1988

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

INVESTIGATION OF PRACTICAL APPLICATIONS OF H_∞ CONTROL THEORY TO THE DESIGN OF CONTROL SYSTEMS FOR LARGE SPACE STRUCTURES

Prepared by:	R. Dennis Irwin
Academic Rank:	Assistant Professor
University and Department:	Ohio University Department of Electrical and Computer Engineering
NASA/MSFC:	
Laboratory:	Structures and Dynamics
Division:	Control Systems
Branch:	Pointing Control Systems
MSFC Colleague:	Henry Waites
Date:	September 13, 1988
Contract:	NGT 01-002-099 The University of Alabama

INVESTIGATION OF PRACTICAL APPLICATIONS OF H_∞ CONTROL THEORY TO THE DESIGN OF CONTROL SYSTEMS FOR LARGE SPACE STRUCTURES

by

R. Dennis Irwin
Assistant Professor of Electrical and Computer Engineering
Ohio University
Athens, Ohio

ABSTRACT

The work documented here involves the investigation of the applicability of H_∞ control theory to the unique problems of large space structure(LSS) control. A complete evaluation of any technique as a candidate for large space structure control involves (1) analytical evaluation, (2) algorithmic evaluation, (3) evaluation via simulation studies, and (4) experimental evaluation. This report documents the results of analytical and algorithmic evaluations.

The analytical evaluation involves the determination of the appropriateness of the underlying assumptions inherent in the H_∞ theory, the determination of the capability of the H_∞ theory to achieve the design goals likely to be imposed on an LSS control design, and the identification of any LSS specific simplifications or complications of the theory. The results of the analytical evaluation are presented in the form of a tutorial on the subject of H_∞ control theory with the LSS control designer in mind.

The algorithmic evaluation of H_∞ for LSS control pertains to the identification of general, high level algorithms for effecting the application of H_∞ to LSS control problems, the identification of specific, numerically reliable algorithms necessary for a computer implementation of the general algorithms, the recommendation of a flexible software system for implementing the H_∞ design steps, and ultimately the actual development of the necessary computer code. The results and status of the algorithmic evaluation are presented.

Finally, the state of the art in H_∞ applications is summarized with a brief outline of the most promising areas of current research. Recommendations on further work are included, with emphasis on the LSS control problem.

INTRODUCTION

The original concept of an H_∞ performance criterion is that of Zames [1], who introduced the concept of minimum sensitivity and provided a solution in the SISO case. However it is highly probable that the need for such an optimization criterion has at various times been recognized by frequency domain oriented control engineers for quite some time. In any event, Zames motivation appears to be due to the recognition by the late 1970's that the by then well established LQG approach to control design suffered from fundamental limitations with regard to robustness(sensitivity). The fact that LQG based designs are inherently non-robust was quite surprising due to the fact that full state feedback quadratic optimal control designs have long been known to possess excellent robustness properties in the SISO case [2] and Safanov and Athans [3] had recently proved a corresponding result for MIMO full state feedback designs. The unexpected and unwanted result that the addition of a Kalman filter estimator can lead to arbitrarily small stability margins even in the SISO case was dramatically illustrated by an example of Doyle [4]. This led to abandonment of indirect approaches to achieving robustness and ultimately to the direct solution of the robustness optimization (minimum sensitivity, H_∞) problem by Zames.

The solution to the more difficult MIMO H_∞ optimization problem was obtained first by Francis, Helton and Zames [5] in 1984, who credit Cheng and Pearson [6] with an independent solution. Whatever the origin of the final solution, the stabilizing controller parametrization of Desour et al.[7] was the breakthrough that turned the H_∞ optimization problem into a practical control design approach, as opposed to a mathematical excercise in well established interpolation theory. In fact, it is expected that the mathematical tables will now be turned and, rather than the controller parametrization yielding practical applications of known mathematical theory, it will now motivate research into new mathematical optimization problems. This is the philosophy behind Vidyasagar's book on the factorization approach to control system synthesis [8].

While the "complete" solution to the original H_∞ optimization problems exists, work continues on the problem of controller order reduction, the complexity of the required algorithms, the possible equivalence to certain restricted classes of LQG problems(frequency weighted approaches, in particular), and the problem of simultaneously achieving both performance and robustness.

The approach taken here is pragmatic. H_∞ will be evaluated based on what can be done at the present time, with emphasis on the LSS problem, in regard to both the analytical issues of applicability and the practical issues of software development. An outline of the H_∞ philosophy and theory is presented in a tutorial fashion. General algorithms that are well established are then presented and the actual numerical algorithms required are outlined in detail. Finally, possible design approaches appropriate for the LSS problem are outlined and design studies are suggested.

H_∞ THEORY

A Standard Problem

Although the original H_∞ problems were minimum sensitivity problems, a number of classical control design problems can be cast in the same form. The standard problem formalization is a way of casting any one of the problems of disturbance attenuation, command tracking, and robust stabilization into the form of a disturbance attenuation problem. The general form of the standard problem is given in block diagram form in Figure 1, where W(s) is a vector of command and/or disturbance inputs, Z(s) is a vector of regulated outputs which may be either tracking errors or actual regulated quantities, U(s) is a vector of controller outputs, and Y(s) is a vector of available measurements.

