TP-15 3:00

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

NEW METHODS FOR CONTROL SYSTEM DESIGN USING MATRIX INTERPOLATION

Zhiqiang Gao
Dept. of Electrical Engineering
Cleveland State University
Euclid Ave. at E. 24th St.
Cleveland, Ohio 44115
gao@csvaxe.csuohio.edu

Panos J. Antsaklis
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
Panos.Antsaklis@nd.edu

Abstract: In this paper, it is shown how various familiar problems in control design and system identification can be formulated as interpolation problems, where powerful methods that have been recently developed can be used. This leads to new, more effective, and more practical solutions to classical system and control problems. Numerical issues and computer algorithm development are also discussed.

I. Introduction

Many system and control problems can be formulated in terms of matrix equations where polynomial or rational solutions with specific properties are of interest. It is known that equations involving just polynomials can be solved by either equating coefficients of equal power of the indeterminate s or equivalently by using the values obtained when appropriate values for s are substituted in the given polynomials; in the latter case one uses results from the classical theory of polynomial interpolation. Similarly one may solve polynomial matrix equations using the theory of polynomial matrix interpolation; this approach has significant advantages and it appears in a rather complete form in [1].

In this paper, recent developments in control design using methods based on matrix interpolation theory are outlined. Numerous control system constraints and properties that are expressed in terms of conditions on a polynomial or rational matrix R(s), can be written in an easier to handle form in terms of $R(s_j)$, where $R(s_j)$ is R(s) evaluated at certain (complex) values $s = s_j$ $j=1, \ell$. We shall call such conditions in terms of $R(s_j)$, interpolation (type) conditions on R(s). The relationship between conditions on $R(s_j)$ and properties of R(s) has been addressed in [1] and they have led to a new methodology further explained here which offers alternative approaches to many well known control problems. In particular, in section II, the fundamental results from matrix interpolation theory are outlined. A loop gain design approach based on the interpolation algorithm is introduced in section III; results on a new approach to self-tuning control are presented. An efficient algorithm for system identification in frequency domain is described in section IV. In section V, the Diophantine equation is solved. Other control applications are illustrated in section VI. Computer code development is discussed in section VII.

II. Matrix Interpolation Theory

Certain fundamental results from the theory of polynomial and rational matrix interpolation are briefly outlined here; additional results are included in section V. Full details can be found in [1].

Polynomial Matrix Interpolation:

The basic polynomial interpolation problem can be stated as follows:

Given ℓ distinct complex scalars s_j $j = 1, \ell$ and ℓ corresponding complex values b_j, there exists a unique polynomial q(s) of degree $n = \ell - 1$ for which

$$q(s_j) = b_j \quad j = 1, \ell \tag{2.1}$$

That is, an nth degree polynomial q(s) can be uniquely represented by the $\ell = n+1$ interpolation (points or doublets or) pairs (s_j, b_j) $j = 1, \ell$.

The polynomial matrix interpolation theory deals with this interpolation problem in matrix case. Let $S(s) :=$ blk diag{[1, s, ..., s^{d_i}]'} where d_i $i = 1$, m are non-negative integers; let $a_j \neq 0$ and b_j denote (mx1) and (px1) complex vectors respectively and s_j complex scalars.

Theorem 2.1[1]: Given interpolation (points) triplets (s_j, a_j, b_j) $j = 1, \ell$ and nonnegative integers d_i with $\ell = \sum d_i + m$ such that the $(\sum d_i + m)x\ell$ matrix

$$S_\ell := [S(s_1)a_1, ..., S(s_\ell)a_\ell] \tag{2.2}$$

has full rank, there exists a unique (pxm) polynomial matrix Q(s), with ith column degree equal to d_i, $i = 1$, m for which

$$Q(s_j)\, a_j = b_j \quad j = 1, \ell \tag{2.3}$$

Proof: Since the column degrees of Q(s) are d_i, Q(s) can be written as

$$Q(s) = QS(s) \tag{2.4}$$

where Q $(px(\sum d_i + m))$ contains the coefficients of the polynomial entries. Substituting in (2.3), Q must satisfy

$$QS_\ell = B_\ell \tag{2.5}$$

where $B_\ell := [b_1, ..., b_\ell]$. Since S_ℓ is nonsingular, Q and therefore Q(s) are uniquely determined. □

It should be noted that when $p=m=1$ and $d_1 = \ell - 1 = n$ this theorem reduces to the polynomial interpolation theorem.

Rational Matrix Interpolation:

Similarly to the polynomial matrix case, the problem here is to represent a (pxm) rational matrix H(s) by interpolation triplets or points (s_j, a_j, b_j) $j = 1, \ell$ which satisfy

$$H(s_j)a_j = b_j \quad j = 1, \ell \tag{2.6}$$

where s_j are complex scalars and $a_j \neq 0$, b_j complex (mx1), (px1) vectors respectively.

