Semi-Annual Progress Report
NASA Grant NAG-2-691

Advanced Rotorcraft Control Using Parameter Optimization

Brett VanSteenwyk & Uy-Loi Ly [1]
Department of Aeronautics and Astronautics, FS-10
University of Washington
Seattle, Washington 98195

Publications:

- Brett VanSteenwyk and Uy-Loi Ly, *Reliable Numerical Computation in An Optimal Output-Feedback Design,* Proceedings of the 1991 AIAA Guidance, Navigation and Control Conference, New Orleans, Louisiana. A copy of the proceeding paper is enclosed.
- Brett VanSteenwyk and Uy-Loi Ly, *Design Challenges for the UH-60 Rotorcraft,* In preparation to be submitted to the 1992 AIAA Guidance, Navigation and Control Conference, Hilton Head, South Carolina. A draft of the related research results is enclosed.

[1]This work is supported in part by NASA Ames Research Center under grant contract NAG-2-691.

Presented at the 1991 AIAA GNC Conference

RELIABLE NUMERICAL COMPUTATION IN AN OPTIMAL OUTPUT-FEEDBACK DESIGN

Brett VanSteenwyk and Uy-Loi Ly *
Department of Aeronautics and Astronautics, FS-10
University of Washington, Seattle, WA 98195

Abstract

This paper presents a reliable algorithm for the evaluation of a quadratic performance index and its gradients with respect to the controller design parameters. The algorithm is part of a design algorithm for optimal linear dynamic output-feedback controller that minimizes a finite-time quadratic performance index. The numerical scheme is particularly robust when it is applied to the control-law synthesis for systems with densely packed modes and where there is a high likelihood of encountering degeneracies in the closed-loop eigensystem. This approach through the use of an accurate Padé series approximation does not require the closed-loop system matrix to be diagonalizable. The algorithm has been included in a control design package for optimal robust low-order controllers. Usefulness of the proposed numerical algorithm has been demonstrated using numerous practical design cases where degeneracies occur frequently in the closed-loop system under an arbitrary controller design initialization and during the numerical search.

1. Introduction

Traditional design methods in linear optimal control for continuous-time systems have been extensively treated in recent literature [1]. Development of these control systems are usually based on characterization of the control problem under the setting of optimization of the two-norm of a set of controlled output responses to random disturbance inputs or initial conditions. Additional consideration of design robustness is taken by formulating the problem to include H^∞-norm bound constraints for a class of additive and multiplicative uncertainties applied at the plant inputs and/or outputs. Solutions are obtained for both the state- and output-feedback design problems and involve in the majority of cases solving an appropriate set of algebraic Riccati equations [2,3]. Theoretical studies of these approaches have been

*The work of B. VanSteenwyk and U. Ly is supported in part by NASA Ames Research Center under grant contract NAG-2-691.

the major concern of researchers in the control field and major breakthrough has been made in recent work by Stoorvogel [4,5]. An alternate and seldomly mentioned design option for robust multivariable control of linear time-invariant systems is based on direct numerical optimization of a quadratic performance index with an arbitrarily specified controller structure. We believe that careful formulation of the design problem under nonlinear constrained optimization can be of great value in the synthesis of robust multivariable control systems.

Early work in this area have been published by Levine and Athans [6], Anderson and Moore [7], and extensive review of the subject was performed by Makila [8]. Recently, a new look into parameter optimization to multivariable control synthesis is provided by Ly [9] where he used a quadratic performance based on finite-time horizon. In the latter work, a numerical optimization technique based on [10] was used. At each design iteration it requires a precise evaluation of a finite-time quadratic performance index and its gradients with respect to the design parameters. Analytical expressions have been developed to evaluate these quantities under the key assumption that the closed-loop system being diagonalizable. This assumption is found to be unsatisfactory and is the cause of convergence difficulties in the iterative search when it attempts to invert an ill-conditionned eigenvector matrix for the diagonalization. The work presented in this paper is to resolve this numerical difficulty and thereby extends the results of Ly [9] for cases where the closed-loop systems are degenerate, i.e the closed-loop system has repeated eigenvalues and the corresponding set of system eigenvectors does not span the whole state-space of the closed-loop system.