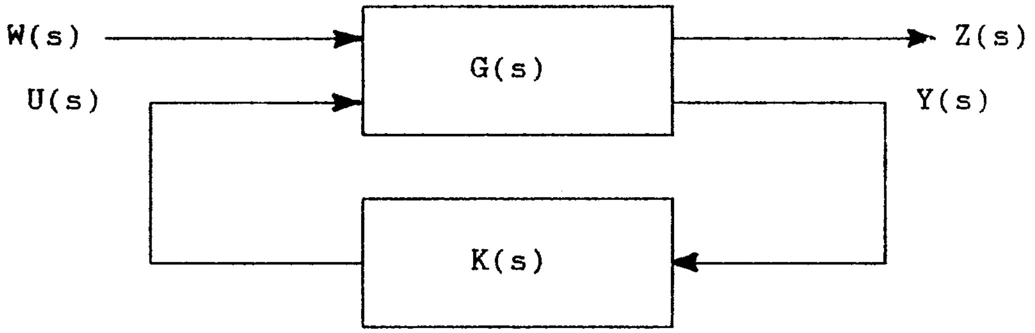

Figure 1

The relationships between the various signals are

$$Z = G_{11}W + G_{12}U, \tag{1}$$

$$Y = G_{21}W + G_{22}U, \tag{2}$$

and

$$U = KY. \tag{3}$$

These equations can be represented in the form of the more detailed block diagram of Figure 2.

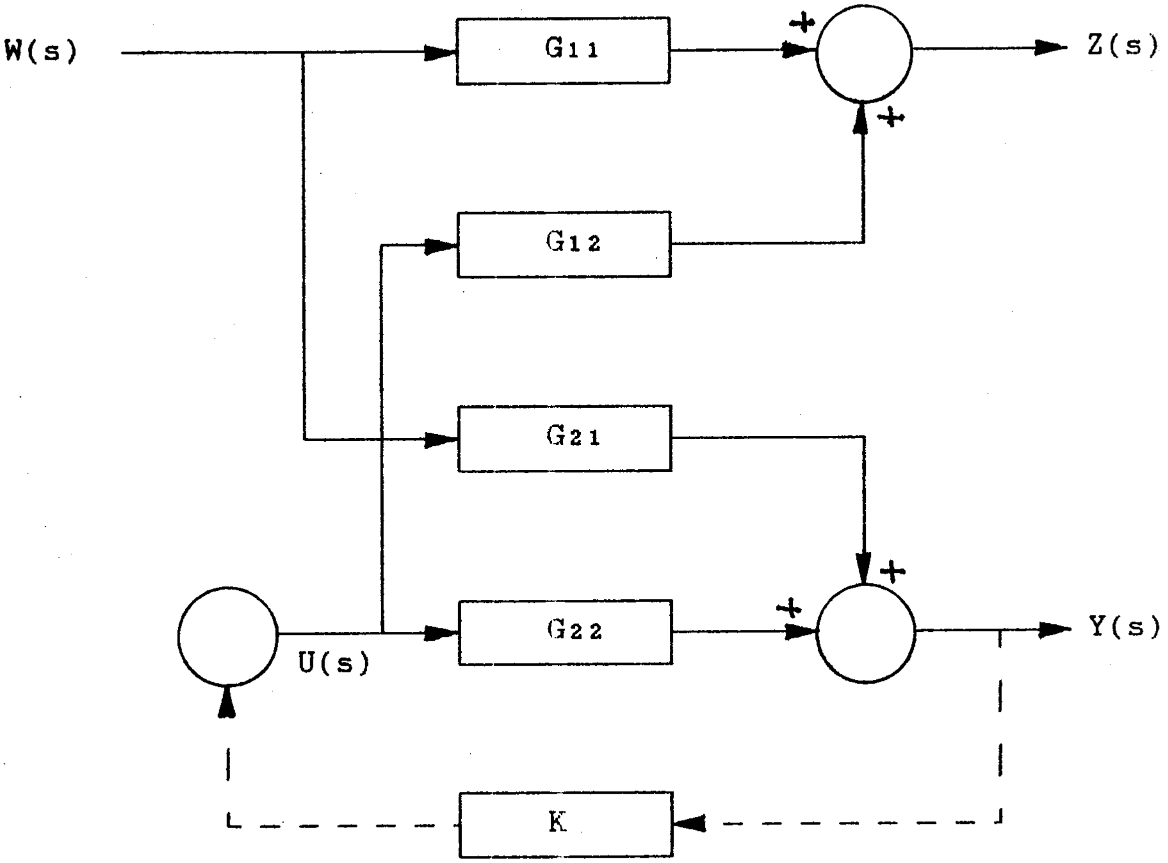

Figure 2

The significance of the standard problem is that it provides a way of unifying several different problems into one problem which can be solved via H∞ optimization techniques. The approach to casting a particular problem in standard form is to (1) define the signals Z, W, Y, and U in terms of the appropriate signals of the actual system at hand, (2) write equations for these signals based on the interconnections of the actual system, and (3) rewrite these equations in the form of Equations (1) - (3). As an example casting a simple problem in standard form consider the system of Figure 3, where the general design goal is to minimize the tracking error R - C.

