A Comparison of Descent and Continuation Algorithms for H_2 Optimal, Reduced-Order Control Design

By

Emmanuel G. Collins, Jr. and Debashis Sadhukhan
Department of Mechanical Engineering
Florida A&M University – Florida State University
2525 Pottsdamer St.
Tallahassee, FL 32310-6046
(904) 487-6373
FAX: (904) 487-6337
Ecollins@eng.fsu.edu

E. G. Collins, Jr., and D. Sadhukhan, "A Comparison of Descent and Continuation Algorithms for H_2 Optimal Reduced-Order Control Design," *International Journal of Control*, Vol. 69, No. 5, pp. 647-662, 1998.

A Comparison of Descent and Continuation Algorithms for H_2 Optimal, Reduced-Order Control Design

by

Emmanuel G. Collins, Jr. and Debashis Sadhukhan
Department of Mechanical Engineering
Florida A&M University - Florida State University
2525 Pottsdamer St.
Tallahassee, FL 32310-6046
(904) 487-6373
FAX: (904) 487-6337
ecollins@eng.fsu.edu

Abstract

Reduced-order control is important in control engineering practice due to inevitable limitations on the throughput of the control processors. This paper considers the direct design of H_2 optimal, reduced-order controllers via parameter optimization approaches. Two classes of numerical methods are considered: 1) descent methods, and 2) continuation (or homotopy) methods. In particular, four standard descent methods are compared with a recently developed continuation algorithm by using three examples appearing in the literature. The continuation algorithm is seen to be much more reliable than the descent methods.

1. Introduction

One of the deficiencies of modern control laws, developed by simply solving a pair of decoupled Riccati equations, in particular, linear-quadratic-guassian (LQG) control and full-order H_∞ control, is that the resultant control laws are always of the order of the design plant. These techniques, though relatively easy to implement computationally, do not allow the designer to constrain the architecture (e.g. order and degree of centralization) of the controller. Such constraints are often necessary in engineering practice due to throughput limitations of the control processors. Reduced-order control is therefore of paramount importance in practical control design. This paper focuses on the design of H_2 optimal, reduced-order controllers.

Two main approaches have been developed to solve the H_2 optimal, reduced-order design problem. The first approach attempts to develop approximations to the optimal reduced-order controller by reducing the dimension of an LQG controller(Yousuff and Skelton 1984a, Yousuff and Skelton 1984b, Anderson and Liu 1989, Villemagne and Skelton 1988, Liu *et al. 1990*). These methods are attractive because they require relatively little computation and should be used if possible. Unfortunately, they tend to yield controllers that either destabilize the system or have poor performance as the requested controller dimension is decreased or the requested control authority level is increased. Hence, if used in isolation, these methods do not yield a reliable methodology for reduced-order design. In addition, these methods do not extend to the design of decentralized controllers. However, it should be mentioned that, in regards to reduced-order control design, the indirect approaches at worst are valuable in providing good initial conditions for the direct approaches described below.

In contrast to controller reduction, direct approaches attempt to directly synthesize an optimal, reduced-order (or decentralized) controller by a numerical optimization scheme. There are two main classes of parameter optimization approaches to direct control design. The first class relies on the use of descent methods (Kramer and Calise 1987, Kuhn and Schmidt 1987, Kwakernaak and Sivan 1972, Ly *et al. 1985*, Mukhopadhyay 1982, Mukhopadhyay 1987, Voth and Ly 1991). Algorithms in this class reduce the H_2 cost at each iteration. The second class relies on the use of continuation methods (Collins *et al.* 1994, Mercadal 1991). In contrast to the descent methods, the H_2 cost is not necessarily reduced at each iteration. It should be mentioned that continuation algorithms (Collins *et al.* 1996b) have also been developed to solve the "optimal projection equations," a set of four coupled Lyapunov and Riccati equations that characterize the H_2 optimal, reduced-order compensator. However, this approach will not be considered here.

From a practical design perspective it is important to determine which class of methods tends to be more numerically robust. As with the vast majority of numerical methods for nonconvex optimization problems, answers to these questions are extremely difficult to prove analytically. Instead, we must rely on numerical experimentation to observe trends. Hence, in this paper the behavior of some standard descent methods (i.e., steepest descent, conjugate gradient, BFGS Quasi-Newton, and Newton's method) (Fletcher 1987) are compared to the corresponding behavior of the continuation algorithm of (Collins *et al.* 1994) by considering design for three reduced-order control design problems appearing in the literature. The results clearly indicate that the continuation algorithm tends to be more numerically robust and is most efficient when the controller is constrained to a tridiagonal form.