The paper is organized as follows. Section 2 reviews the problem formulation for a linear optimal control design using direct parameter optimization. Analytical expressions for the evaluation of the quadratic cost function and its gradients with respect to the controller design parameters are also given in section 2. The current approach to evaluate these quantities are briefly reviewed in section 3. An alternate numerical scheme for the exponential matrix and convolution integrals involving exponential matrices is presented in section 4. Approximation methods for the evaluation of these matrix quantities using Padé series are described in details in sections 5, 6, and 7. A design algorithm based on these numerical solution schemes has been incorporated into a computer-aided design package described in [9]. A simple design problem to motivate the need for a numerical algorithm that handles degeneracies in the closed-loop system matrix is given in section 8. Optimal solutions are obtained using the proposed method and the diagonalization method from [9]. The numerical algorithm has also been applied to the synthesis of an active control system for the JPL large space structure developed under the LSCL research program [16]. Results of this application are presented in section 9. Conclusions are given in section 10.

2. Problem Formulation

In this section, we recall the problem formulation described in Ly [9] for the control synthesis of a robust low-order controller in a linear time-invariant system. The system $P^i(s)$ is controlled by a constant-gain controller $C(s)$ as depicted in Figure 1 where $y_c^i(s)$ is the controlled output vector, $y_s^i(s)$ the measurement output vector, $w^i(s)$ the disturbance input vector and $u^i(s)$ the control input vector. As a consistent notation, the superscript i is used throughout this paper to denote the system model at the i^{th} plant condition. Note that the

controller $C(s)$ is considered to be *fixed*, i.e does not vary with the design condition. It is modelled as a linear time-invariant system of arbitrary order whose formulation accomodates both a feedforward and a feedback controller structures. Robustness requirement in the context of our problem formulation is defined under the conditions that the control system $C(s)$ stabilizes the plant $P^i(s)$ over a class of design conditions ($i = 1, N_p$).

State equations describing the system model $P^i(s)$ of Figure 1 are as follows. Notice that, in the problem formulation, we assume without loss of generalities that all the system states are initially acquiescent. This assumption is not restrictive since one can always establish impulsive inputs $w^i(t)$ together with the appropriate influence matrix to represent any state initial conditions. At the i^{th} plant condition, the system design model is described by equations (1)-(3) below.

State Equations:

$$\begin{cases} \dot{x}^i(t) = F^i x^i(t) + G^i u^i(t) + \Gamma^i w^i(t) \\ x^i(0) = 0 \end{cases} \tag{1}$$

where $x^i(t)$ is a $n \times 1$ plant state vector, $u^i(t)$ an $m \times 1$ control vector, $w^i(t)$ an $m' \times 1$ disturbance-input vector, F^i an $n \times n$ state matrix, G^i an $n \times m$ control distribution matrix and Γ^i an $n \times m'$ input-disturbance distribution matrix.

Measurement Equations:

$$y_s^i(t) = H_s^i x^i(t) + D_{su}^i u^i(t) + D_{sw}^i w^i(t) \tag{2}$$

where $y_s^i(t)$ is a $p \times 1$ measurement vector, H_s^i a $p \times n$ state-to-measurement distribution matrix, D_{su}^i a $p \times m$ control-to-measurement distribution matrix and D_{sw}^i a $p \times m'$ input-disturbance-to-measurement distribution matrix.

Criterion Equations:

$$y_c^i(t) = H_c^i x^i(t) + D_{cu}^i u^i(t) + D_{cw}^i w^i(t) \tag{3}$$

where $y_c^i(t)$ is a $p' \times 1$ criterion vector, H_c^i a $p' \times n$ state-to-criterion distribution matrix, D_{cu}^i a $p' \times m$ control-to-criterion distribution matrix and D_{cw}^i a $p' \times m'$ input-disturbance-to-criterion distribution matrix.

For generality, the disturbances $w^i(t)$ are modeled as outputs of a linear time-invariant system excited by either impulse inputs or white noises. In this manner, one can shape the disturbance signals to have any deterministic responses (e.g filtered step functions, sinusoidal functions, exponentially decayed or growing sinusoidal functions, etc...) or to model stochastic inputs with any given power spectral density functions. At the i^{th} plant condition, the disturbance model is given by equations (4)-(5) below.

Disturbance State Equations:

$$\begin{cases} \dot{x}_w^i(t) = F_w^i x_w^i(t) + \Gamma_w^i \eta^i(t) \\ x_w^i(0) = 0 \end{cases} \tag{4}$$

where $x_w^i(t)$ is a $n' \times 1$ disturbance state vector, $\eta^i(t)$ a $m' \times 1$ vector of either parameterized random impulses (i.e $\eta^i(t) = \eta_o^i \delta(t)$ with $E[\eta_o^i] = 0$, and $E[\eta_o^i \eta_o^{iT}] = W_o$), or white-noise processes $\eta^i(t)$ with zero mean and covariance $E[\eta^i(t)\eta^{iT}(\tau)] = W_o \delta(t - \tau)$. The matrix W_o is an $m' \times m'$ diagonal positive semi-definite matrix, F_w^i an $n' \times n'$ state matrix of the disturbance model and Γ_w^i an $n' \times m'$ input-distribution matrix.