It is shown[1] that the rational matrix interpolation problem reduces to a special case of polynomial matrix interpolation. To see this:

Write $H(s) = \tilde{D}^{-1}(s)\tilde{N}(s)$ where $\tilde{D}(s)$ and $\tilde{N}(s)$ are (pxp) and (pxm) polynomial matrices respectively. Then (2.6) can be written as $\tilde{N}(s_j)a_j = \tilde{D}(s_j)b_j$ or as

$$[\tilde{N}(s_j), -\tilde{D}(s_j)] \begin{bmatrix} a_j \\ b_j \end{bmatrix} = Q(s_j)c_j = 0 \quad j = 1, \ell \tag{2.7}$$

That is the rational matrix interpolation problem for a pxm rational matrix H(s) can be seen as a polynomial interpolation problem for a p×(p+m) polynomial matrix $Q(s) := [\tilde{N}(s), -\tilde{D}(s)]$ with interpolation points $(s_j, c_j, 0) = (s_j, [a_j', b_j']', 0)\ j = 1, \ell$. There is also the additional constraint that $\tilde{D}^{-1}(s)$ exists.

III. A New Loop Gain Design Approach

Consider the linear, time invariant, SISO control system in Figure 1. P(s) is the transfer function of the plant, C(s) is the compensator to be determined. A classical approach to feedback design is to work directly with the loop gain transfer function, L(s)=P(s)C(s). The stability and performance specifications are interpreted as constraints on the loop gain frequency response, $L(j\omega)=P(j\omega)C(j\omega)$. From these constraints, the designer knows that the $|L(j\omega)|$ should be large up to a frequency and small beyond another frequency; he also knows roughly where the crossover frequency, ω_c, should be from the transient response requirements. The Nyquist stability theorem puts additional constraints on $L(j\omega)$ in terms of the encirclement of the -1+j0 point by the graph of $L(j\omega)$, gain and phase margins, etc.

In the existing loop shaping design techniques, C(s) is determined iteratively such that $L(j\omega)$ satisfies all constraints. This process of finding the appropriate C(s) requires a great deal of human intuition and experience. Compromises are often made in the transient response specifications, the compensator complexity, the actuation limit, and the stability robustness, etc. In the following, a new procedure is introduced.

Problem Formulation[3]

Let the design specifications be expressed as interpolation constraints of the form,

$$L(j\omega_i) = \alpha_i, \quad i = 1, \ell \tag{3.1}$$

where α_i are complex numbers. Note that this is a reasonable assumption since most design specifications such as command following in certain frequency range, crossover frequency, gain and phase margins, stability robustness against high frequency unmodeled dynamics, etc. can be translated as constraints on magnitude and phase of $L(j\omega)$ at a set of frequencies, $\{\omega_i\}$. Thus, the design problem becomes to find a compensator C(s) such that

$$L(j\omega_i) = P(j\omega_i)C(j\omega_i) \qquad i = 1, \ell \tag{3.2}$$

where $P(j\omega_i)$ is the given frequency response of the plant evaluated at ω_i. Write $L(j\omega_i) = \alpha_i$ and $P(j\omega_i) = \beta_i$, $i = 1, \ell$; the numerator and denominator coefficients of C(s) can then be obtained by solving the set of linear algebraic equations

$$C(j\omega_i)\beta_i = \alpha_i, \quad i = 1, \ell \tag{3.3}$$

Given the degree of C(s), α_i, and β_i, solving C(s) from (3.3) can be seen as the rational function interpolation problem in the form of (2.6).

Discussion

One can find a proper C(s) that satisfies all constraints in (3.3) by picking the degree of C(s) high enough. In practice, however, it is often required to find the compensator of lowest order which meets all the specifications. Furthermore, the loop gain constraints are usually indications, rather than absolute criterion, of the open loop frequency response that will lead to satisfactory closed loop performance. Therefore, the design objective can be seen as to find a compensator C(s) of lowest order such that the loop gain frequency response stays in a close neighborhood of the points specified in (3.3). The tolerance of error can be predetermined by the designer and a search algorithm can be used to find the solution. This algorithm will repeatedly solve the interpolation problem (3.3) while increasing the order of C(s) until it finds the solution within the error tolerance. A computer algorithm of this type was developed in [6] for system identification purposes. Modification of it for design purposes is straightforward.

The loop gain constraints in the loop shaping design approach are mostly given as a set of inequalities, such as $|L(j\omega)| > \alpha$, for $\omega_1<\omega<\omega_2$, etc. This is obviously more flexible than the equality constraints shown in (3.1). Note that the solutions are usually least square solutions and they do not solve equation (3.3) exactly. The constraints (3.1) must be selected reasonably so that they can be met by using a relatively simple compensator. For example, if the loop gain is required to have the magnitude decreasing over a frequency range, one should allow the phase to drop over the same range.