The paper is organized as follows. Section 2 formulates the H_2 optimal, reduced-order dynamic compensation problem as a constrained parameter optimization problem and discusses various basis options for the controller in order to reduce the size of the controller parameter vector. Section 3 briefly describes the algorithms for the descent and continuation methods for the H_2 optimal, reduced-order control problem. The descriptions emphasize how the constraint that the controller be stabilizing is taken into account in the algorithms. Section 4 uses the three examples mentioned above to present a comparison of both the numerical robustness and speed of convergence of the descent and continuation methods. Finally, Section 5 presents conclusions.

Notation

$\mathcal{R}^n$	$n \times 1$ real vector
z^T	transpose of z
$\mathrm{tr}(M)$	trace of the sqare matrix M
E	expectation operator
$\mathrm{dom}(g)$	domain of the function $g(\cdot)$

2. H_2 Optimal, Reduced-Order Dynamic Compensation

2.1. Problem Formulation

Consider the system

$$\dot{x}(t) = Ax(t) + Bu(t) + D_1 w(t) \tag{2.1}$$

$$y(t) = Cx(t) + Du(t) + D_2 w(t) \tag{2.2}$$

$$z(t) = E_1 x(t) + E_2 u(t) \tag{2.3}$$

where $x \in \mathcal{R}^{n_x}$, $u \in \mathcal{R}^{n_u}$, $y \in \mathcal{R}^{n_y}$, $z \in \mathcal{R}^{n_z}$, $w \in \mathcal{R}^{n_w}$ is white noise with unit intensity, D_2 has full row rank, and E_2 has full column rank. We desire to design a n_c^{th} order dynamic com pensator,

$$\dot{x}_c(t) = A_c x_c(t) + B_c y(t) \tag{2.4}$$

$$u(t) = -C_c x_c(t) \tag{2.5}$$

where $n_c \leq n$, which minimizes the steady state performance criterion

$$J(A_c, B_c, C_c) \triangleq \lim_{t \to \infty} E[z^T(t) z(t)]. \tag{2.6}$$

In the frequency domain, the cost in (2.6) may be interpreted as

$$J(A_c, B_c, C_c) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{tr}[G_{zw}(j\omega) G_{zw}^*(j\omega)] d\omega \tag{2.7}$$

where $G_{zw}(s)$ is the transfer function from w to z and the right hand side is the square of the H_2 norm of $G_{zw}(s)$.

The state-space evolution of the closed-loop system corresponding to (2.1)-(2.5) is described by

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{D}w(t) \tag{2.8}$$

where

$$\tilde{x}(t) \triangleq \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}, \quad \tilde{A} \triangleq \begin{bmatrix} A & -BC_c \\ B_cC & A_c - B_cDC_c \end{bmatrix}, \quad \tilde{D} \triangleq \begin{bmatrix} D_1 \\ B_cD_2 \end{bmatrix}. \tag{2.9}$$

Now, the cost (2.6) can be expressed as

$$J(A_c, B_c, C_c) = \lim_{t\to\infty} E[\tilde{x}^T(t)\tilde{R}\tilde{x}(t)] \tag{2.10}$$

where

$$\tilde{R} \triangleq \begin{bmatrix} R_1 & R_{12}C_c \\ C_c^T R_{12}^T & C_c^T R_2 C_c \end{bmatrix}, \quad R_1 \triangleq {E_1}^T E_1, \quad R_{12} \triangleq 2{E_1}^T E_2, \quad R_2 \triangleq {E_2}^T E_2. \tag{2.11}$$

(Note that since E_2 has full column rank, $R_2 > 0$.)

To guarantee that the cost J is finite and independent of initial conditions we restrict our attention to the set of stabilizing compensators, $\mathcal{S}_c \triangleq \{(A_c, B_c, C_c) : \tilde{A} \text{ is asymptotically stable}\}$. Assume $(A_c, B_c, C_c) \in \mathcal{S}_c$ and define $\tilde{Q} \in \mathcal{R}^{(n_x+n_c)\times(n_x+n_c)}$ to be the closed-loop steady-state covariance, i.e.,

$$0 = \tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \tilde{V} \tag{2.12}$$

where

$$\tilde{V} \triangleq \begin{bmatrix} V_1 & V_{12}B_c^T \\ B_cV_{12}^T & B_cV_2B_c^T \end{bmatrix}, \quad V_1 \triangleq {D_1}^T D_1, \quad V_{12} \triangleq 2{D_1}^T D_2, \quad V_2 \triangleq {D_2}^T D_2. \tag{2.13}$$

(Note that since D_2 has full row rank, $V_2 > 0$.) The cost function J can now be expressed as

$$J(A_c, B_c, C_c, \tilde{Q}) = \mathrm{tr}\tilde{Q}\tilde{R}. \tag{2.14}$$

The objective is to minimize the cost function J subject to the constraint (2.12).