Disturbance Output Equations:

$$w^i(t) = H^i_w x^i_w(t) + D^i_w \eta^i(t) \tag{5}$$

where $w^i(t)$ is a $m' \times 1$ disturbance output vector, H^i_w an $m' \times n'$ disturbance output matrix and D^i_w an $m' \times m'$ direct feedthrough distribution matrix.

State model of the controller $C(s)$ in Figure 1 is that of a linear time-invariant system described by equations (6)-(7) below.

Controller State Equations:

$$\begin{cases} \dot{z}(t) = Az(t) + By^i_s(t) \\ z(0) = 0 \end{cases} \tag{6}$$

where $z(t)$ is a $r \times 1$ controller state vector, A a $r \times r$ state matrix of the controller and B a $r \times p$ measurement-input distribution matrix.

Control Equations:

$$u^i(t) = Cz(t) + Dy^i_s(t) \tag{7}$$

where $u^i(t)$ is an $m \times 1$ feedback control vector, C an $m \times r$ control-output distribution matrix and D an $m \times p$ direct feedthrough matrix.

For control-law synthesis, all the elements of the controller state matrices can be chosen as design parameters and some of them can be left fixed at pre-assigned values. In addition, if needed, linear and nonlinear equality or inequality constraints can be established among the selected design parameters in order to ensure a particular design structure. For convenience in the derivation of the performance index and its gradients with respect to the controller design parameters, we define a matrix C_o that groups all the controller state matrices (A, B, C, D) in one compact form as follows,

$$C_o = \begin{bmatrix} D & C \\ B & A \end{bmatrix}_{(m+r)\times\ (p+r)} \tag{8}$$

Thus, specification of the matrix C_o will completely define the controller state model. Obviously, for the case of a static output-feedback design (i.e the controller order $r = 0$), we simply have $C_o = D$. In Section 8, we will formulate a control design problem based on the minimization of a performance index using the controller $C(s)$ defined in equations (6)-(7).

To examine the entire class of H^2-norm based control problems and to handle the problem of sensitivity to plant modeling uncertainties, we define the objective function given in equations (9) and (10). This formulation turns out to be versatile and well-posed for the setting of a nonlinear constrained optimization problem. However, depending on the types of disturbance model, that is whether the disturbance outputs $w^i(t)$ are responses to impulse or white-noise inputs, different definitions for the objective function are needed. They are given below.

(a) For random impulse inputs $\eta^i(t)$:

$$\begin{aligned} J(t_f) \ &= \ \tfrac{1}{2} \sum_{i=1}^{N_p} \{ W^i_p \int_0^{t_f} e^{2\alpha' t} \\ & E \left[\ y^{iT}_c(t) Q^i y^i_c(t) + u^{iT}(t) R^i u^i(t) \ \right] dt \} \end{aligned} \tag{9}$$

The expectation operator $E[-]$ is over the ensemble of the random variables η_o^i in the parameterized impulse inputs $\eta^i(t) = \eta_o^i \delta(t)$. Control design problems formulated with the above performance index $J(t_f)$ are often classified under the category of deterministic control. Under this category are, for example, the familiar control problems of command tracking control, disturbance rejection of unwanted but known external input signals, implicit and explicit model-following designs, H^2-control to initial conditions and H^∞-control to sinusoidal inputs.

(b) For random white-noise inputs $\eta^i(t)$:

$$J(t_f) = \frac{1}{2} \sum_{i=1}^{N_p} W_p^i E_{\alpha^i} \left[\begin{array}{l} y_c^{iT}(t_f) Q^i y_c^i(t_f) + \\ u^{iT}(t_f) R^i u^i(t_f) \end{array} \right] \tag{10}$$

The expectation operator $E_{\alpha^i}[-]$ is over the ensemble of the random processes defined in the input variables $\eta^i(t)$ for a closed-loop system destabilized by a factor α^i. The destabilization effectively adds a value α^i to the diagonal elements of the closed-loop system matrix. With the given performance index, one can address the entire class of H^2-norm based control design problems. For example, we can solve for the linear quadratic regulator design (LQ), linear quadratic gaussian (LQG) design, loop transfer recovery (LTR),closed-loop transfer recovery (CLTR), model reduction based on a minimization of H^2-norm of the error.