A unique feature of this design approach is that it does not require the explicit mathematic model of the plant. To carry out the design, the only information needed from the plant is its frequency response at a set of frequencies, $\{\omega_i\}$. Consequently, not only the major portion of the system identification process is eliminated, but also the errors associated with it. Furthermore, it makes it feasible to implement an automatic design process on-line such that the compensator can be adjusted as the dynamics of the plant changes. This will be addressed later in Section IV.

The extension of the new design method to MIMO system design requires the design specifications be expressed in terms of frequency response of the loop gain transfer matrix L(s) at a set of discrete frequencies, $L(j\omega_i)$, $i=1, \ell$. Once this is accomplished, the same procedure for SISO systems can be applied with few modifications.

Example 3.1 [3]: Consider the feedback control system shown in Figure 1. Assume that P(s) = 1/(s+1)(s+5) and the design specifications are as follows: the cross over frequency be around $\omega=1$ rad/sec; the output disturbance be attenuated at least 40 dB for $\omega \le .01$ rad/sec; the gain and phase margin be above 4 and 30 degrees, respectively; and finally, the system remains stable when there is unmodeled dynamics of the magnitude up to 40 dB for $\omega \ge 10$ rad/sec.

Translating the closed-loop specifications to loop constraints, the crossover frequency and stability margin conditions are directly applied to the loop gain; the disturbance rejection and stability robustness conditions can be interpreted as $|P(j\omega)C(j\omega)|\ge 100$, $\omega\le .01$ rad/sec, and $|P(j\omega)C(j\omega)|<.01$, $\omega\ge 10$ rad/sec. From these constraints, four interpolation pairs are selected as shown in Table 1. Note that the interpolation constraints are selected with some conservatism so that the inaccuracies in the approximate solutions can be tolerated to a certain degree.

Solving the interpolation problem in (3.3), an approximate solution is found

$$C(s) = \frac{-.26s+6.27}{s}$$

The resulting crossover frequency is .9 rad/sec; gain margin is 3.8, phase margin is about 35°; $|P(j\omega)C(j\omega)| = 104$, at $\omega = .01$, $|P(j\omega)C(j\omega)| = .006$ at $\omega = 10$ and dropping.

Table 1 Interpolation constraints

ω rad/sec	.01	1	2	10
$\|P(j\omega)C(j\omega)\|$	100	1	.2	.005
$\arg(P(j\omega)C(j\omega))$	-90°	-130°	-160°	-270°
$\|C(j\omega)\|$	500	10	2.5	.67
$\arg(C(j\omega))$	-84°	-70°	-75°	-120°

Towards Self-Tuning Control[3]

The dynamics of the physical process may change. The change may happen quickly or slowly depending on the nature of the plant. For example, the performance of actuators may degrade slowly with time which corresponds to slow changes in the dynamics of the system. On the other hand, if a failure suddenly occurs in an actuator, it will introduce dramatic variations in the system which corresponds to quick changes in system dynamics. In either situation, the compensator C(s) designed for the original plant P(s) may become ineffective and needs to be adjusted during operation. In the following, a self-tuning control system is proposed to address such problems.

The new design approach discussed above integrate the modeling and design into one process. Once the design specifications are given in terms of loop gain frequency response, the rest of the design can be carried out by a computer algorithm. The frequency response of the plant can be found as the ratio of Fourier transform of the input and output. Or, it can be calculated as

$$P(j\omega_i) = \frac{S_{uu}(j\omega_i)}{S_{uy}(j\omega_i)}$$

where $S_{uu}(j\omega_i)$ and $S_{uy}(j\omega_i)$ are the auto- and cross spectra of the input and output time history. With $L(j\omega_i)$ and $P(j\omega_i)$ given for $i = 1, \ell$, the compensator C(s) is obtained by solving the linear algebraic equation (2.5) on-line.

System Configuration

Based on the above discussion, a conceptual configuration of a self-tuning control system is shown in Figure 2. In this system, the input and output data in time domain is continuously recorded and the frequency response $P(j\omega_i)$ is obtained using the fast Fourier transform (FFT). From the new $\{P(j\omega_i)\}$ and the given constraints on loop gain, $\{L(j\omega_i)\}$, the supervisory control, a higher level decision making mechanism, determines if the compensator should be updated. This is done by comparing the frequency response of the loop gain transfer function at a set of frequencies $\{\omega_i\}$ with the desired one. If the difference exceeds a predetermined limit, the tuning algorithm will be executed. The tuning algorithm receives $P(j\omega_i)$ and $L(j\omega_i)$ from the supervisory control block and determines the new compensator C(s) using the design method discussed in Section II. Thus, as the dynamics of the plant changes, the performance of the closed loop system is maintained by adjusting C(s). This process can be completely automated without human intervention. Furthermore, the computational complexity of the algorithm is expected to be reasonable for on-line operation since it only involves solving a set of linear algebraic equations.

