m$ is in the *clam*. The rest of this section is devoted to describing the *clam*. Theorem 2 says that the clam is a convex polytope and provides a list of its vertices. Theorem 3 describes which hyperplanes determined by the vertices of the clam form the surface of the set and theorem 6 gives criteria for deciding which of these surface hyperplanes form the top and which form the bottom of the *clam*.

Theorem 2. *The clam is the convex hull of the points* $P_k = (t_1^k, t_2^k, \ldots, t_m^k)$, $k = 0, 1, \ldots, n$. *In the case* $n = \infty$, *the point* $P_\infty = (0, 0, \ldots, 0)$ *is not in the clam, but is a vertex of the closure. If* $m = 1$, *only* P_0 *and* P_n *are vertices. For* $m > 1$ *all* P_k *are vertices of the clam.*

Proof. In lieu of remark (a), setting $j = 0$ in (2.1) gives that $\sum_{k=0}^{n} x_k = 1$. Hence a nonnegative solution to (2.1) exists if and only if $\tilde{\beta} = (1, \beta_1, \beta_2, \ldots, \beta_m)$ is in the convex hull of the points $(1, t_1^k, t_2^k, \ldots, t_m^k)$, $k = 0, 1, \ldots, n$. This holds if and only if $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ is in the convex hull of the points $P_k = (t_1^k, t_2^k, \ldots, t_m^k)$, $k = 0, 1, \ldots, n$.

For $m = 1$, the P_k's are just points in the interval $(0, 1]$, so the only vertices of the *clam* are $P_0 = 1$ and $P_n = t_1^n$. For $m > 1$, no three P_k's are collinear, so all P_k's are vertices. ■

This description in terms of vertices is enough to identify which vectors in R^m lie in the *clam* if m is small. For $m = 1$, just check if β_1 is in the interval $[t_1^n, 1]$, and for $m = 2, 3$ it is still relatively easy to check that a given β lies below all the hyperplanes that determine the top of the *clam* and above all the hyperplanes that determine the bottom. For larger m, a closer look is needed to decide which combinations of vertices determine hyperplanes that form the surface of the *clam*.

Notation. If $S = \{k_1, k_2, \ldots, k_m\}$ is a set of distinct nonnegative integers, then H^S denotes the hyperplane determined by the vertices $P_{k_1}, P_{k_2}, \ldots, P_{k_m}$. Note that H^S consists of the points $y = (y_1, y_2, \ldots, y_m)$ such that

$$0 = det \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ y_1 & t_1^{k_1} & t_1^{k_2} & \ldots & t_1^{k_m} \\ y_2 & t_2^{k_1} & t_2^{k_2} & \ldots & t_2^{k_m} \\ \vdots & \vdots & \vdots & & \vdots \\ y_m & t_m^{k_1} & t_m^{k_2} & \ldots & t_d^{k_m} \end{pmatrix}. \tag{2.2}$$

If no points of H^S lie in the interior of the *clam*, then H^S is called a surface hyperplane.

Theorem 3. *(1) For* $n < \infty$, H^S *is a surface hyperplane of the clam if and only if the following condition holds. If* $\{i, i+1, i+2, \ldots, i+r\} \subset S$, $i - 1 \notin S$, $i + r + 1 \notin S$, *then either* $i = 0$ *or* $i + r = n$ *or* r *is odd. (2) For* $n = \infty$, H^S *is a surface hyperplane of the clam if and only if the following condition holds. If* $\{i, i+1, i+2, \ldots, i+r\} \subset S$, $i - 1 \notin S$, $i + r + 1 \notin S$, *then either* $i = 0$ *or* r *is odd.*

In other words, the hyperplanes that contribute to the surface of the *clam* can be distinguished from those that do not by looking at the at the list of indices of the vertices that determine the hyperplane. These indices must occur in strings of even length except that single vertices or strings of odd length can occur at the beginning ($i = 0$) or at the end ($i = n + r$) of the list.

Examples. For $n = 20$ and $m = 5$, the index sets $\{0, 1, 2, 6, 7\}$, $\{3, 4, 13, 14, 20\}$, and $\{0, 1, 13, 14, 20\}$ correspond to surface hyperplanes but the sets $\{1, 2, 3, 6, 7\}$, $\{3, 4, 13, 14, 15\}$, and $\{0, 1, 2, 14, 20\}$ do not.