Note that the performance indices given in equations (9) and (10) are evaluated to a finite-time horizon t_f. The use of a finite time plays a significant role in the implementation of a reliable design algorithm for the optimum steady-state solution. It should be recognized that the objective function is well-defined regardless of whether the feedback control-law is stabilizing or not. Furthermore, a class of problems associated with command tracking of neutrally stable or unstable target responses (e.g step and ramp commands, sinusoidal trajectories) are only tractable under the setting of a finite-time objective function but not in the confine of a steady-state objective function where $t_f \to \infty$. In practice, steady-state results, whenever possible, are usually achieved when the terminal time t_f is equal to five or six times the slowest time constant in the closed-loop system responses.

There are other unique features, besides the concept of design to a finite terminal time t_f, that we have incorporated into the design objective function of equations (9)-(10). First of all, this objective function is not the usual quadratic cost function defined in traditional linear optimal control problems. It is instead a weighted average of quadratic performance indices evaluated over the entire set of design conditions ($i = 1, N_p$). Different weights are assigned to each plant condition through the scalar variable W_p^i where $W_p^i \geq 0$. Of course, if $N_p = 1$, then we recover the usual quadratic cost function for a single *nominal* design condition. The time-weighted factor $e^{2\alpha^i t}$ further allows us to impose directly a stability requirement for the closed-loop eigenvalues. Namely, when a steady-state design has been achieved and the optimum objective function is bounded, then closed-loop system eigenvalues for the controllable modes will have real parts less than $-\alpha^i$. Finally, the weighting matrices Q^i and R^i are symmetric and positive *semi*-definite matrices. Note that our solution approach to the minimization of the objective function $J(t_f)$ based on nonlinear optimization does not require the control weighting matrix R^i to be positive definite. In fact, in some design problems such as command tracking and model reduction, an objective function representing simply the tracking or model-matching errors does not include the control term, hence $R^i = 0$.

In this section, we provide analytical expressions for the evaluation of the objective function $J(t_f)$ and its gradients $\partial J/\partial C_o$ with respect to the controller matrix C_o. Details of the derivation can be found in [9]. For simple technicality, the problem formulation assumes that there is no implicit-loop paths within the feedback control system. Namely, the control input $u^i(t)$ or the measurement output $y_s^i(t)$ should not have any direct link to itself. This translates into the conditions that one of the product DD_{su}^i or $D_{su}^i D$ must be zero. This is not restrictive since in practice either actuation or sensor dynamics would be incorporated into the design models and thereby resulting in a system that satisfies the above assumption; or one can simply reformulate an equivalent problem with a set of measurement outputs where $D_{su}^i = 0$. Let's assume without loss of generality that we have the case of $DD_{su}^i = 0$, then the closed-loop system can be written in the form [9] shown in figure 2 or simply,

$$\dot{x}'^i(t) = F'^i x'^i(t) + \Gamma'^i \eta^i(t) \text{ with } x'^i(0) = 0 \tag{11}$$

where

$$[F'^i]_{(n+r+n')\times(n+r+n')} = \begin{bmatrix} F^i + & G^i C & (\Gamma^i + \\ G^i D H_s^i & & G^i D D_{sw}^i) H_w^i \\ B(\mathrm{I} + & A + & B(\mathrm{I} + \\ D_{su}^i D) H_s^i & BC_{su}^i C & D_{su}^i D) D_{sw}^i H_w^i \\ 0 & 0 & F_w^i \end{bmatrix} \tag{12}$$

$$[\Gamma'^i]_{(n+r+n')\times m'} = \begin{bmatrix} (\Gamma^i + G^i D D_{sw}^i) D_w^i \\ B(\mathrm{I} + D_{su}^i D) D_{sw}^i D_w^i \\ \Gamma_w^i \end{bmatrix} \tag{13}$$

$$[H_s'^i]_{p\times(n+r+n')} = [(\mathrm{I} + D_{su}^i D) H_s^i \; D_{su}^i C \; (\mathrm{I} + D_{su}^i D) D_{sw}^i H_w^i] \tag{14}$$

$$[D_s'^i]_{p\times m'} = \left[(\mathrm{I} + D_{su}^i D) D_{sw}^i D_w^i\right] \tag{15}$$

$$[H_c'^i]_{p'\times(n+r+n')} = [H_c^i + D_{cu}^i D H_s^i \; D_{cu}^i C \; (D_{cu}^i D D_{sw}^i + D_{cw}^i) H_w^i] \tag{16}$$

and

$$[C'^i]_{m\times(n+r+n')} = \left[D H_s^i \; C \; D D_{sw}^i H_w^i\right] \tag{17}$$