The Lagrangian $\mathcal{L}$ is defined by

$$\mathcal{L}(A_c, B_c, C_c, \tilde{Q}, \tilde{P}) \triangleq \mathrm{tr}\tilde{Q}\tilde{R} + \mathrm{tr}[\tilde{P}(\tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \tilde{V})] \tag{2.15}$$

where $\tilde{P}$ is the Lagrange multiplier matrix. The compensator (A_c, B_c, C_c) is optimal if it satisfies the stationary conditions

$$\frac{\partial\mathcal{L}}{\partial A_c} = 0, \quad \frac{\partial\mathcal{L}}{\partial B_c} = 0, \quad \frac{\partial\mathcal{L}}{\partial C_c} = 0, \tag{2.16}$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{Q}} = \tilde{A}\tilde{P} + \tilde{P}\tilde{A}^T + \tilde{R} = 0, \tag{2.17}$$

and

$$\frac{\partial \mathcal{L}}{\partial \tilde{P}} = \tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \tilde{V} = 0. \tag{2.18}$$

Both the descent and continuation algorithms aim at finding $(A_c, B_c, C_c) \in \mathcal{S}_c$ that satisfy the above conditions.

Subsequently, we will represent the controller by a parameter vector θ, for example,

$$\theta = \begin{bmatrix} \text{vec}(A_c) \\ \text{vec}(B_c) \\ \text{vec}(C_c) \end{bmatrix}. \tag{2.19}$$

Let the mapping from a state space representation of a controller (A_c, B_c, C_c) to the parameter vector θ be given by $g(\cdot)$, such that

$$\theta = g(A_c, B_c, C_c) \tag{2.20}$$

and define

$$\Theta = \{\theta = g(A_c, B_c, C_c) : (A_c, B_c, C_c) \in S_c \cup \text{dom}(g)\} \tag{2.21}$$

Now, assuming $\theta \in \Theta$, the H_2 cost functional and the corresponding Lagrangian can be expressed respectively as $J(\theta, \tilde{Q})$ and $\mathcal{L}(\theta, \tilde{Q}, \tilde{P})$. The problem is therefore to find $\theta \in \Theta$ such that

$$0 = \frac{\partial \mathcal{L}}{\partial \theta}(\theta, \tilde{Q}, \tilde{P}). \tag{2.22}$$

subject to (2.17) and (2.18).

2.2. Reduction of the Dimension of the Controller Parameter Vector (θ)

It is desired that the parameter vector θ be as small as possible. Hence, we desire to represent the controller matrix with the fewest parameters possible. The minimal number of parameters $p_{\min}$ with which a compensator can be represented is given by (Denery *et al.* 1971, Martin and Bryson 1980)

$$p_{\min} = n_c(m + l). \tag{2.23}$$

One canonical form which allows representation of a controller with a minimal number of parameters is the modal form described in (Martin and Bryson 1980). This form will be called here the Second-Order Polynomial (SP) form. For this parameterization a triple (A_c, B_c, C_c) of order n_c has the following structure.

$$A_c = \text{block-diag}\{A_{c,1}, A_{c,2}, ..., A_{c,n_c}\} \tag{2.24}$$

where $A_{c,i}$ is 2×2, $i = \{1, 2, ..., n_c\}$ and each $A_{c,i}$ (with the exception of A_{c,n_c} if the row dimension of A_c is odd) has the form

$$A_{c,i} = \begin{bmatrix} 0 & 1 \\ a_{c,i}^{(1)} & a_{c,i}^{(2)} \end{bmatrix}, \tag{2.25}$$

to allow for either a complex conjugate set of poles or two real poles. B_c is completely full and

$$C_c = [C_{c,1}, C_{c,2}, , ..., C_{c,r}], \tag{2.26}$$

where $C_{c,i}$ has the form

$$C_{c,i} = \begin{bmatrix} 1 & 0 \\ * & * \\ \vdots & \vdots \\ * & * \end{bmatrix}. \tag{2.27}$$

The controller canonical form described in (Kailath 1980) also allows representation of a controller with a minimal number of parameters. For SISO systems in controller canonical form the A_c matrix is a companion matrix. In particular, A_c has the form

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & 0 \\ * & * & * & * & \cdots & * \end{bmatrix}. \tag{2.28}$$

In addition,

$$B_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{2.29}$$

and C_c is completely full. A dual form of the controller canonical form is the observable canonical form (Kailath 1980).

Another minimal parameter basis is the "input normal Riccati form" of (Davis *et al.* 1994). In this basis A_c is completely determined by B_c and C_c.