The proof of theorem 3 depends on the following lemma (see page 221 in [9]).

Lemma 4. *If* f *is a real function given by* $f(s) = c_1 t_1^s + c_2 t_2^s + \ldots + c_k t_k^s$ *with* $1 \geq t_1 > t_2 > \ldots > t_k > 0$, *then* f *has at most* $k - 1$ *roots. If* f *has* $k - 1$ *distinct roots, then each root is simple.*

Proof of theorem 3. The hyperplane $H = H^S$, contributes to the surface of the *clam* if and only if all vertices of the *clam* lie on one side of (or in) H. Consider the function

$$g(\lambda) = det \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ t_1^\lambda & t_1^{k_1} & t_1^{k_2} & \ldots & t_1^{k_m} \\ t_2^\lambda & t_2^{k_1} & t_2^{k_2} & \ldots & t_2^{k_m} \\ \vdots & \vdots & \vdots & & \vdots \\ t_m^\lambda & t_m^{k_1} & t_m^{k_2} & \ldots & t_m^{k_m} \end{pmatrix}.$$

By (2.2), two vertices P_a and P_b lie on the same side of H if and only if

$$sign\ g(a) = sign\ g(b). \tag{2.3}$$

Thus H contributes to the surface of the *clam* if and only if (2.3) holds for all vertices P_a and P_b that do not lie in H: that is, if and only if (2.3) holds for all $a, b \in S^c = \{1, 2, \ldots, n\} \setminus S$.

Clearly $g(k_1) = g(k_2) = \ldots = g(k_m) = 0$. By lemma 4, these are simple roots of g, and g has no other roots. Hence g changes sign at each k_i and nowhere else. Thus (2.3) holds for all $a, b \in S^c$ if and only if g changes sign an even (or zero) number of times between a and b whenever $a, b \in S^c$. This means that the roots of g must occur in strings of even length or in strings that start with $k_1 = 0$ or end with $k_m = n$. ■

Having characterized the hyperplanes that contribute to the surface of the *clam*, it will be convenient to separate them into those that form the "top" and those that form the "bottom" of the set. Here top and bottom are with respect to the last coordinate.

Definition 5. Let H be a hyperplane in R^m and let q be a point in R^m. $H(q)$ will denote the "value of H above q". That is to say, $H(q)$ is the number such that $(q_1, q_2, \ldots, q_{m-1}, H(q)) \in H$.

TOP denotes the set of surface hyperplanes, H, such that $H(\beta) \geq \beta_m$ for all β in the *clam* and BOT is the set of surface hyperplanes, H, with $H(\beta) \leq \beta_m$ for all β in the *clam*.

It will be convenient to view $H(q)$ as the root of the polynomial of degree one defined by

$$g(\lambda) = det\ M_q(\lambda) \tag{2.4}$$

where

$$M_q(\lambda) = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ q_1 & t_1^{k_1} & t_1^{k_2} & \dots & t_1^{k_m} \\ q_2 & t_2^{k_1} & t_2^{k_2} & \dots & t_2^{k_m} \\ \vdots & \vdots & \vdots & & \vdots \\ q_{m-1} & t_{m-1}^{k_1} & t_{m-1}^{k_2} & \dots & t_{m-1}^{k_m} \\ \lambda & t_m^{k_1} & t_m^{k_2} & \dots & t_m^{k_m} \end{pmatrix}. \qquad (2.5)$$

Theorem 6. *Let $H = H^{\{k_1,k_2,\dots,k_m\}}$ be a surface hyperplane of the clam. Let $j \geq 0$ be the smallest integer such that $j \neq k_i$, $i = 1, 2, \dots, m$. (For example, if H is determined by the vertices $P_0, P_1, P_2, P_5, P_6, P_{13}, P_{14}$, then $j = 3$.) If j is even, $H \in BOT$. If j is odd, $H \in TOP$.*

The proof of the theorem depends on the following lemma, the proof of which has been omitted.