With these definitions, equations (2), (3) and (7) for $y_s^i(t), y_c^i(t)$ and $u^i(t)$ become

$$y_s^i(t) = H_s'^i x'^i(t) + D_s'^i \eta^i(t) \tag{18}$$

$$y_c^i(t) = H_c'^i x'^i(t) + (D_{cu}^i D D s w^i + D_{cw}^i) D_w^i \eta^i(t) \tag{19}$$

$$u^i(t) = C'^i x'^i(t) + D D_{sw}^i D_w^i \eta^i(t) \tag{20}$$

For a well-posed performance index $J(t_f)$, product of direct feedthrough term $(D_{cu}^i D D_{sw}^i + D_{cw}^i) D_w^i$ in the criterion output $y_c^i(t)$ and the penalty weighting matrix Q^i must be zero. Similarly, product of the direct feedthrough term $DD_{sw}^i D_w^i$ in the control output $u^i(t)$ and the

penalty weighting matrix R^i must also be zero. Under these circumstances, the performance index $J(t_f)$ in equations (9) and (10) can be written as,

$$J(t_f) = \frac{1}{2}\sum_{i=1}^{N_p} W_p^i \, Trace\{L^i(t_f)\Gamma^i W_o^i \Gamma^{iT}\} \tag{21}$$

where

$$\begin{aligned} L^i(t_f) \quad &= \int_0^{t_f} e^{(F^i+\alpha^i I)t} \\ &[H_c^{iT} Q^i H_c^i + C^{iT} R^i C^i] e^{(F^i+\alpha^i I)^T t} dt \end{aligned} \tag{22}$$

In the derivation of the analytical gradients of the performance index $J(t_f)$ with respect to design parameters in C_o of the controller state matrices, it is convenient to express the closed-loop system matrices in terms of C_o explicitly, as suggested in [9],

$$F'^i = F_o^i + (G_o^i + T_2 C_o D_1^i) C_o H_o^i \tag{23}$$

$$\Gamma'^i = \Gamma_o^i + (G_o^i + T_2 C_o D_1^i) C_o D_o^i \tag{24}$$

$$H_c'^i = H_1^i + D_2^i C_o H_o^i \tag{25}$$

$$\begin{aligned} C'^i &= \begin{bmatrix} DH_s^i & C & DD_{sw}^i H_w^i \end{bmatrix} \\ &= T_1 C_o H_o^i \end{aligned} \tag{26}$$

where

$$[F_o^i]_{(n+r+n')\times\,(n+r+n')} = \begin{bmatrix} F^i & 0 & \Gamma^i H_w^i \\ 0 & 0 & 0 \\ 0 & 0 & F_w^i \end{bmatrix} \tag{27}$$

$$[G_o^i]_{(n+r+n')\times\,(m+r)} = \begin{bmatrix} G^i & 0 \\ 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \tag{28}$$

$$[\Gamma_o^i]_{(n+r+n')\times\,m'} = \begin{bmatrix} \Gamma^i D_w^i \\ 0 \\ \Gamma_w^i \end{bmatrix} \tag{29}$$

$$[H_o^i]_{(p+r)\times\,(n+r+n')} = \begin{bmatrix} H_s^i & 0 & D_{sw}^i H_w^i \\ 0 & \mathrm{I} & 0 \end{bmatrix} \tag{30}$$

$$[H_1^i]_{p'\times\,(n+r+n')} = [H_c^i \;\; 0 \;\; D_{cw}^i H_w^i] \tag{31}$$

$$[D_o^i]_{(p+r)\times\,m'} = \begin{bmatrix} D_{sw}^i D_w^i \\ 0 \end{bmatrix} \tag{32}$$

$$[D_1^i]_{(p+r)\times\,(m+r)} = \begin{bmatrix} D_{su}^i & 0 \\ 0 & 0 \end{bmatrix} \tag{33}$$

$$[D_2^i]_{p'\times\,(m+r)} = [D_{cu}^i \;\; 0] \tag{34}$$

$$[T_1]_{m\times\ (m+r)} = [\mathrm{I}\ \ 0] \tag{35}$$

$$[T_2]_{(n+r+n')\times\ (m+r)} = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \tag{36}$$

$$[T_3]_{(p+r')\times\ (n+r+n')} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathrm{I} & 0 \end{bmatrix} \tag{37}$$