It is also possible to represent the controller in a basis where the number of free parameters p satisfies

$$p_{\min} < p < p_{\max} \triangleq n_c \, (n_c + m + l). \tag{2.30}$$

One such basis is the tridiagonal basis (Geist 1991, Parlett 1992) in which the controller state matrix is constrained to have nonzero elements only on the diagonal, the super-diagonal, and the sub-diagonal. That is,

$$A_c = \begin{bmatrix} * & * & & & \\ * & * & * & 0 & \\ & * & * & \ddots & \\ & 0 & \ddots & \ddots & \\ & & & * & * \end{bmatrix}, \tag{2.31}$$

and B_c and C_c are completely full. For this form the number of free parameters is given by

$$p = p_{\min} + (3n_c - 2)$$

It is important to recognize that, given a particular basis, there are stabilizing controllers that have no state space realization in that basis, such that if $g(\cdot)$ in (**??**) requires the representation of a controller in a given basis $S_c \cup \mathrm{dom}(g) \subset S_c$. Hence, the set Θ, defined by (**??**) corresponds to a smaller set of controllers than the set S_c. When the controller is restricted to a particular basis in the algorithms to follow, this reduction in the size of the feasible set sometimes leads to numerical ill-conditioning or even algorithm failure.

3. Parameter Optimization Algorithms

This section first gives a general description of the algorithms corresponding to the descent methods. It then briefly describes a continuation algorithm. Particular attention is given to the modification of these algorithms to take into account the con straint $\theta \in \Theta$.

3.1. Descent Methods

Descent methods are designed to search for solutions to the unconstrained optimization problem

$$\min_{\theta} J(\theta). \tag{3.1}$$

The user is required to supply an initial parameter vector θ_0. A descent algorithm then has the following structure.

A Descent Algorithm

1. Let $k = 0$.

2. Determine a search direction d_k.

3. Use a one dimensional line search to find α_k that minimizes $J(\theta_k + \alpha d_k)$ with respect to α.

4. Set $\theta_{k+1} = \theta_k + \alpha_k d_k$

5. If the gradient $\frac{\partial J}{\partial \theta}(\theta_{k+1})$ is sufficiently small, then let the optimal solution $\theta^* = \theta_{k+1}$ and stop, else let $k = k + 1$ and go to Step 2.

Alternative descent methods differ primarily in the way they compute the descent direction d_k. For example, in the steepest descent method d_k corresponds to the negative of the gradient. Conjugate gradient and Quasi-Newton methods compute d_k us ing only cost and gradient information while Newton's method requires computation of the Hessian matrix. Note that for the H_2 optimal, reduced-order control problem it is not difficult to show that if (2.17) and (2.18) are satisfied, then the gradient satisfies

$$\frac{\partial J}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \theta}. \tag{3.2}$$

Hence, the gradient may be computed by constructing and differentiating the Lagrangian.

Recognize that the H_2 optimal, reduced-order control problem is not the unconstrained optimization problem (3.1) but is actually the constrained optimization problem

$$\min_{\theta \in \Theta} J(\theta) \tag{3.3}$$

where Θ is defined by (**??**). One way to take into account the constraint $\theta \in \Theta$ is to modify the line search subalgorithm of Step 3 to ensure that if $\theta_k \in \Theta$, θ_{k+1} is also in Θ. (It is assumed that $\theta_0 \in \Theta$.) An example of such a modification is discussed in (Kuhn and Shmidt 1987). The descent algorithms compared in this paper use the following modification to the line search algorithm of (Fletcher 1987). During the bracketing phase of the line search algorithm a stability check is carried out to make sure that the right end of the bracket results in a stable closed-loop system. If not, then the bracket is halved. This simple modification ensures that all points within the bracket result in a stable closed-loop system.

3.2. Continuation Methods

Continuation techniques can be used to solve the zero finding problem

$$0 = f(\theta), \tag{3.4}$$

where $f : \mathcal{R}^p \to \mathcal{R}^p$. In the context of H_2 optimal, reduced-order control, (3.4) corresponds to (2.20). Continuation techniques work by finding a C^2 function $H : \mathcal{R}^p \times [0, 1) \to \mathcal{R}^p$ that satisfies certain properties, including the following:

1. $H(\theta, 1) = f(\theta)$;

2. $0 = H(\theta, 0)$ has an easily found or known solution θ_0.

They then trace the zero curve described by

$$0 = H(\theta, \lambda), \quad \lambda \in [0, 1). \tag{3.5}$$

This is accomplished by differentiating (3.5) with respect to λ to obtain Davidenko's differential equation

$$0 = H_\lambda(\theta, \lambda) + H_\theta(\theta, \lambda)\theta_\lambda(\lambda) \tag{3.6}$$

where $H_\lambda \triangleq \frac{\partial H}{\partial \lambda}$, $H_\theta \triangleq \frac{\partial H}{\partial \theta}$, and $\theta_\lambda \triangleq \frac{d\theta}{d\lambda}$, which together with $\theta(0) = \theta_0$ defines an initial value problem. Predictor-corrector, numerical integration schemes are then used to solve this initial value problem, that is to follow the curve (3.5) from the solution θ_0 of $0 = H(\theta, 0)$ to a solution θ^* of $0 = H(\theta, 1)$. In particular a continuation algorithm has the following structure.

A Continuation Algorithm

1. Let $\lambda = 0$ and $\theta(\lambda) = \theta_0$.

2. Use (3.6) to compute the tangent vector θ_λ, such that $\theta_\lambda(\lambda) = -H_\theta(\theta, \lambda)^{-1} H_\lambda(\theta, \lambda)$.