Lemma 7. *If $1 \geq t_1 > t_2 > \dots > t_m > 0$ and $0 \leq k_1 < k_2 < \dots < k_m < \infty$, then*

$$det \begin{pmatrix} t_1^{k_1} & t_1^{k_2} & \dots & t_1^{k_m} \\ t_2^{k_1} & t_2^{k_2} & \dots & t_2^{k_m} \\ \vdots & \vdots & & \vdots \\ t_m^{k_1} & t_m^{k_2} & \dots & t_m^{k_m} \end{pmatrix}$$

is positive if $m = 1, 4, 5, 8, 9, 12, 13, \dots$ ($m = 0$ mod 4 or $m = 1$ mod 4) and negative if $m = 2, 3, 6, 7, 10, 11, \dots$ ($m = 2$ mod 4 or $m = 3$ mod 4).

Proof of theorem 6. Let H and j be as in the theorem and let $M_{P_j}(\lambda)$ be as in 2.4 and 2.5. The m-th co-ordinate of vertex P_j has value t_m^j. Hence if $H(P_j) < t_m^j$, then H lies below vertex P_j and since H is a surface hyperplane of the *clam*, $H \in BOT$. Similarly, if $H(P_j) > t_m^j$, then $H \in TOP$.

To prove the theorem, there are 8 cases to consider: $m = l$ *mod* 4 for $l = 0, 1, 2, 3$ with j even and j odd for each l. Only the case of $m = 1$ *mod* 4 and j odd will be presented. The other cases follow similarly.

$g'(\lambda) = (-1)^m det\ M_{P_j}^0$ where $M_{P_j}^0$ is the $m \times m$ matrix obtained by deleting the first column and last row of M_{P_j}. Since $m = 1$ *mod* 4, lemma 7 gives that $det\ M_{P_j}^0 > 0$. Hence,

$$g' = -det\ M_{P_j}^0 < 0. \qquad (2.6)$$

Now consider $g(t_m^j)$. By j column exchanges in $M_{P_j}(t_m^j)$, we get

$$det\ M_{P_j}(t_m^j) = (-1)^j det\ \tilde{M}_{P_j} \qquad (2.7)$$

where $\tilde{M}_{P_j}$ is an $m+1 \times m+1$ matrix in the format considered in lemma 7 (namely the columns are arranged so that the powers of the t's strictly increase from left to right). Since $m = 1$ *mod* 4, $m + 1 = 2$ *mod* 4 and so by lemma 7, $det\ \tilde{M}_{P_j} < 0$. Since j is odd, (2.7) gives that

$$g(t_m^j) = det\ M_{P_j}(t_m^j) = -det\ \tilde{M}_{P_j} > 0.$$

In summary, we have that g is a polynomial of degree 1, g is positive at t_m^j and by (2.6), g has negative derivative. Since $H(P_j)$ is the sole root of g, $H(P_j) > t_m^j$ and so $H \in TOP$. ■

3 Algorithm

To decide if there is polynomial of degree n that interpolates a data set (t_j, β_j), $j = 0, \dots, m$, one must check that the vector β, as normalized in remark (a), lies above all the hyperplanes in BOT and below all the hyperplanes in TOP. Unless m and n are very small, this is not practical.

The following algorithm provides an answer by homing in on the hyperplane in BOT for which $H(\beta)$ is the largest (denoted HB^{max}) and the hyperplane in TOP for which $H(\beta)$ is the smallest (denoted HT^{min}). These hyperplanes have the property that they form the portions of the surface of the *clam* that lie directly below and above β (if β is in the *clam*).

The last statement can be made more explicit as follows. For a surface hyperplane, H, let $H|_{clam} = H \cap cl(clam)$, where cl denotes closure and let E denote projection onto the first $m-1$ coordinates. The relevant properties are:

1. The collection $\{E(H|_{clam} : H \in TOP\}$ tessellates $E(clam)$.

2. The collection $\{E(H|_{clam} : H \in BOT\}$ tessellates $E(clam)$.

3. If $\beta \in clam$ and $\tilde{H}$ is the hyperplane in TOP such that $E(\beta) \in E(\tilde{H}|_{clam})$, then $(\beta_1, \beta_2, \dots, \beta_{m-1}, \tilde{H}(\beta)) \in clam$. Since, for all $H \in TOP$, $H(\bar{\beta}) \geq \bar{\beta}_m$ for all $\bar{\beta} \in clam$, $H(\beta) \geq \tilde{H}(\beta)$. Hence, $\tilde{H} = HT^{min}$.