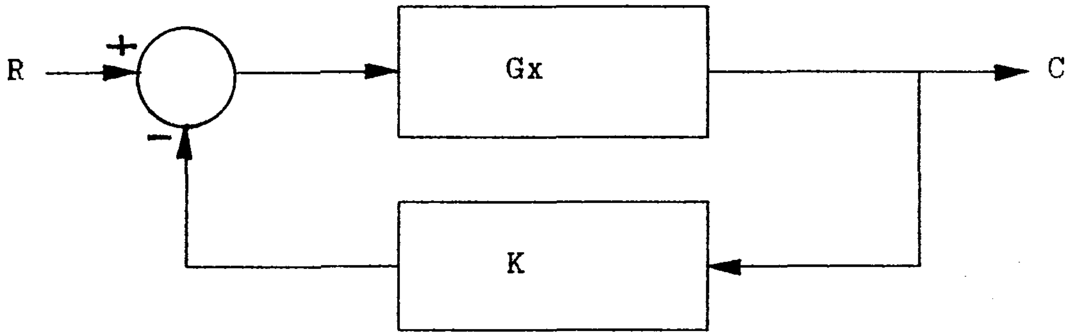

Figure 3

In Figure 3, the following definitions are made:

$Z = R - C$

$W = R$

$U = KC$

$Y = C$

Then

$Z = W - Gx(W - U) = (1 - Gx)W + GxU$

$Y = Gx(W - U)$

$U = KY$

which are in the form of a standard problem.

Mathematical Preliminaries

Before continuing with the parametrization of all stabilizing controllers, some quick definitions are needed.

DEFINITION

The space RH_∞(scalar case) consists of all those rational functions with real coefficients which are stable and $| G(\infty) | < \infty$. For functions in RH_∞, the norm can be calculated via

$$\| F \|_\infty = \sup\{ |F(jw)| : w \text{ real} \}. \tag{4}$$

DEFINITION

The space RH_∞(multivariable case) consists of all those matrices whose entries are stable rational functions with real coefficients and $\| G(\infty) \|_2 < \infty$. For matrices in RH_∞ the norm can be calculated as

$$\| G \|_\infty = \sup_w \{ \| G(jw) \|_2 \} \qquad (5)$$

DEFINITION

The space RH_2(scalar case) consists of all those rational functions with real coefficients which are stable and $| G(\infty) | = 0$. For functions in RH_2, the norm can be calculated via

$$\| F \|_2 = \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} |F(jw)|2 \, dw \right]^{1/2} . \qquad (6)$$

DEFINITION

The space RH_2(multivariable case) consists of all those vectors whose entries are stable rational functions with real coefficients and whose elements are in RH_2. For vectors in RH_2 the norm can be calculated as

$$\| F \|_2 = \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} \|F(jw)\|2 \, dw \right]^{1/2} . \qquad (7)$$

Simply stated, RH_∞ is the space of stable transfer functions (either scalar or multivariable) and RH_2 is the space of finite energy signals (scalar or vector valued). In H_∞ optimal control, the optimality criterion for choosing the controller K is to minimize

$$\| T \|_\infty = \sup_{w}\{ \| T(jw) \|_2 \}$$

where T(s) is the closed loop transfer function from W(s) to Z(s) in the standard problem. The significance of this optimality criterion stems from the fact that

$$\| T \|_\infty = \sup\{ \| TW \|_2 \mid \| W \|_2 = 1 \} \qquad (8)$$

which simply means that our goal is to minimize the output energy of the standard system for the "worst" input of unit energy. This goal is in contrast to the usual quadratic optimality criterion in which the goal is to minimize the output energy for a single, particular input.

The fact that H_∞ optimization is not appropriate for all control design problems is seldom pointed out. However, in cases in which the system inputs (disturbances) are well known and the system model is also well known, there is little justification for the additional complexity of H_∞ optimal control design. Indeed, in this case the usual quadratic optimal control approach should yield better performance than that of an H_∞ approach.

Stabilizing Controller Parametrization

The mathematical theory on which H_∞ optimization is based requires that the optimality criterion be stated in a form that requires only the choice between a set of transfer functions or matrices in RH_∞. However, once the optimal transfer function out of this set is chosen, it must in some way yield a controller which also internally stabilizes the system. The stabilizing controller parametrization accomplishes this goal by providing a one to one correspondence between elements of RH_∞ and all stabilizing controllers.

The controller parametrization is based on factorization theory in RH_∞. A simplified derivation in the scalar case can be found in Irwin [9]; the complete derivation in the general case can be found in Vidyasagar [8] or Francis [10].