3. For some $\Delta\lambda$ such that $\lambda = \lambda + \Delta\lambda \le 1$, use current and past values of H and H_λ to predict $\theta(\lambda + \Delta\lambda)$ by using polynomial curve fitting.

4. Let $\lambda \leftarrow \lambda + \Delta\lambda$ and θ_0 be the prediction of $\theta(\lambda)$.

5. For $k = 0, 1, 2, \ldots$ until convergence, do

$$\theta_{k+1} = \theta_k - H_\theta(\theta_k, \lambda)^{-1}\theta_k.$$

   Then, let $\theta(\lambda) = \theta_{k+1}$.

6. If $\lambda < 1$, go to Step 2, else if $\lambda = 1$, then let the solution $\theta^* = \theta(\lambda)$ and stop.

The initializing controller θ_0 in the algorithm for H_2 optimal, reduced-order control (Collins *et al.* 1994, Collins *et al.* 1996a) is usually found by applying a controller reduction method such as balanced controller reduction (Yousuff and Skelton 1984a) to a low authority LQG controller since this usually yields a nearly optimal, reduced-order controller. The initial weights $(R_1)_0$, $(R_{12})_0$, $(R_2)_0$, $(V_1)_0$, $(V_{12})_0$, $(V_2)_0$ corresponding to the low authority LQG controller are then deformed into the desired weights along the homotopy path. The reader is referred to (Collins *et al.* 1994) for further details.

The algorithm of (Collins *et al.* 1994) also assumes that the prediction $\theta(\lambda + \Delta\lambda) \in \Theta$ such that it corresponds to a controller that stabilizes the closed-loop system. If $\theta(\lambda + \Delta\lambda) \notin \Theta$, then the algorithm reduces the size of $\Delta\lambda$. In particular, $\Delta\lambda \leftarrow \frac{1}{2}\Delta\lambda$. This is a small but necessa ry modification. As subsequently discussed, it is at this point that the algorithm sometimes fails.

4. Numerical Examples

4.1. Description of Problems

The first problem is a noncollocated axial vibration control problem involving an axial beam with four circular disks attached. This problem was introduced in (Cannon and Rosenthal 1984) and also studied in (Collins *et al* 1994). The plant is 8th order while we consider the design of a 4th order controller.

The second problem was introduced in (Ly *et al.* 1985) and involves flight control for a NAVION aircraft. The model is 7th order and we consider the design of a 4th order controller.

The third problem was introduced in (Martin and Bryson 1980) and involves vibration control of a flexible spacecraft. The model is 6th order while we again consider the design of a 4th order controller.

The system matrixes (A, B, C, D) and the weighting matrices $(R_1, R_2, R_{12}, V_1, V_2, V_{12})$ are given in the Appendix. Note that the matrices R_2 and V_2 are multiplied by ρ which is allowed to change from 10 to 1 in order to deform the low authority controller to a higher authority controller using the homotopy algorithm. In the case of the descent algorithms ρ is fixed at 1.

For each example, a low authority optimal LQG controller (corresponding to $\rho = 10$) was first designed. The order of this controller was then reduced using the modified balanced controller reduction technique of (Yousuff and Skelton 1984a). This reduced order sub-optimal controller was then converted into an optimal low authority controller using a few Newton iterations. This controller was used as the starting point for both the continuation and descent optimization methods. These numerical examples are included in Tables 1 through 3 and were run on a 90 MHz, Pentium PC. Table 4, which only presents data for the continuation algorithm, was produced using a 120 MHz Pentium PC.

We also design higher order controllers for all three examples using continuation algorithms with the controller unconstrained and with the controller constrained to the tridiagonal basis, in order to compare the two methods. These numerical examples are included in Table 4 and were run on a 120 MHz Pentium PC.

Method/Basis	Unconstrained		Tridiagonal		SPF		CCF	
	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)
Continuation	19.8	55	14.8	50.8	374	1422	349.8	1987
Newton	13.85	41.1	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls
BFGS	13.45	41.3	3.4	15.3	f-ls	f-ls	f-ls	f-ls
Conjugate Grad.	56.5	167.5	13.7	46.8	f-osc	f-osc	f-ls	f-ls
Steepest Des.	227	727	68.3	207.3	f-osc	f-osc	f-osc	f-osc

Table 1: FOUR DISKS

4.2. Observations

Comparison of the various algorithms are given in Tables 1, 2, and 3. The letter "f" denotes failure of an algorithm. In particular "f-ls" denotes failure due to the step length parameter computed by the line search, becoming extremely small. This occurs because of the modification to the line search algorithm, to make sure that the controller results in a stable closed-loop. During the line search if the resulting controller violates this constraint, as previously described, the step size is reduced by half. This sometimes results in extremely small step sizes for which there is no appreciable decrease in cost or change in the search direction. Hence the algorithm effectively comes to a stop.