4. Similarly, if $\beta \in clam$ and $\tilde{H} \in BOT$ is such that $E(\beta) \in E(\tilde{H}|_{clam})$, then $\tilde{H} = HB^{max}$.

The steps for the algorithm are as follows.

1. Starting with the data set (t_j, β_j), normalize so that the conditions in remark (a) are satisfied.

2. Use theorem 6 to select initial surface hyperplanes $HT_0 \in TOP$ and $HB_0 \in BOT$.

3. Construct sequences of hyperplanes $HT_i \in TOP$ and $HB_i \in BOT$ as follows.

3a. If $HT_i(\beta) < \beta_m$, stop. $\beta \notin$ *m-clam.*

3b. If $HB_i(\beta) > \beta_m$, stop. $\beta \notin$ *m-clam.*

3c. Of all hyperplanes in TOP which are adjacent to HT_i, select the hyperplane H for which $H(\beta)$ is the smallest. (How to find adjacent hyperplanes is described below.) If $H(\beta) < HT_i(\beta)$, set $HT_{i+1} = H$ and continue. If $H(\beta) \geq HT_i(\beta)$, set $HT^{min} = HT_i$ and terminate the sequence HT_i.

3d. Similarly, of all in hyperplanes in BOT which are adjacent to BT_i, select the one for which $H(\beta)$ is the largest. If $H(\beta) > HT_i(\beta)$, set $HT_{i+1} = H$ and continue. If $H(\beta) \leq HB_i(\beta)$, set $HB^{max} = HB_i$ and terminate the sequence HB_i.

4. If stopping criteria 3a or 3b are not encountered, finitely many iterations deliver hyperplanes HT^{min} and HB^{max}. In this case, β is in the *clam*.

Several of these steps require some amplification. To begin with, the following hyperplanes can always be used for initialization in step 2. If m is even, $HT_0 = H^{\{0,1,\ldots,m-2,n\}}$ and $HB_0 = H^{\{0,1,\ldots,m-1\}}$. If m is odd, $HT_0 = H^{\{0,1,\ldots,m-1\}}$ and $HB_0 = H^{\{0,1,\ldots,m-1,n\}}$.

Adjacent hyperplanes mentioned in steps 3c,d are hyperplanes that share $m-1$ vertices. By theorem 3, surface hyperplanes of the *clam* are determined by pairs of consecutive vertices with possible exceptions being that vertices P_0 and/or P_n may appear unpaired. Surface hyperplanes adjacent to H can be found by shifting the indices of one pair of vertices of H once to the left or once to the right. There a few rules for identifying pairs of vertices correctly and shifting them to identify adjacent hyperplanes. (1) The vertices P_0 and P_n cannot be part of a pair if they are part of string of odd length. (2) Except as noted in (1), every vertex must be part of exactly one pair. (There is only one way to assign pairs for every set of vertices that determines a surface hyperplane.) (3) If a pair is part of a string of vertices at the beginning (end) of the interval $[0, n]$, then a left (right) shift is not possible for that pair. (4) If a pair is part of an internal string, a left (right) shift requires all members of the string to the left (right) of the pair to be shifted also.

With these restrictions, each set of indices obtained by an allowable shift determines a hyperplane adjacent to H that is in the same collection (TOP or BOT) as H and all such adjacent hyperplanes may be found in this way.

Example. Suppose $n = 13$, $m = 9$ and $S = \{0,1,6,7,8,9,11,12,13\}$. The hyperplane $H^S \in BOT$ since the first nonnegative integer not in S is 2 (even). The pairs are $\{0,1\}$, $\{6,7\}$, $\{8,9\}$, $\{11,12\}$. By rule (1), P_{13} is not part of a pair. The following sets determine all the hyperplanes in BOT that are adjacent to H^S.