DEFINITION

A doubly co-prime factorization of a transfer function matrix G is a set of matrices in RH_∞ which satisfy

$$G = NM^{-1} = M_1^{-1}N_1 \qquad (9)$$

$$\begin{bmatrix} X_1 & -Y_1 \\ -N_1 & M_1 \end{bmatrix} \begin{bmatrix} M & Y \\ N & X \end{bmatrix} = I \qquad (10)$$

It turns out that a doubly co-prime factorization of G_{22} is all that is needed to parametrize both the set of all stabilizing controllers and all stable transfer functions from W to Z. It is assumed that stabilizing G_{22} is equivalent to stabilizing G in the standard problem. The calculation of a doubly co-prime factorization is addressed in Nett et al. [11].

The end result of the controller parametrization is that all controllers which stabilize the system of the standard problem can be expressed as

$$\begin{aligned} K &= (Y - MQ)(X - NQ)^{-1} \\ &= (X_1 - QN_1)^{-1}(Y_1 - QM_1). \end{aligned} \qquad (11)$$

where Q is some matrix in RH_∞.

The parametrization of Equation 11 has significance well beyond the H_∞ optimal control problem. For example, it provides a simple stability check for a given candidate controller K; simply solve Equation 11 for Q and if Q is in RH_∞ the controller stabilizes the system and if Q is not in RH_∞ the controller does not stabilize the system.

The closed loop transfer function matrix of the system of the standard problem can also be parametrized with Q as the parameter:

$$Z = [T_1 - T_2QT_3]W \qquad (12)$$

where

$$T_1 = G_{11} + G_{12}MY_1G_{21} \qquad (13)$$

$$T_2 = G_{12}M \qquad (14)$$

$$T_3 = M_1G_{21}. \qquad (15)$$

For systems which are open loop stable a valid doubly co-prime factorization is

$Y = 0$

$Y_1 = 0$

$X = I$

$X_1 = I$

$M = I$

$M_1 = I$

$N = G_{22}$

$N_1 = G_{22}$

so that a stabilizing controller is given by

$$K = -Q(I - G_{22}Q)^{-1}$$

and the closed loop transfer function matrix parametrization is

$T_1 = G_{11}$

$T_2 = G_{12}$

$T_3 = G_{21}$.

Notice that there is almost no work involved in parametrizing the problem when the system is open loop stable. In LSS problems such a situation can often be arranged via low bandwidth pre-compensation of the rigid body behavior. However, this fact does nothing to simplify the H_∞ optimization process.

The standard H_∞ optimization problem can be stated as finding a matrix Q in RH_∞ such that

$$\| T_1 - T_2QT_3 \|_\infty$$

is minimized. This is sometimes called the model matching problem. Once such a Q is found, Equation 11 can be used to find a controller which simultaneously achieves the optimization and stabilizes the system.

A few comments are in order regarding why the H_∞ problem is difficult. The reasons are most easily seen by considering only those situations in which G is open loop stable and examining situations for which the equation

$$T_1 - T_2QT_3 = 0$$

can be solved. First, T_2 and T_3 must have left and right inverses, respectively, which implies that the system must have at least as many regulated quantities as controller outputs and at least as many external inputs as measurements. Neither of these conditions is overly restrictive as they are both equivalent to requiring that the system not have too many sensors or too many effectors. Then

$$Q = T_2^{-1}T_1T_3^{-1}.$$

However, to be able to find a stabilizing controller, Q must be in RH_∞, which implies that T_2^{-1} and T_3^{-1} must be in RH_∞. This is equivalent to saying that T_2 and T_3 must have no transmission zeros in the right half of the complex plane and that they be non-strictly proper. In other words, T_2 and T_3 must be minimum phase and the resulting inverses must be realizable. The case in which they are non-minimum phase is the usual problem considered in H_∞ optimization theory. The case in which they are strictly proper can be dealt with via weighting matrices and therefore yields sub-optimal designs.

Solution of the Model Matching Problem

The theory of the H_∞ optimization step is quite advanced and involved in the multivariable case. Probably the best way to approach a presentation of the solution to derive it for the much simpler SISO case and then simply present the generalizations required for the solution in the MIMO case. A fairly simple SISO derivation using well known system theoretic concepts is contained in [9]. Another SISO derivation is contained in Francis [10], which also gives an authoritative proof (but not really a derivation) for the MIMO case. Various other approaches are documented in Cheng and Pearson [6], Safanov and Verma [12], Kwakernaak [13], and Grimble [14]. The approach here is to present the required steps in the solution in the form of an algorithm taken from Francis [10] with comments relating to the differences between the SISO and MIMO cases.

As mentioned previously, the major problem involved with the model matching problem is that the inverses of T_2 and T_3 may not exist as matrices in RH_∞. The consequence is that the optimality criterion is non-zero. The theory of H_∞ optimization tells how to find a Q in RH_∞ which achieves the minimum value. Although the theory is general enough to handle non-square, singular T_2 and T_3, in the interest of clarity the following explanation and algorithm deals only with the square and nonsingular case. The completely general theory can be found in Francis [10] and a more compact explanation without proofs can be found in Irwin [9].