Similarly, "f-λ" denotes failure due to the need for extremely small increments in the homotopy parameter λ. This occurs because of the modification to the prediction step in the continuation algorithms to avoid unstable closed-loops. If during any of these steps, an unstable closed loop is obtained, then, as previously described, the increment to the continuation parameter λ, is decreased by half. This sometimes results in the need for extremely small increments in the homotopy parameter λ for which there is no appreciable change in the controller parameters θ.

The indicator "f-osc" denotes failure in a descent algorithm, due to oscillatory behavior of the gradient of the cost function as the algorithm progresses. Several hundreds of iterations are performed without any perceptible change in the cost function. This phenomena was observed primarily in the steepest descent algorithm and is a well known deficiency of this method (Fletcher 1987).

Tables 1 through 3 reveal that constraining the basis of the controller led to increased failure in

Method/Basis	Unconstrained		Tridiagonal		SPF		CCF	
	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)
Continuation	537	210	470	210.2	f-λ	f-λ	f-λ	f-λ
Newton	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls
BFGS	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls
Conjugate Grad.	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls
Steepest Des.	f-osc	f-osc	f-osc	f-osc	f-ls	f-ls	f-ls	f-ls

Table 2: NAVION

Method/Basis	Unconstrained		Tridiagonal		SPF		CCF	
	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)	M Flops	Time (sec)
Continuation	12.6	10.38	21.7	21.1	25.1	23.4	121.8	94.85
Newton	11.7	9.94	5.9	9.88	5.22	7.25	7.57	8.9
BFGS	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls	f-ls
Conjugate Grad.	35.88	25.3	253.4	159.8	74.8	51.19	517.7	302.6
Steepest Des.	200.8	147.2	f-osc	f-osc	5.66	6.92	285	193

Table 3: SPACECRAFT

Example(Controller Order)/Basis	Unconstrained		Tridiagonal	
	M Flops	Time (sec)	M Flops	Time (sec)
Fourdisks (6^{th} order)	1028.5	294.5	412.8	164
Fourdisks (8^{th} order)	3192.7	596	809.9	239.4
Navion (6^{th} order)	2815	600.8	1566.2	457.7
Spacecraft (6^{th} order)	347.9	130.1	196.5	98.4

Table 4: Continuation: Unconstrained vs Tridiagonal

both the continuation and descent algorithms (i.e. by both the "f-λ" and the "f-ls" failure modes). One apparent reason for this is that constraining the controller to a particular basis reduces the feasible subspace, as discussed at the end of Section 2. Algorithm failure due to basis constraints was also observed in (Kuhn and Schmidt 1987).

The numerical conditioning of the algorithms when using the tridiagonal basis was better than when using the second order polynomial form (SPF) and the controller canonical form (CCF) and is apparently due to the fact that the tridiagonal form is a more general representation than SPF and CCF. In fact, SPF is a special case of a tridiagonal form. This phenomena can be seen by a comparison of execution times for the respective bases in Tables 1 through 3. For most cases, the execution times for the SPF and the CCF bases, are several times larger than the those for the tridiagonal basis, even though the SPF and CCF are minimal parameter bases, while the tridiagonal is not, and hence have smaller parameter vectors θ. This is due to the decreased numerical conditioning associated with the use of the SPF and CCF basis.

The conjugate gradient method, as might be expected, usually converged faster than the steepest descent method. The convergence times for the Newton, BFGS and continuation methods were equivalent and usually much less than the convergence times for the conjugate gradient and the steepest descent methods. However all the descent methods including Newton and BFGS methods fail much more often than did the continuation methods. This is especially true when the controller basis is unconstrained or tridiagonal, in which case the continuation method never failed.

In Tables 1 through 3, it is seen that the run times of the continuation algorithm for the unconstrained and tridiagonal cases were very similar. However, as the controller dimension increases, the size of the parameter vector associated with the unconstrained "basis" increases much more rapidly than the parameter vector associated with the tridiagonal basis. Hence, it is expected that the convergence times for the tridiagonal case will increase much less rapidly than the unconstrained case as the controller dimension increases. This is confirmed in Table 4 which was generated using a 120 MHz Pentium PC.

5. Conclusions

This paper has used three examples to compare the behavior of four standard descent algorithms with a recently developed continuation algorithm for H_2 optimal, reduced-order design. The results clearly indicate that the continuation algorithm is much more numerically robust than any of the descent algorithms, especially when the controller basis is unconstrained or tridiagonal. The numerical conditioning of the algorithms always decreased when the controller was constrained to a minimal parameter basis. However numerical conditioning of the algorithm when using the tridiagonal form, a nonminimal parameter basis, was better than when using the second order polynomial form and the controller canonical form which are minimal parameter basis. The numerical conditioning of the continuation algorithms, when using the tridiagonal basis is unchanged or at worst marginally decreased, as compared to the unconstrained case. Hence, when using the tridiagonal basis, the advantage of a smaller parameter vector θ, usually outweighs the disadvantage of reduced numerical conditioning due to a basis constraint. This advantage, as expected, becomes more apparent as the order of the controller is increased.