It has been shown in [9] that derivative of the performance index $J(t_f)$ with respect to the controller matrix C_o (i.e $\partial\mathcal{J}/\partial C_o$) can be obtained explicitly from the following set of equations,

$$\begin{aligned} &\partial\mathcal{J}/\partial C_o = \\ &\textstyle\sum_{i=1}^{N_p} W_p^i\{(D_2^{i\mathrm{T}}QH_c^{ri} + T_1^{\mathrm{T}}RC^{ri})\mathcal{X}^i(t_f)H_o^{i\mathrm{T}} + \\ &(G_o^i + T_2C_oD_1^i)^{\mathrm{T}}[\mathcal{M}^i(t_f)H_o^{i\mathrm{T}} + \mathcal{L}^i(t_f)\Gamma^{ri}W_o^iD_o^{i\mathrm{T}}] + \\ &T_2^{\mathrm{T}}[\mathcal{M}^i(t_f)H_o^{i\mathrm{T}} + \mathcal{L}^i(t_f)\Gamma^{ri}W_o^iD_o^{i\mathrm{T}}](D_1^iC_o)^{\mathrm{T}}]\} \end{aligned} \tag{38}$$

where

$$\begin{aligned} \mathcal{X}^i(t_f) \ &= \\ &\textstyle\int_0^{t_f} e^{(F^{ri}+\alpha\mathrm{I})t}\,\Gamma^{ri}W_o^i\Gamma^{ri\mathrm{T}}e^{(F^{ri}+\alpha\mathrm{I})^{\mathrm{T}}t}\,dt \end{aligned} \tag{39}$$

$$\begin{aligned} \mathcal{L}^i(t_f) \ &= \\ &\textstyle\int_0^{t_f} e^{(F^{ri}+\alpha\mathrm{I})^{\mathrm{T}}t}\,(H_c^{ri\mathrm{T}}QH_c^{ri} + C^{ri\mathrm{T}}RC^{ri})e^{(F^{ri}+\alpha\mathrm{I})t}\,dt \end{aligned} \tag{40}$$

$$\begin{aligned} \mathcal{M}^i(t_f) \ &= \ \textstyle\int_0^{t_f}\int_0^t e^{(F^{ri}+\alpha\mathrm{I})^{\mathrm{T}}(t-\sigma)}(H_c^{ri\mathrm{T}}QH_c^{ri} + \\ &C^{ri\mathrm{T}}RC^{ri})e^{(F^{ri}+\alpha\mathrm{I})t}\Gamma^{ri}W_o^i\Gamma^{ri\mathrm{T}}e^{(F^{ri}+\alpha\mathrm{I})^{\mathrm{T}}\sigma}\,d\sigma\,dt \end{aligned} \tag{41}$$

From equations (21)-(22) and (38)-(41), evaluation of the performance index $J(t_f)$ and its gradients $\partial\mathcal{J}/\partial C_o$ would require algorithms for efficient computation of the following two types of integrals of matrix exponentials: $\mathcal{X}(t) = \int_0^t e^{A\tau}Be^{C\tau}\,d\tau$ and $\mathcal{M}(t) = \int_0^t\int_0^v e^{A(v-s)}Be^{Cv}De^{Es}\,ds\,dv$. In the next section, we review briefly the numerical algorithm developed in [9] for the computation of $\mathcal{X}(t)$ and $\mathcal{M}(t)$.

3. Current Method for Evaluating $\mathcal{X}(t)$ and $\mathcal{M}(t)$

Previous methods for evaluating $\mathcal{X}(t)$ and $\mathcal{M}(t)$ involve basically the diagonalization of the matrices A, C and E in the exponential functions. The procedure requires the determination of the eigenvalues and eigenvectors of these matrices. It is further assumed for convenience that similarity transformations can be constructed from these eigenvectors to diagonalize the respective matrices. Namely, there exist nonsingular transformations V_A and V_C and V_E such that

$$A = V_A\Lambda_AV_A^{-1}\ ,\ C = V_C\Lambda_CV_C^{-1}\ ,\ E = V_E\Lambda_EV_E^{-1} \tag{42}$$

where the matrices Λ_A, Λ_C and Λ_E are diagonal. Under these assumptions one can express for example the exponential function of e^{At} as

$$e^{At} = e^{V_A\Lambda_AV_A^{-1}t} = V_Ae^{\Lambda_At}V_A^{-1} \tag{43}$$

Usage of this decomposition in the calculation of $\mathcal{X}(t)$ is shown below.