$R1 = \{1,2,6,7,8,9,11,12,13\}$
$L2 = \{0,1,5,6,8,9,11,12,13\}$
$R2 = \{0,1,7,8,9,10,11,12,13\}$
$L3 = \{0,1,5,6,7,8,11,12,13\}$
$R3 = \{0,1,6,7,9,10,11,12,13\}$
$L4 = \{0,1,6,7,8,9,10,11,13\}$

Here $R1$ is the set obtained from set S by shifting the first pair to the right, etc. Note that pair 1 cannot shift left and pair 4 cannot shift right, but that all other shifts are allowed. The right shift of pair $\{6,7\}$ in R2 requires shifting the pair $\{8,9\}$ too. Similarly, the left shift of $\{8,9\}$ in L3 requires shifting the pair $\{6,7\}$. The hyperplanes associated with these index sets are in BOT and share $m-1=8$ vertices with H^S.

4 Application to feedback controller design

In this section we show how nonnegative interpolating polynomials arise in feedback controller design problems as indicated in the introduction.

The z-transform, $\hat{h}$, of a sequence, $h = \{h_i\}_{i=0}^{\infty}$, is defined by $\hat{h}(z) = \sum_{i=0}^{\infty} h_i z^i$. With this definition, a stable transfer function has all its poles at values of z such that $|z| > 1$. The symbol z also denotes the unit delay operator and polynomial, X, of order n_x is given by $X(z) = \sum_{i=0}^{n_x} x_i z^i$.

4.1 Application to maximally robust feedback controller design

In this section we outline the problem examined in [3, 4] for the design of an integrating feedback controller giving maximal robustness against plant numerator coefficient uncertainty.

Consider an uncertain SISO discrete-time plant given by

$$\hat{y} = \frac{B^0(z) + \Delta B(z)}{A^0(z)} \hat{u}, \tag{4.1}$$

where $\hat{y}$ is the plant output, $\hat{u}$ is its input, B^0 and A^0 which comprise the nominal plant are known coprime polynomials with $B^0(0) = 0$, $B^0(1) \neq 0$, $A^0(0) = 1$ and $\Delta B(z)$ is an unknown but constant (i.e. time-invariant) numerator uncertainty $\Delta B(z) = \sum_{i=1}^{n_{\Delta B}} \Delta b_i z^i$ with $n_{\Delta B}$ arbitrarily large. The plant has no poles on the stability boundary. With no further loss of generality all of the plant poles are assumed to be unstable ($A(z) = 0 \Rightarrow |z| < 1$). We assume all the plant poles are distinct and satisfy $t_i \in (0,1)$. We order them according to $1 = t_0 > t_1 > \ldots > t_m > 0$ where $z = t_0$ is a pole of the controller. The plant is in an integrating feedback control system with $\hat{u}$ given by

$$\hat{u} = \frac{G(z)}{(1-z)F(z)}(\hat{w} - \hat{y}) \tag{4.2}$$

where $\hat{w}$ is the reference input and $G(z)$ and $F(z)$, with $F(0) = 1$ ensuring properness, are compensator polynomials.

The system closed loop poles are the roots of characteristic polynomial (CP) $V(z)$, given by

$$V(z) = A^0(z)(1-z)F(z)+(B^0(z)+\Delta B(z))G(z). \tag{4.3}$$

We require all the roots of $V(z)$ to be stable i.e. $V(z) = 0 \Rightarrow |z| > 1$, but only know the nominal part of the plant. If the true plant is connected to a compensator designed to give characteristic polynomial $V^0(z)$ with the nominal plant $B^0(z)/A^0(z)$, the closed loop characteristic equation will be

$$V(z) = V^0(z) + \Delta V(z) = 0$$

where from (4.3),

$$\Delta V(z) = G(z)\Delta B(z). \tag{4.4}$$

We wish to design controller polynomials $G(z)$ and $F(z)$ in order to stabilize the nominal plant and to maximize the stability margin ρ given by

$$\rho = \min_{\substack{\exists z_0: |z_0|=1 \\ V(z_0)=0}} \|\Delta b\|_1. \tag{4.5}$$

In other words, for any perturbation Δb such that $\|\Delta b\|_1 < \rho$, the closed loop system is stable.