References

[1] B. D. O. Anderson and Y. Liu, 1989, Controller reduction: concepts and approaches. *IEEE Transactions on Automatic Control*, **34**, 802-812.

[2] R. H. Cannon and D. E. Rosenthal, 1984, Experiments in control of flexible structures with noncolocated sensors and actuators. *Journal of Guidance, Control and Dynamics*, **7**, 546-553.

[3] E. G. Collins, Jr., L. D. Davis and S. Richter, 1994, Design of reduced-order, H_2 optimal control using a homotopy algorithm. *International Journal of Control*, **61**, 849-852.

[4] E. G. Collins, Jr., W. M. Haddad and S. S. Ying, 1996a, Nearly nonminimal Linear-Quadratic-Gaussian compensators for reduced-order control design initialization. *Journal of Guidance, Control and Dynamics*, **19**, 259-261.

[5] E. G. Collins, Jr., S. S. Ying, and W. M. Haddad, 1996b, Reduced-Order Dynamic compensation using the Hyland and Bernstein optimal projection equations. *Journal of Guidance, Control, and Dynamics*, to appear.

[6] L. D. Davis, E. G. Collins, Jr., and A. S. Hodel, 1994, A parameterization of minimal plants. *IEEE Transactions on Automatic Control*, **39**, 849-852.

[7] D. G. Denery, 1971, An identification algorithm that is insensitive to initial parameter estimates. *AIAA Journal*, **9**, 371-377.

[8] C. De Villemagne and R. E. Skelton, 1988, Controller reduction using canonical interactions. *IEEE Transactions on Automatic Control*, **33**, 740-750.

[9] R. Fletcher, 1987, *Practical methods of optimization*: Second Edition, John Wiley and Sons, New York.

[10] G. A. Geist, 1991, Reduction of a general matrix to tridiagonal Form. *SIAM Journal of Matrix Analysis and Applications*, **12**, 362-373.

[11] T. Kailath, 1980, *Linear systems*, Prentice-Hall, New Jersey.

[12] F. S. Kramer and A. J. Calise, 1987, Fixed-order dynamic compensation for multivariable linear systems. *Journal of Guidance and Control*, **11**, 80-85.

[13] U. Kuhn and G. Schmidt, 1987, Fresh look into the design and computation of optimal output feedback controls for linear multivariable systems. *International Journal of Control*, **46**, 75-95, 1987.

[14] H. Kwakernaak and R. Sivan, 1972, *Linear optimal control Systems*, Wiley-Interscience.

[15] Y. Liu, B. D. O. Anderson and U. L. Ly, 1990, Coprime factorization controller reduction with Bezout identity induced frequency weighting. *Automatica*, **26**, 233-249.

[16] U. L. Ly, A. E. Bryson and R. H. Cannon, 1985, Design of low-order compensators using parameter optimization. *Automatica*, **21**, 315-318.

[17] G. D. Martin and A. E. Bryson, Jr., 1980, Attitude control of a flexible spacecraft. *Journal of Guidance, Control and Dynamics*, **3**, 37-41.

[18] M. Mercadal, 1991, Homotopy approach to optimal, linear quadratic, fixed architecture compensation. *Journal of Guidance, Control and Dynamics*, **14**, 1224-1233.

[19] V. Mukhopadhyay, J. R. Newsom, and I. Abel, 1982, Reduced-order optimal feedback control law for flutter suppression. *Journal of Guidance and Control*, **5**, 389-395.

[20] V. Mukhopadhyay, 1987, Stability robustness improvement using constrained optimization techniques. *Journal of Guidance and Control*, **10**, 172-177.

[21] V. Mukhopadhyay, 1989, Digital robust control law synthesis using constrained optimization. *Journal of Guidance and Control*, **12**, 175-181, 1989.

[22] B. N. Parlett, 1992, Reduction to Tridiagonal form and minimal realization. *SIAM Journal of Matrix Analysis and Applications*, **12**, 567-593.

[23] C. Voth and U. L. Ly, 1991, Design of a total energy control autopilot using constrained parameter optimization. *Journal of Guidance, Control and Dynamics*, **14**, 927-935.

[24] A. Yousuff and R. E. Skelton, 1984a, A note on balanced controller reduction. *IEEE Transactions on Automatic Control*, **29**, 254-257.

[25] A. Yousuff and R. E. Skelton, 1984b, Controller reduction by controller cost analysis. *IEEE Transactions on Automatic Control*, **29**, 520-530.

APPENDIX

The state space descriptions of the 3 examples are as follows.