Comparison to self-tuning adaptive control

The proposed tuning method is similar in concept to self-tuning adaptive control, as it is defined in the literature. The objective of both methods is to adjust the compensator to accommodate the changes in the plant. The implementations, however, are very different. The new method has the following unique characteristics

a) It does not estimate the parameters of the plant, directly or indirectly.

b) There is no assumption made on the structure of the compensator. The order of the compensator is determined only to satisfy the design constraints.

c) The compensator is only adjusted when necessary and it is done quickly in one step. For this purpose, a decision making mechanism, perhaps in forms of a rule based system, is required.

IV. Applications in System Identification

The current results in literature on system identification from frequency response all require that the frequency response of the system to be identified, $\tilde{G}(j\omega_i)$, i=1,2, ..., be given. For SISO systems, this does not pose much difficulty as one can always take the ratio of $y(j\omega_i)$ and $u(j\omega_i)$ to obtain the frequency response $\tilde{G}(j\omega_i)$. Unfortunately, it is not so trivial for MIMO systems considering all possible couplings between various inputs and outputs. Therefore, the assumption that $\tilde{G}(j\omega_i)$ is given seems quite restrictive and impractical, particularly for MIMO systems.

The nature of system identification problem dictates that one must work with the measurements $y(j\omega_i)$ and $u(j\omega_i)$, instead of $\tilde{G}(j\omega_i)$. Ideally, the transfer function matrix G(s) should be determined such that it fits the measurements as follows

$$G(j\omega_i)u(j\omega_i) = y(j\omega_i) \quad i = 1, 2, \ldots \tag{4.1}$$

For SISO systems $G(j\omega)$, $u(j\omega)$ and $y(j\omega)$ are scalars; while for MIMO systems $G(j\omega)$ is a matrix; $u(j\omega)$ and $y(j\omega)$ are vectors. The problem of interests is to determine G(s) such that the error, $y(j\omega_i) - G(j\omega_i)u(j\omega_i)$, is minimized in some sense.

It is usually more convenient to deal with polynomial matrices than rational matrices. Assuming G(s) is a pxm rational matrix, let $G(s)=D^{-1}(s)N(s)$ be a left coprime fraction representation, where D(s) and N(s) are (pxp) and (pxm) polynomial matrices, respectively. Equation (4.1) is equivalent to

$$N(j\omega_i)u(j\omega_i) = D(j\omega_i)y(j\omega_i) \quad i = 1, 2, \ldots \tag{4.2}$$

and the error can be defined as

$$E_i = N(j\omega_i)u(j\omega_i) - D(j\omega_i)y(j\omega_i) \quad i = 1, 2, \ldots \tag{4.3}$$

Now the problem can be formulated as follows:

Problem Formulation [6]

Given column degrees of N(s) and D(s), and the input and output measurements, $u(j\omega_i)$ and $y(j\omega_i)$ i=1, 2, ..., find a proper transfer function matrix, $G(s)=D^{-1}(s)N(s)$, such that the cost function $J = \|EW\|_f$ is minimized. Here, the matrix $W = \mathrm{diag}\{w_1, w_2, \ldots\}$, is a diagonal weighting matrix where w_i reflects the weight at frequency ω_i; E is the error matrix defined as $E = [E_1, E_2, \ldots]$, where E_i is defined in (4.3). Note that the column degrees of N(s) must not be greater than those of D(s) for a proper solution transfer function matrix to exist.

Computer algorithms have been developed to solve the SISO and MIMO identification problems using this matrix interpolation method [4,5,6]. The new algorithms are more numerically efficient than the existing ones, because the solutions are obtained by solving a set of linear equations while the existing approaches [8-11] formulate the problem as a nonlinear least square problem and employ mathematical programming techniques to iteratively determine the solution. A simple illustrative example is given as follows:

Example 4.1 A 2x2 transfer function matrix is given as

$$\tilde{G}(s) = \begin{bmatrix} s+1 & 0 \\ 1 & s+2 \end{bmatrix}^{-1} \begin{bmatrix} s & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{s}{s+1} & \frac{2}{s+1} \\ \frac{-s}{(s+1)(s+2)} & \frac{s-1}{(s+1)(s+2)} \end{bmatrix}$$

From matrix interpolation theory, $\ell = \sum d_i + m + p = 7$ measurements can be used to uniquely determine this transfer function matrix. Seven frequencies ω_i $i = 1,7$ are arbitrarily chosen within the range of zero to ten radian/second. Seven arbitrary input, $u(j\omega_i)$ $i = 1,7$, are generated and the output response is obtained as $y(j\omega_i) = \tilde{G}(j\omega_i)u(j\omega_i)$. Using $\{y(j\omega_i), u(j\omega_i), i=1,7\}$, the method determined the G(s). The difference between the coefficients of $\tilde{G}(s)$ and G(s) is in the range of 10^{-15}. More details can be found in [5,6].