We begin with the problem of minimizing

$$T_1 - T_2QT_2T_3$$

where T_1, T_2, and T_3 are in RH_∞. This problem is converted to another equivalent problem by using inner-outer factorizations of T_2 and T_3. An inner-outer factorization of a matrix G is two matrices G_i and G_o such that

$$G = G_iG_o \qquad (16)$$

where

$$G_i^{\sim}G_i = I \qquad (17)$$

and G_o is right invertible with the right inverse in RH_∞. In the case G is square, invertible, and is in RH_∞, G_o is nonsingular in RH_∞. A co-inner-outer factorization of G is given by

$$G = G_{co}G_{ci} \qquad (18)$$

where

G_{co}^T is right invertible in RH_∞

and

$$(G_{ci}^T)^{\sim}(G_{ci}^T) = I. \qquad (19)$$

The first step in the H_∞ optimization solution is to find an inner-outer factorization of T_2 and a co-inner-outer factorization of T_3:

$$T_2 \square T_{2i}T_{2o} \tag{20}$$

$$T_3 = T_{3co}T_{3ci}. \tag{21}$$

A description of a reliable state-space approach to calculating inner-outer factorizations can be found in Francis [10]. The closed loop transfer function matrix can then be written as

$$T_1 - T_{2i}T_{2o}QT_{3co}T_{3ci}. \tag{22}$$

Since inner and co-inner factors have unity norm by definition it is true that

$$\|T_1 - T_{2i}T_{2o}QT_{3co}T_{3ci}\|_\infty = \|T_{2i}^{-1}T_1T_{3ci}^{-1} - T_{2o}QT_{3co}\|_\infty. \tag{23}$$

Letting

$$R = T_{2i}^{-1}T_1T_{3ci}^{-1} \tag{24}$$

and

$$X = T_{2o}QT_{3co} \tag{25}$$

then

$$\| T_1 - T_2QT_3 \|_\infty = \| R - X \|_\infty. \tag{26}$$

Now if X in RH_∞ can be found to minimize the norm, then Q in RH_∞ can be found by solving Equation 25, since T_{2o} and T_{3co} are invertible in RH_∞. Still more work is involved, however, before the application of the fundamental theory for finding such an X.

The next step is to find a minimal antistable/stable decomposition such that

$$R = R_1 + R_2 \qquad (27)$$

where R_1 is completely unstable ($R_1^{\sim}$ is in RH_∞) and strictly proper and R_2 is in H_∞. The stable part of R is trivially included in X by letting

$$X = X_1 + X_2 \qquad (28)$$

where

$$X_2 = R_2. \qquad (29)$$

Then we have

$$\| T_1 - T_2QT_3 \|_\infty = \| R_1 - X_1 \|_\infty \qquad (30)$$

where R_1 is antistable and X_1 is stable. This means that the H_∞ problem reduces to the problem of finding the closest stable matrix to a fixed antistable matrix. Finding a fixed antistable matrix R_1 such that Equation 30 holds is the main added difficulty incurred for the case of singular T_2 and T_3. However the iterative procedure required obscures the central issues and is not included, since no additional algorithms or theories are required. The calculation of the antistable/stable decomposition of R is potentially the least reliable step in the calculation of an H_∞ optimal controller, since the corresponding state space representation of R must be converted to nearly block Jordan form; this process is infamous for its numerical instabilities.

The next step in the process of calculating Q is to find α, the minimum value of the optimality criterion:

$$\alpha = \min\| T_1 - T_2QT_3 \|_\infty = \min\| R_1 - X_1 \|_\infty. \qquad (31)$$

The value of α can be calculated by

$$\alpha^2 = \max \{ \text{ eigenvalues of } W_cW_o \} \qquad (32)$$

where W_c and W_o are the controllability and observability grammians of any state-space realization of R_1 (A,B,C) and can be calculated by solving the Liapunov equations

$$AW_c + W_cA^T = BB^T \tag{33}$$

$$A^TW_o + W_oA = C^TC. \tag{34}$$

From Equation 31 we also have

$$\min\| R_1/(\alpha+\beta) - X_1/(\alpha+\beta) \|_\infty = \alpha/(\alpha+\beta) \tag{35}$$

where $\beta > 0$.

Let

$$R' = R_1/(\alpha+\beta) \tag{36}$$

and

$$X' = X_1/(\alpha+\beta). \tag{37}$$

By Equation 35

$$\alpha/(\alpha+\beta) \le \min\| R' - X' \|_\infty < 1. \tag{38}$$

Equation 38 seems rather unnecessary, but it turns out that it is easier to find X′ such that

$$\| R' - X' \|_\infty \le 1 \tag{39}$$

than it is to find X_1 in the original problem equal to a specified value. In any case when we find X′ which satisfies Equation 39 we have

$$\alpha/(\alpha+\beta) \le \min\| R' - X' \|_\infty \le \| R' - X' \|_\infty \tag{40}$$

and since β is arbitrary we can find X′ as close to the optimal as is desired.