AXIAL VIBRATION PROBLEM

$$A = \begin{bmatrix} -0.0370 & 1.8496 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.8496 & -0.0370 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0282 & 1.4097 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.4097 & -0.0282 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0153 & 0.7648 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.7648 & -0.0153 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B^T = \begin{bmatrix} -0.1497 & 0.5691 & -0.4961 & -1.2918 & -2.1635 & -1.3608 & 0.1045 & 0.9955 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.1526 & 0.0426 & 0.1181 & -0.0434 & 0.0481 & -0.0768 & 0.2511 & -0.0257 \end{bmatrix}, \quad D = 0$$

$$R_1 = 1.0e{-003} \begin{bmatrix} 0.0007 & -0.0031 & -0.0013 & -0.0030 & -0.0039 & -0.0020 & -0.0011 & -0.0152 \\ -0.0031 & 0.0135 & 0.0057 & 0.0132 & 0.0171 & 0.0088 & 0.0048 & 0.0661 \\ -0.0013 & 0.0057 & 0.0024 & 0.0056 & 0.0072 & 0.0037 & 0.0021 & 0.0280 \\ -0.0030 & 0.0132 & 0.0056 & 0.0130 & 0.0167 & 0.0086 & 0.0048 & 0.0649 \\ -0.0039 & 0.0171 & 0.0072 & 0.0167 & 0.0215 & 0.0111 & 0.0061 & 0.0835 \\ -0.0020 & 0.0088 & 0.0037 & 0.0086 & 0.0111 & 0.0057 & 0.0031 & 0.0429 \\ -0.0011 & 0.0048 & 0.0021 & 0.0048 & 0.0061 & 0.0031 & 0.0017 & 0.0238 \\ -0.0152 & 0.0661 & 0.0280 & 0.0649 & 0.0835 & 0.0429 & 0.0238 & 0.3240 \end{bmatrix}$$

$$R_2 = \rho, \quad {R_{12}}^T = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0], \quad V_2 = \rho, \quad {V_{12}}^T = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0].$$

$$V_1 = \begin{bmatrix} 0.0368 & -0.1560 & 0.2049 & 0.4960 & -0.5984 & -0.3537 & -0.0200 & -0.1909 \\ -0.1560 & 0.6615 & -0.8690 & -2.1031 & 2.5374 & 1.4997 & 0.0850 & 0.8097 \\ 0.2049 & -0.8690 & 1.1417 & 2.7630 & -3.3335 & -1.9703 & -0.1117 & -1.0637 \\ 0.4960 & -2.1031 & 2.7630 & 6.6866 & -8.0673 & -4.7682 & -0.2702 & -2.5742 \\ -0.5984 & 2.5374 & -3.3335 & -8.0673 & 9.7332 & 5.7529 & 0.3260 & 3.1058 \\ -0.3537 & 1.4997 & -1.9703 & -4.7682 & 5.7529 & 3.4002 & 0.1927 & 1.8357 \\ -0.0200 & 0.0850 & -0.1117 & -0.2702 & 0.3260 & 0.1927 & 0.0109 & 0.1040 \\ -0.1909 & 0.8097 & -1.0637 & -2.5742 & 3.1058 & 1.8357 & 0.1040 & 0.9910 \end{bmatrix}$$

The NAVION

$$A = \begin{bmatrix} -0.0450 & 0.0360 & 0 & -0.3220 & 0 & 0.0450 & -0.0360 \\ -0.3700 & -2.0200 & 1.7600 & 0 & 0 & 0.3700 & 2.0200 \\ 0.1910 & -3.9600 & -2.9800 & 0 & 0 & -0.1910 & 3.9600 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0 & -1.0000 & 0 & 1.7600 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.4820 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.0570 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & -0.2820 & -11.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad D^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 & -1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0625 & 0 & 0 \\ -1.0000 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R_2 = \rho \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}, \quad {R_{12}}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$V_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 111.9343 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 111.9261 \end{bmatrix}$$

$$V_2 = \rho \begin{bmatrix} 0.1600 & 0 & 0 \\ 0 & 0.1600 & 0 \\ 0 & 0 & 100.0000 \end{bmatrix}, \quad {V_{12}}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

SPACECRAFT

$$A = \begin{bmatrix} -0.0836 & -2.1904 & 0 & 0 & 0 & 0 \\ 2.1904 & 0 & 0 & 0 & 0 & 0 \\ -0.0738 & -1.7519 & -0.0107 & -0.8710 & 0 & 0 \\ 0 & 0 & 0.8710 & 0 & 0 & 0 \\ -0.0738 & -1.7519 & -0.0063 & -0.6487 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -0.0066 \end{bmatrix}, \quad D = 0$$

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4290 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4290 \end{bmatrix}, \quad R_2 = \rho, \quad R_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$$V_1 = 1.0e-006 \begin{bmatrix} 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad V_2 = 2.1000e-008\ \rho, \quad V_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$