V. Solving The Diophantine Equation

The solution methodology is based on certain fundamental theoretical results[1] repeated here for convenience. They are developed first for general polynomial matrix equations and they are later specialized for the Diophantine equation. Consider the equation

$$M(s)L(s) = Q(s) \tag{5.1}$$

where L(s) (txm) and Q(s) (kxm) are given polynomial matrices. Determine the polynomial matrix solutions M(s) (kxt) when they exist.

First consider the left hand side of equation (5.1). Let

$$M(s) := M_0 + \dots + M_r s^r \tag{5.2}$$

and $d_i := \deg_{ci}[L(s)]$ $i = 1, m$. If

$$\hat{Q}(s) := M(s)L(s) \tag{5.3}$$

then $\deg_{ci}[\hat{Q}(s)] = d_i + r$ for $i = 1, m$. According to the basic polynomial matrix interpolation Theorem 2.1, the matrix $\hat{Q}(s)$ can be uniquely specified using $\sum(d_i+r))+m = \sum d_i+m)(r+1)$ interpolation points. Therefore consider ℓ interpolation points (s_j, a_j, b_j) $j = 1, \ell$ where

$$\ell = \sum d_i + m(r+1) \tag{5.4}$$

Let $S_r(s) :=$ blk diag$\{[1, s, \dots, s^{d_i+r}]'\}$ and assume that the $(\sum d_i + m(r+1))\times\ell$ matrix

$$S_{r\ell} := [S_r(s_1)a_1, \dots, S_r(s_\ell)a_\ell] \tag{5.5}$$

has full rank; that is the assumptions in Theorem 2.1 are satisfied. Note that for distinct s_j, $S_{r\ell}$ will have full column rank for almost any set of nonzero a_j [1]. Now in view of Theorem 2.1 the matrix $\hat{Q}(s)$ which satisfies

$$\hat{Q}(s_j)a_j = b_j \quad j = 1, \ell \tag{5.6}$$

is uniquely specified given these ℓ interpolation points (s_j, a_j, b_j). To solve (5.1), these interpolation points must be appropriately chosen so that the equation $\hat{Q}(s)$ ($= M(s)L(s)$) $=$ Q(s) is satisfied:

Write (5.1) as

$$ML_r(s) = Q(s) \tag{5.7}$$

where

$$M := [M_0, \dots, M_r] \quad (kxt(r+1))$$
$$L_r(s) := [L(s)', \dots, s^r L(s)']' \quad (t(r+1)xm).$$

Let $s = s_j$ and postmultiply (5.7) by a_j $j = 1, \ell$; note that s_j and a_j $j = 1, \ell$ must be so that $S_{r\ell}$ above has full rank. Define

$$b_j := Q(s_j)a_j \quad j = 1, \ell \tag{5.8}$$

and combine the equations to obtain

$$ML_{r\ell} = B_\ell \tag{5.9}$$

where

$L_{r\ell} := [L_r(s_1)a_1, \dots, L_r(s_\ell)a_\ell]$ $(t(r+1)x\ell)$ and

$B_\ell := [b_1, \dots, b_\ell]$ $(kx\ell)$.

Theorem 5.1: Given L(s), Q(s) in (5.1), let $d_i := \deg_{ci}[L(s)]$ $i = 1, m$ and select r to satisfy

$$\deg_{ci}[Q(s)] \le d_i + r \quad i = 1, m \tag{5.10}$$

Then a solution M(s) of degree r exists if and only if a solution M of (5.9) does exist; $M(s) = M\,[I, sI, \dots, s^r I]'$. □

It is not difficult to show that solving (5.9) is equivalent to solving

$$M(s_j)c_j = b_j \quad j = 1, \ell \tag{5.11}$$

where

$$c_j := L(s_j)a_j,\ b_j := Q(s_j)a_j \quad j = 1, \ell \tag{5.12}$$

M(s) that satisfy (5.11) are obtained by solving

$$MS_{r\ell} = B_\ell \tag{5.13}$$

where $S_{r\ell} := [S_r(s_1)c_1, \dots, S_r(s_\ell)c_\ell]$ $(t(r+1)x\ell)$, with $S_r(s) := [I, sI, \dots, s^r I]'$ $(t(r+1)xt)$ and $B_\ell := [b_1, \dots, b_\ell]$ $(kx\ell)$. Solving (5.13) is an alternative to solving (5.9).