The next, and final, step in the optimization procedure is to find X′ such that with R′ given in Equation 36 , the relationship of Equation 39 holds. To begin, let (A, B, C) be a realization of R′ and set

$$N = (I - W_o W_c)^{-1}. \tag{41}$$

$$L_1 = (A,\ N^T,\ C,\ 0) \tag{42}$$

$$L_2 = (-A^T,\ NW_oB,\ B^T,\ I). \tag{43}$$

Then an X′ (not the only one) which satisfies Equation 39 is given by

$$X' = R' - L_1 L_2^{-1} \tag{44}$$

and the H_∞ optimization problem is solved, at least to an arbitrarily tight tolerance. The preceding development can be summarized as an algorithm:

Step 1

Cast the problem at hand in the form of the standard problem of Equations 1 - 3.

Step 2

Find a doubly co-prime factorization of G_{22} which satisfies Equations 9 and 10.

Step 3

Parameterize the closed loop transfer function matrix by using the results of Step 2 to find T_1, T_2, and T_3 of Equations 13 - 15.

Step 4

Find inner-outer and co-inner-outer factorizations of T_2 and T_3.

Step 5

Use Equations 24 and 25 to find R and X.

Step 6

Find an antistable/stable decomposition of R, obtaining R_1, R_2, and X_2.

Step 7

Calculate α, the optimal norm, using Equations 32 - 34.

Step 8

Choose a $\beta > 0$ to achieve the desired tolerance in Equation 40.

Step 9

Calculate R' from Equation 36.

Step 10

Calculate X' from Equations 41 - 44.

Step 11

Calculate X_1 from Equation 37 and X from Equation 28.

Step 12

Calculate Q by solving Equation 25.

Step 13

Calculate the H_∞ optimal controller from Equation 11.

Although the algorithm is stated in rather concise form, most of the separate steps involve a series of calculations themselves. The software requirements are discussed in the next section.

SOFTWARE SYSTEM AND ALGORITHM SUMMARY

Due to the complexity of the algorithms for accomplishing H_∞ control designs, a preliminary software system is under development. The purpose of the preliminary software is to lend flexibility and reliability to the software development process. The philosophy on which the software system is based is to develop a relatively small number of independent commands accessible via the particular operating system in use. Data is shared between these different low-level commands via a "state space system" representation which is sufficiently flexible to allow for all types of numerical data storage and which relies on the sequential file capabilities of the particular computer. An added benefit of this approach is that internal memory requirements are those of each particular module, rather than the memory required for the total design process. This approach is currently implemented on an MSDOS based 32-bit COMPAQ computer with Microsoft Fortran. However, the approach is fully consistent with any operating system and Fortran compiler capable of providing an interface between the operating system command line interpreter and Fortran program modules. This is certainly the case for MSDOS and UNIX based operating systems and can at the very least be modified for use on DEC VMS based machines. Of course, once design methods for H_∞ control become well established, the preliminary system can be converted to a more conventional single-module form, with the added advantage of the ability to directly evaluate memory requirements based on the experience of the preliminary system.

Data Representation

The only representation for shared data between modules is that of state space system representations stored in sequential files on system disk space. This representation allows for the storage of general linear continuous and discrete-time systems, two-dimensional array data, one-dimensional array data, and scalar data. For example, the system

$$x = Ax + Bu$$

$$y = Cx + Du$$

would be stored in a file named "system1" in the following form:

1st Record	# inputs(m), # outputs(p), # states(n)
Next n Records	rows of A
Next n Records	rows of B
Next p Records	rows of C
Next p Records	rows of D.

An i x j matrix Z would be stored in file "Z" as:

1st Record	j, i, 0
Next i Records	rows of Z

so that a matrix is stored in the form of a system with j inputs, i outputs, and 0 states. A scalar quantity is stored as a 0 state system with 1 input and 1 output.

Low Level Modules

Low level modules are kept as simple as possible to allow for simplified debugging and to minimize internal storage. Typically, these modules do not destroy the contents of the input files. A list of the low level modules which exist at this time and their functions follows. Linear algebraic routines are taken, when possible, from LINPACK and EISPACK.

SADD A B C

Takes the systems contained in A and B, adds them as if their respective transfer function matrices were added, and places the result in the file C.

SMULT A B C

Takes the systems contained in A and B, multiplies them as if their respective transfer function matrices were multiplied, and places the result in the file C.

SINV A C

Takes the system A, inverts it as if its transfer function matrix were inverted, and places the result in file C. Limited to invertible systems (non-strictly proper).

SCOM A B C D S

Takes individual matrix elements of a system representation A, B, C, D in separate files and places them in a single system file format in S.

SSEP S A B C D

Takes the system representation S and places the matrices A, B, C, D in separate files.