$$\mathcal{X}(t) = \int_0^t e^{A\tau} B e^{C\tau}\, d\tau$$

$$= V_A \left\{ \int_0^t e^{\Lambda_A \tau} \mathcal{B} e^{\Lambda_C \tau}\, d\tau \right\} V_C^{-1} \tag{44}$$

where $\mathcal{B} = V_A^{-1} B V_C$. Advantage of this approach is based on the fact that the exponential function of a diagonal matrix is also diagonal. In this case, time integration in $\mathcal{X}(t)$ can be performed directly by explicit integration of product of scalar exponential functions. The resulting numerical algorithm is quite accurate and efficient, provided that the transformation matrices V_A and V_C are not ill-conditionned. A similar procedure can also be applied to the evaluation of $\mathcal{M}(t)$. Complete discussion can be found in Appendices C and D of [9]. However, breakdown of this algorithm will occur when the matrices A , C or E become degenerate or near degenerate; a situation that becomes eminent when we address control of flexible structures with densely packed modes as demonstrated in the design examples of sections 8 and 9.

Clearly, in order to have a reliable design algorithm for optimal low-order output-feedback control synthesis [9], one must develop a robust numerical scheme to evaluate matrix integrals of the form shown in $\mathcal{X}(t)$ and $\mathcal{M}(t)$ for the case of a *degenerate* system.

4. Alternative Approaches for Solving $\mathcal{X}(t)$ and $\mathcal{M}(t)$

One rather simple approach is to evaluate

$$\mathcal{X}(t) = \int_0^t e^{A\tau} B e^{C\tau}\, d\tau \quad , \tag{45}$$

$$\mathcal{M}(t) = \int_0^t \int_0^v e^{A(v-s)} B e^{Cv} D e^{Es}\, ds\, dv \tag{46}$$

directly using techniques based on numerical quadrature. Efficiency of numerical integration techniques is poor; especially when it requires small integration step size for satisfactory accuracy in the case of stiff system matrices A, C and E. Another possibility is to use some types of algebraic Lyapunov equations for the solution of $\mathcal{X}(t)$ and $\mathcal{M}(t)$. For example, it can be easily shown that the matrix $\mathcal{X}(t)$ can be obtained from the solution of the following Lyapunov equation,

$$A\mathcal{X}(t) + \mathcal{X}(t)C = \left[e^{A\tau} B e^{C\tau} \right]_0^t \tag{47}$$

Solution of equation (47) exists if $\lambda_i(A) + \lambda_j(C) \neq 0$. This condition will not be satisfied in general for arbitrary system matrices A and C. Thus, from practical purposes $\mathcal{X}(t)$ and likewise $\mathcal{M}(t)$ cannot be solved from a scheme based on Lyapunov equations.

Another possible approach is based on the direct use of exponential matrix. It is well-known [12] that convolution integrals involving matrix exponentials, as represented in the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$, can be derived from the matrix exponential of an augmented

matrix. It can be shown that the matrix $\mathcal{X}(t)$ can be derived from a product of the following matrix exponentials,

$$\mathcal{X}(t) = e^{At}\,[\,I \quad 0\,]\,exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} t\right\}\begin{bmatrix} 0 \\ I \end{bmatrix} \tag{48}$$

Thus, computation of $\mathcal{X}(t)$ now involves the computation of a matrix exponential. A simple reliable algorithm for computing the matrix exponential is given in section 5.

In a similar fashion, one can express the matrix $\mathcal{M}(t)$ in terms of a submatrix of a matrix exponential. To see this, we start from its definition

$$\begin{aligned}
\mathcal{M}(t) &= \int_0^t \int_0^v e^{A(v-s)} B e^{Cv} D e^{Es}\, ds\, dv \\
&= \int_0^t e^{-As} \left\{ \int_s^t e^{Av} BeCv\, dv \right\} De^{Es}\, ds \\
&= -\int_0^t e^{-As} \left\{ \int_t^s e^{Av} Be^{Cv} dv \right\} \\
&\quad * De^{Es}\, ds
\end{aligned} \tag{49}$$

Let's perform a change of integration variable $v = t - r$. We have,

$$\begin{aligned}
\mathcal{M}(t) &= \int_0^t e^{-As} \left\{ \int_0^{t-s} e^{A(t-r)} B e^{C(t-r)}\, dr \right\} \\
&\quad * De^{Es}\, ds \\
&= -\int_t^0 e^{A(t-s)} \left\{ \int_0^{t-s} e^{-Ar} B e^{-Cr} dr \right\} \\
&\quad * e^{Ct} D e^{-E(t-s)} d(t-s) e^{Et} \\
&= \int_0^t e^{Aq} \left\{ \int_0^q e^{-Ar} B e^{-Cr}\, dr \right\} \\
&\quad * e^{Ct} D e^{-Eq}\, dq\, e^{Et} \\
&= \int_0^t \left\{ \int_0^q e^{A(q-r)} B e^{Ct/2} e^{-Cr}\, dr \right\} \\
&\quad * e^{Ct/2} D e^{E(t-q)} dq
\end{aligned} \tag{50}$$