The Diophantine Equation

An important case of (5.1) is the Diophantine equation:

$$M(s)L(s) = [X(s), Y(s)]\begin{bmatrix} D(s) \\ N(s) \end{bmatrix} = Q(s) \tag{5.14}$$

where the polynomial matrices D(s), N(s) and Q(s) are given and X(s), Y(s) are to be found. Theorem 5.1 guarantees that all solutions of (5.14) of degree r are found by solving (5.9) (or (5.13)). In the theory of Systems and Control the Diophantine equation used involves a matrix $L(s) = [D'(s), N'(s)]'$ which has rather specific properties. These are exploited to solve the Diophantine equation and to derive conditions for existence of solutions to (5.14) of degree r.

Theorem 5.2: Let r satisfy

$$\deg_{ci}[Q(s)] \le d_i + r \quad i = 1, m \text{ and } r \ge \nu - 1.$$

Then the Diophantine equation (5.14) has solutions of degree r, which can be found by solving (5.9).

Example 5.1: Let

$$D(s) = \begin{bmatrix} s-2 & 0 \\ 0 & s+1 \end{bmatrix}, N(s) = \begin{bmatrix} s-1 & 0 \\ 1 & 1 \end{bmatrix}, Q(s) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

From which $d_1 = d_2 = 1$; $\deg_{ci}Q(s) = 0$, $i=1, 2$; and $\ell = 2 + 2(r+1)$

For $r = 1$, $s_j = -2, -1, 0, 1, 2, 3$ and

$$a_j = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

a solution is

$$[X(s), Y(s)] = \begin{bmatrix} s & -1 & -s & s+1 \\ 1/3 & 1/3 & 0 & -1/3s+2/3 \end{bmatrix}. \qquad \square$$

Numerical Considerations:

Recent study[12] shows that the polynomial matrix interpolation method(PIM) is computational very efficient compared to other methods in the literature, including the elementary operations method(EOM) and the state-space realization method(SSM). It performed satisfactorily in solving both well-conditioned and ill-conditioned Diophantine equations. The benchmark problem used to test all three algorithms was defined as follows:

$$D(s) = \begin{bmatrix} 3(s+1) & s^2-1 \\ 0 & s^2-1 \end{bmatrix}, Q(s) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and $N(s) = \begin{bmatrix} (1+e)^2 s^2 + (1+e)s & 2(1+e)^2 s^2 - 2 \\ (1+e)^2 s^2 - 1 & (1+e)^2 s^2 - 2 \end{bmatrix}$

Note that when e=0, there exists a nontrivial greatest common right divisor which makes the equation unsolvable. Furthermore, the equation becomes ill-conditioned when $e \to 0$.

The initial results in [12] indicate that PIM is significantly more efficient than EOM and SSM. The Matlab codes used in [12] were taken from [1] with little modification.

VI. Other Control Applications

Output Feedback

All proper output feedback controllers of degree r (of order mr) that assign all the closed loop eigenvalues to arbitrary locations are characterized in a convenient way using interpolation results. We are interested in solutions [X(s), Y(s)] (mx(p+m)) of the Diophantine equation where only the roots of $|Q(s)|$ are specified; furthermore $X^{-1}(s)Y(s)$ should exist and be proper. Here the equation to be solved is

$$(X(s_j) D(s_j) + Y(s_j)N(s_j))a_j = 0 \quad j = 1, \ell \tag{6.1}$$

or $ML_{T\ell} = 0$ ($\ell = \Sigma d_i + mr$) ; that is the Σd_i + mr roots of $|X(s) D(s) + Y(s)N(s)|$ are to take on the values s_j $j = 1, \ell$. Note the difference between the problem studied in Section IV, where Q(s) is known, and the problem studied here where only the roots of $|Q(s)|$ (or $|Q(s)|$ within multiplication by some nonzero real scalar) are given. It is clear that there are many (in fact an infinite number) of Q(s) with the desired roots in $|Q(s)|$. So if one selects in advance a Q(s) with desired roots in $|Q(s)|$ that does not satisfy any other design criteria as it is typically done, then one really solves a more restrictive problem than the eigenvalue assignment problem. It is shown here that one does not have to select Q(s) in advance. The vectors a_j can then be seen as design parameters and they can be selected almost arbitrarily to satisfy requirements in addition to pole assignment.