SSTRAN S1 T S2

Applies the similarity transform T to S1 and places the result in S2.

SHTRANS S1 S2

Performs the state space operations on S1 which are equivalent to forming $G^T(-s)$ and places the result in S2.

ZCRIC A B Q R Z

Forms the "Hamiltonian" or "Z-matrix" required for Laub's Riccati equation solution [15]. A, B, Q, R are the equation coefficient matrices and Z is the resulting Hamiltonian matrix. (continuous form)

ZCLIA A Q Z

Same function as ZCRIC except for the solution of a Liapunov equation. (continuous form)

CSCHURS Z W1 W2

Forms an orthogonal similarity transformation which is used in Laub's method for solving Riccati and Liapunov equations. W1 and W2 are matrices formed from blocks of the transformation.

RSCHUR A T C

Transforms the matrix A to real schur form (block upper diagonal) via an orthogonal transformation T.

STRANS S1 S2

Takes the transpose of (the transfer function matrix of) S1 and plces the result in S2

Low level modules still to be developed are:

ASDECOMP R R1 R2

Given a system R finds R1 antistable and R2 stable such that R = R1 + R2. (this requires what is almost a transformation to Jordan canonical form; an original algorithm has been developed to avoid many of the numerical instabilities, based on a method by Golub and Van Loan [16] for transforming a matrix to a special block diagonal form.)

MINIMAL S1 S2

Finds a minimal representation of the system S1 and places the result in file S2. (current plans are to use the "staircase" algorithm of Mayne [17]; if this algorithm proves unreliable a method due to Davison et. al [18] will be used. Davison's method is based on random choices of feedback to identify unobservable and/or uncontrollable modes.)

MSQRT A C

Finds a square root of a positive definite matrix A and places the result in file C

INFNORM S1 SCALAR

Finds the infinity norm of the system S1 and places the result in the file SCALAR. This involves first calculating the frequency response matrix of S1, and then calculating the maximum singular value at each frequency. This is the most numerically intensive step in the software system. An extremely efficient algorithm by Laub [20] will be used to calculate the frequency response matrix.

High Level Modules

High level modules will generally consist of batch-type files which contain a group of low level module commands. The full H_∞ design system will in turn be comprised of a combination of the low level and high level modules. Most of the high level modules are currently under development. A partial list of the required modules and their functions follows.

RICCATI AND LIAPUNOV EQUATION SOLUTIONS (existing)

Name: CRICSOL, CLIASOL

Other modules required:

ZCRIC, ZCLIA, CSCHURS, SMUL
SINV

DOUBLY CO-PRIME FACTORIZATIONS

Name: DCOFAC

Other modules required:

SADD, SMULT, SSEP, SCOM
STRANS, CRICSOL

FORM T_1, T_2, T_3 (TRANSFER FUNCTION PARAMETRIZATION)

Name: TFORM

Other modules required:

SMULT, SADD, DCOFAC

SPECTRAL FACTORIZATION

Name: SPECFAC

Other modules required:

ASDECOMP, SADD, SMULT, SINV
STRANS, CRICSOL, MSQRT, SSEP
SCOM, SHTRANS, MINIMAL

CO-SPECTRAL FACTORIZATION

Name: COSPECFAC

Other modules required:

STRANS, SPECFAC

INNER-OUTER FACTORIZATION

Name: IOFAC

Other modules required:

SHTRANS, SPECFAC, SMULT, SINV

CO-INNER-OUTER FACTORIZATION

Name: COIOFAC

Other modules required:

STRANS, IOFAC

COMPUTING THE DISTANCE FROM A GIVEN ANTISTABLE SYSTEM TO THE NEAREST STABLE SYSTEM

Name: HNORM

Other modules required:

SSEP, CLIASOL, SCOM, RSCHUR
SMULT

The reader at this point can appreciate the complexity of the software required to undertake an H_∞ design. The author believes that the module based approach is necessary to minimize development time and to maintain the required computer resources at a reasonable level.

SUGGESTIONS FOR FURTHER INVESTIGATION

It has already been noted that a complete evaluation of the applicability of H_∞ control to LSS problems must involve design studies, simulation studies, and ultimately tests on the MSFC LSS/GTF. To these ends, the obvious next step in the evaluation is to complete the development of a workable software system.

Additionally, there are several other areas of investigation that deserve attention.

Although the strategy of casting disturbance attenuation, tracking and robust stabilization problems via the standard problem formulation is very powerful when combined with H_∞ optimization, the actual design difficulties in LSS control involve the intelligent trade-off of performance and robustness. There are at least two approaches that immediately present themselves for further investigation. The first is to attempt to cast both of these goals simultaneously into the standard problem with a scalar "emphasis ratio" as the trade-off design parameter. The second approach is to borrow from the classical concept of minor loop design. In this approach, an inner loop design would first be accomplished with robust stabilization as the goal. An outer loop would then be designed with performance in mind. The idea is that the outer loop could be designed with a high degree of confidence in the model used, since the inner, or minor, loop had been designed to minimize the effects of plant variations.