From [3, 4], for a plant as in (4.1) with any nominal stabilizing integrating controller (4.2), $\rho \leq |B^0(1)|$. A sufficient condition for $\rho = |B^0(1)|$ is that the set of linear equations

$$\sum_{k=0}^{n} x_k t_j^k = \frac{1}{B^0(t_j)}, \qquad j = 0, 1, \ldots, m \tag{4.6}$$

has a solution $x \in l_1$ with $sgn(x) = sgn(B^0(1))$ (or equivalently $\|x\|_1 = |\sum_{i=0}^{n} x_i|$) for sufficiently large n. This is precisely the form of (2.1) with $\beta_j = 1/B^0(t_j)$ for $j = 0, 1, \ldots, m$. Here x is the impulse response of a certain closed loop transfer function namely $G(z)/V^0(z)$. The special case of a nonnegative solution is interesting because results from l_1 optimization and duality theory can be applied to obtain closed form solutions for the controller order and optimal nominal closed loop poles for some simple plants. Details are shown in [3, 4].

4.2 Application to design of nonovershooting controllers

In this application, described in [5, 6], the plant has no uncertainty and the controller is given by

$$\hat{u} = \frac{S(z)\hat{w} - G(z)\hat{y}}{F(z)}.$$

The tracking error $\hat{\phi}$ is given by $\hat{\phi} = \hat{w} - \hat{y}$. The plant has q nonminimum phase zeros $z_1, z_2, \ldots, z_q$ and the command generator $\hat{w}(z)$ has p nonminimum phase zeros $r_1, r_2, \ldots, r_p$. Necessary and sufficient conditions for internal stability that $\hat{\phi}$ must satisfy are $\phi \in l_1$, $\hat{\phi}(z_i) = \hat{w}(z_i)$ for $i = 1, 2, \ldots, q$ and $\hat{\phi}(r_i) = 0$ for $i = 1, 2, \ldots, p$. A simple change of variables allows all coefficients of one equation to be set to one. The algorithm from Section 3 can then be used to obtain necessary conditions on the plant nonminimum phase zeros for no overshoot to be possible.

References

[1] M. A. Dahleh and J. B. Pearson, "Optimal Rejection of Persistent Disturbances, Robust Stability, and Mixed Sensitivity Minimization," *IEEE Trans. Automat. Contr.*, Vol. 33, pp. 722-731, 1988.

[2] M. A. Dahleh, "BIBO Stability Robustness in the Presence of Coprime Factor Perturbations," *IEEE Trans. Automat. Contr.*, Vol. AC-37, pp. 352-355, 1992.

[3] M. E. Halpern and R. J. Evans, "The Design of Maximally Robust Pole-Placement Controllers," *Proceedings of the 3rd European Control Conference*, Vol. 4, Pt 2, pp. 3751-3756, Rome, 1995.

[4] M. E. Halpern and R. J. Evans, "The Design of Maximally Robust Pole-Placement Controllers," Melbourne Uni. Dept. of E & EE internal report 12/94.

[5] G. Deodhare and M. Vidyasagar, "Control system design via infinite linear programming," *Int. J. Contr.*, Vol. 55, pp. 1351-1380, 1992.

[6] R. D. Hill and M. E. Halpern, "Minimum Overshoot Design for SISO Discrete-Time Systems," *IEEE Trans. Automat. Contr.*, Vol. 38, pp. 155-158, 1993.

[7] D. M. W. Leenaerts, "Application of Interval Analysis for Circuit Design," *IEEE Trans. Circuits Syst.*, Vol. 37, pp. 803-807, 1990.

[8] E. Zeheb, "On Solving the Equation $Ax = B$ with More Variables Than Equations," *IEEE Trans. Automat. Contr.*, Vol. AC-39, pp. 833-834, 1992.

[9] J. Kuntzmann, *Méthodes Numerériques Interpolation - Dérivées*, Dunod, Paris, 1959.

[10] S. Karlin and W. J. Studden *Tchebycheff systems: with applications to analysis and statistics*, John Wiley & Sons, 1966.