Theorem 6.1 Let $r \geq \nu-1$. Then (X(s), Y(s)) exists such that all the n+mr zeros of $|X(s) D(s) + Y(s)N(s)|$ are arbitrarily assigned and $X^{-1}(s)Y(s)$ is proper. □

Example 6.1:

Let $D(s) = \begin{bmatrix} s-2 & 0 \\ 0 & s+1 \end{bmatrix}$, $N(s) = \begin{bmatrix} s-1 & 0 \\ 1 & 1 \end{bmatrix}$

with n = deg$|D(s)|$ = 2. Here there are deg$|X(s)D(s) + Y(s)N(s)|$ = n + mr = 2 + 2r closed-loop poles to be assigned. Note that $r \geq \nu - 1 = 1 - 1 = 0$.

i) For r = 0 and $\{(s_j, a_j), j = 1,2\} = \{(-1, [1\ 0]'), (-2, [0\ 1]')\}$, a solution of $ML_{T\ell} = 0$ is

$M = \begin{bmatrix} 2 & 0 & -3 & 0 \\ 0 & 2 & 1 & 2 \end{bmatrix}$. For this case, M = M(s) = [X(s) Y(s)].

ii) For r = 1, and $\{(s_j, a_j), j = 1,4\} = \{(-1, [1\ 0]^T), (-2, [0\ 1]^T), (-3, [-1\ 0]^T), (-4, [0\ -1]^T)\}$

a solution of $ML_{T\ell} = 0$ is

$[X(s)\ Y(s)] = \begin{bmatrix} s-7 & -1 & 12 & s+1 \\ 5 & s+4 & -6 & s+4 \end{bmatrix}$

Note that $X(s)^{-1}Y(s)$ exists and it is proper. □

Choosing a Closed Loop Transfer Function Matrix

One of the challenging problems in control design is to choose an appropriate closed loop transfer function matrix that satisfies all the control specifications, which can be obtained from the given plant by applying an internally stable feedback loop. To guarantee the internal stability of feedback control systems, both locations and zero directions of the RHP zeros of the plant must be considered; these zeros must appear as zeros of the closed loop transfer function matrix. Consider the stable model matching problem [15]:

Given proper rational matrices P(s) (pxm) and T(s) (pxq), find a proper and stable rational matrix M(s) such that the equation

$$P(s)M(s) = T(s) \tag{6.2}$$

holds. Here, P(s) usually represents the open loop plant, M(s) the open loop compensator, and T(s) the desired transfer function. It is known that a stable solution for (6.2) exists if and only if T(s) has as it zeros all the RHP zeros of P(s) together with their directions[15]. Such conditions can be easily interpreted as interpolation constraints as shown below.

Let the coprime fraction representations of H(s) and T(s) be $P(s) = N(s)D^{-1}(s)$ and $T(s) = N_T(s)D_T^{-1}(s)$. The direction associated with a zero of P(s), z_j, is given by the vector a_j which satisfies $a_jN(z_i) = 0$. T(s) will have the same zero, z_i, together with its direction i.e. $a_jN_T(z_i) = 0$ and this must be taken into consideration when T(s) is selected.

Example 6.2: Consider a diagonal T(s); that is the control specifications demand diagonal decoupling of the system. Let

$$P(s) = \frac{1}{s+1}\begin{bmatrix} s-1 & 0 \\ 1 & 1 \end{bmatrix}$$

with a zero at s=1. Then aP(1)=0 gives a=[1 0] and T(s) must satisfy aT(1)=[1 0]T(1)=0. Since T(s) must be diagonal, $t_{11}(1) = 0$; that is the RHP zero of the plant should appear in the (1,1) entry of T(s). Any T(s) satisfying such condition guarantees that a stable solution exists for (6.2). □

Applications to 2-D Matrix Interpolation Problems

In many system and control applications, one has to deal with polynomial and rational matrices of two or more independent variables[15]. Recent results in [14] have shown that the matrix interpolation theory for single variable, s, can be easily extended to two dimensional systems where the polynomial and rational matrix are of the form P(s,z) and H(s,z), respectively. The definition of the interpolation problem and the existence of the solutions were all developed along similar lines to those described in section II.

VII. Development of Computer Algorithms

Several Matlab programs based on matrix interpolation theory have been developed to solve a variety of problems. The Matlab programs developed to solve the basic interpolation problem defined in (2.2) to (2.5) were presented in the technical report version of [1], together with the routines that solve the polynomial equation of the form M(s)L(s)=Q(s) and the Diophantine equation of the form X(s)D(s)+Y(s)N(s)=Q(s). These programs were used to find solutions for many examples in [1]. Later, a comprehensive Matlab program was developed for the purpose of SISO system identification in frequency domain. Details can be found in [4]. This program was compared favorably against an existing Fortran program[13].