Another issue of critical interest is that of controller complexity. H_∞ optimization techniques typically result in high order controllers. Preliminary results by Doyle et al. [20] indicate that the controller order can be maintained at the order of the open loop plant design model. However, this is probably unrealistic for the very high order models typical of LSS. A possible method for obtaining reduced order controllers is by utilizing balanced model reduction [21] on controllers obtained using the full design model. Glover [22] has shown that the degree of order reduction is closely related to closed loop stability and has derived an infinity norm bound that quantifies the allowable reduction. Safanov et al. [23] have recently presented reliable algorithms for calculating balanced reduced order controllers. It remains to be seen whether significant order reductions can be achieved in practical LSS problems.

Still another issue is whether the additional design complexity of H_∞ optimization gains enough in terms of the design goals to justify its use. The possibility of using the controller/transfer function parametrization to obtain non-optimal, but satisfactory, controllers has not been addressed.

Since the development of H_∞ optimization techniques is still in a state of evolution, it is difficult to make even a preliminary statement at this time regarding the eventual applications. There is no doubt, however, that the theory and computational techniques are sufficiently mature to begin an evaluation process.

REFERENCES

[1] G. Zames, "Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses," IEEE Trans. Auto. Cont., vol. AC-26, pp. 301-320.

[2] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, Wiley, New York, 1972.

[3] M.G. Safonov and M. Athans, "Gain and Phase Margin of Multiloop LQG Regulators," IEEE Trans. Auto. Cont., vol. AC-22, pp. 173-178.

[4] J.C. Doyle, "Guaranteed Margins for LQG Regulators," IEEE Trans. Auto. Cont., April 1978, pp. 756-757.

[5] B.A. Francis, J.W. Helton, and G. Zames, "H_∞ Optimal Feedback Controllers for Linear Multivariable Systems," IEEE Trans. Auto. Cont., vol. AC-29, pp. 888-900.

[6] L. Cheng and J.B. Pearson, "Frequency Domain Synthesis of Multivariable Linear Regulators," IEEE Trans. Auto. Cont., vol. AC-23, pp. 3-15.

[7] C.A. Desoer, R.W. Liu, J. Murray, and R. Saeks, " Feedback System Design: the Fractional Representation Approach," IEEE Trans. Auto. Cont., vol. AC-25, pp. 399-412.

[8] M. Vidyasagar, Control System Synthesis: A Factorization Approach, MIT Press, Cambridge, 1985.

[9] R.D. Irwin, Lecture Notes on H_∞ Optimal Control Theory, given at NASA Marshall Space Flight Center, 1988.

[10] B.A. Francis, "A Course in H_∞ Control Theory," Lecture Notes in Control and Information Sciences vol. 88, Springer-Verlag, 1987.

[11] C.N. Nett, C.A. Jacobson, and M.J. Balas, " A Connection Between State Space and Doubly Coprime Fractional Representations," IEEE Trans. Auto. Cont., vol. AC-29, pp. 831-832.

[12] M.G. Safonov and M.S. Verma, "L_∞ Sensitivity Optimization and Hankel Approximation," IEEE Trans. Auto. Cont., vol. AC-30, pp. 279-280.

[13] H. Kwakernaak, "A Polynomial Approach to Minimax Frequency Domain Optimization of Multivariable Systems," Int. J. Control, vol. 44, pp. 117-156.

[14] M.J. Grimble, "Optimal H_∞ Multivariable Robust Controllers and the Relationship to LQG Design Problems," Int. J. Control, vol. 48, pp. 33-58.

[15] A.J. Laub, "A Schur Method for Solving Algebraic Riccati Equation," IEEE Trans. Auto. Cont., vol. AC-24, pp. 913-921.

[16] G.H. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, 1983.

[17] D.Q. Mayne, "An Elementary Derivation of Rosenbrock´s Minimal Realization Algorithm," IEEE Trans. Auto. Cont., vol. AC-18, pp. 306-307.

[18] E.J. Davison, W. Gesing, and S.H. Wang, "An Algorithm for Obtaining the Minimal Realization of a Linear Time Invariant System and Determining if a System is Stabilizable-Detectable," IEEE Trans. Auto. Cont., vol. AC-23, pp. 1048-1054.

[19] A.J. Laub, "Efficient Multivariable Frequency Response Computations," IEEE Trans. Auto. Cont., vol. AC-26, pp. 407-408.

[20] J.C. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-Space Solutions to Standard H_2 and H_∞ Control Problems," Proc. Amer. Cont. Conf., 1988.

[21] B.C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," IEEE Trans. Auto. Cont., vol. AC-26, pp. 17-31.

[22] K. Glover, "All Optimal Hankel Norm Approximations of Linear Multivariable Systems, and Their L_∞ Error Bounds," Int. J. Control, vol. 39, pp. 1145-1193.

[23] M.G. Safonov and R.Y. Chiang, "A Schur Method for Balanced Model Reduction," Proc. Amer. Cont. Conf., 1988.