Based on the encouraging results from [4], a Matlab program was developed in [5] for MIMO system identification in frequency domain, which has not been adequately addressed in literature. The few existing publications concerning this issue [10-12] were rather brute force extensions of the SISO approach which were quite computational intensive. The new Matlab program offers much improved efficiency and is more practical since it does not require that the frequency response of the system be given. It determines transfer function matrix from a set of input-output data in frequency domain. To improve the numerical property of the algorithm, the Chebyshev polynomials are used as the basis.

Currently, software development for the loop gain design approach discussed in section III is underway at Cleveland State University. It will play a key role in the proposed self-tuning control system, which has many advantages over the adaptive control approach.

Conclusion

Interpolation appears to offer a very flexible and efficient set of tools to solve system and control problems. The results up to now have been extremely encouraging.

REFERENCES

[1] P.J. Antsaklis and Z. Gao, "Polynomial and Rational Matrix Interpolation: Theory and Control Applications", Control Systems Technical Report No. 71, Dept. of Electrical Engineering, Univ. of Notre Dame, September 1992. International Journal of Control, Vol 58, No.2, pp.349-404, 1993.

[2] P.J. Antsaklis and R.L. Lopez, "Control Design Using Polynomial Matrix Interpolation", Proc. of the 23rd IEEE Conf. on Decision and Control, pp 621-622, Las Vegas, Nevada, Dec. 12-14, 1984.

[3] Z. Gao, "An algorithmic approach to loop shaping with applications to self-tuning control systems", Proceedings of the 1993 IEEE International Symposium on Intelligent Control, Chicago, August 1993.

[4] R.V. Savescu, "System Identification from Frequency Response Using Matrix Interpolation Theory," M.S. Thesis, Dept. of Electrical Engineering, Cleveland State University, Oct. 1992.

[5] B. Tabachnik, "A Computer Algorithm for Multi-input multi-output system identification Using Chebyshev Polynomials", M.S. Thesis, Dept. of Electrical Engineering, Cleveland State University, to be completed.

[6] Z. Gao, B. Tabachnik, and R. V. Savescu, " A New Approach to Multi-input multi-output system identification in Frequency Domain", Submitted for publication.

[7] G.F. Franklin, J.D. Powell and M.L. Workman, Digital Control of Dynamic Systems, 2nd Edition, Addison-Wesley Publishing Company, Reading, MA, 1990.

[8] D.S. Bayard, "Multivariable Frequency Domain Identification Via 2-Norm Minimization", Proceedings of the American Control Conference, 1992.

[9] R.L. Dailey, and M.S. Lukich, "MIMO transfer function curve fitting using Chebyshev polynomials," SIAM 35th Anniversary Meeting, Denver, CO., 1987.

[10] P.L. Lin and Y.C. Wu, "Identification of multi-input multi-output linear systems from frequency response data," Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 104, March 1982.

[11] R.C. Seidel, "Transfer function parameter estimation from frequency response data - A Fortran program," NASA Tech. Memo. Washington, D.C., X-3286, Sept. 1975.

[12] F. Kraffer, S. Pejchova, and M. Sebek, "Polynomial Operations: Numerical Performance in Matrix Diophantine Equation," Submitted for publication.

[13] Z. Gao and P.J. Antsaklis, "On Stable Solutions of the O... and Two Sided Model Matching Problems", IEEE Transactions on Automatic Control, Vol.34, No.9, pp. 978 - 982, September 1989.

[14] T. Kaczorek, "Polynomial and Rational Matrix Interpolation for 2-D Systems," IMACS Symposium on , Lille, April 25-27, 1994.

[15] T. Kaczorek, Two-Dimensional Linear Systems, Berlin Springer, 1985.

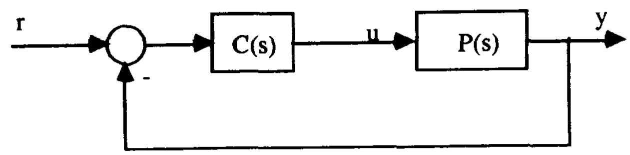

Figure 1 A feedback control system

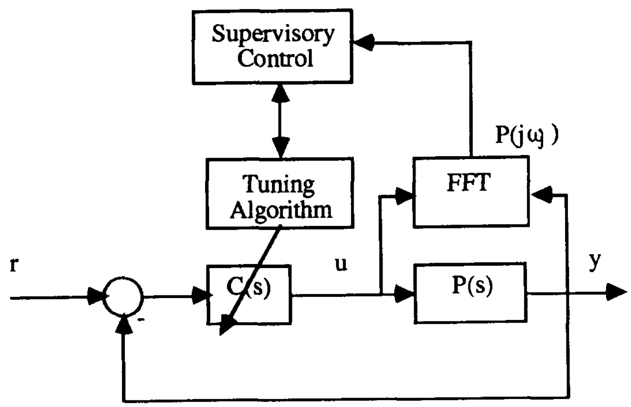

Figure 2 A self-tuning control system