Notice that part of the integrand in equation (50) delimited by braces can be replaced by terms involving the exponential of an augmented matrix. This follows simply from results developed for the matrix $\mathcal{X}(t)$. With this substitution, we obtain

$$\begin{aligned}
\mathcal{M}(t) = \int_0^t \Bigg\{ &[I\ 0]\ \exp\left\{ \begin{bmatrix} A & Be^{Ct/2} \\ 0 & -C \end{bmatrix} q \right\} \begin{bmatrix} 0 \\ I \end{bmatrix} \Bigg\} \\
&* e^{Ct/2} D e^{E(t-q)}\, dq
\end{aligned} \tag{51}$$

$$= \int_0^t [I\ 0]\ \exp\left\{ \begin{bmatrix} A & Be^{Ct/2} \\ 0 & -C \end{bmatrix} (t-q) \right\} * \left(\begin{bmatrix} 0 \\ I \end{bmatrix} e^{Ct/2} D \right) e^{Eq}\, dq$$

$$= [I\ 0\ 0] \exp \left\{ \begin{bmatrix} A & Be^{Ct/2} & 0 \\ 0 & -C & e^{Ct/2}D \\ 0 & 0 & E \end{bmatrix} t \right\} \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

In this section, we have shown that the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$ can be formulated in terms of the solutions of some matrix exponentials. Their evaluation depends therefore strongly on the accuracy and reliability of numerical methods for computing matrix exponential. We will present one such algorithm in section 5. However for computational expediency, special consideration must also be taken to ensure the efficiency of the overall scheme when the upper limit t is large and one of the matrices A, C or D is unstable. Also one must economize memory requirements associated with high dimensionality of the augmented matrix when computing the matrix exponential. These considerations will be elaborated in sections 6 and 7 where we give precise algorithms for the computation of the matrices $\mathcal{X}(t)$ and $\mathcal{M}(t)$ respectively.

5. Numerical Method for the Matrix Exponential

Several numerical methods are available for the computation of the matrix exponential [11]. Among all these, an approximation method based on Padé series is found to be satisfactory [12]. An important component in any numerical routine for matrix exponential is the scaling of the matrix argument prior to the series calculation. Due to the simple result that $e^{At} = (e^{At/2})^2$, a scale factor in terms of powers of two (i.e 2^m) is often used. In this scheme, one can recover the actual value of the original matrix exponential by performing m squarings on the matrix exponential of the scaled matrix. The index m is determined based on the desired size for the scaled matrix. In our algorithm, scaling is applied to the original matrix until its ∞-norm $\|A\|_\infty$ falls below $1/2$.

As mentioned above, the preferred series approximation in our computation of the matrix exponential is the Padé series. Let's review some of the unique features associated with the Padé series for the case of a scalar function $\mathcal{F}(z)$. On its most basic terms, it is a rational function of z of a preselected order that approximates the function $\mathcal{F}(z)$. For a given choice of the order of the numerator (say N) and of the denominator (say M), the Taylor series representation of this Padé series must match the power series representation of $\mathcal{F}(z)$ for the first (N+M+1) terms. Namely,

$$\mathcal{F}(z) \sim P_M^N(z) = \frac{\sum_{i=0}^{N} A_i z^i}{\sum_{i=0}^{M} B_i z^i} \tag{52}$$

In fact, the most common form of the Padé series is known as the diagonal sequence where the numerator and the denominator have the same order (i.e $M = N$). While it is known that the Padé series for the matrix exponential (i.e $\mathcal{F}(z) = e^z$) converges only slightly faster

than the Taylor series for a scalar argument, the improvement is more significant for matrix argument. In the matrix case, Padé series involves computation of a numerator matrix $\mathcal{N}(At)$ and of a denominator matrix $\mathcal{D}(At)$. For a diagonal Padé series of order N, we have

$$\begin{aligned}\mathcal{D}(At) &= \mathrm{I} + \tfrac{(2N-1)!\,N!}{(2N)!\,(N-1)!}At \\ &+ \tfrac{(2N-2)!\,N!}{(2N)!\,2!\,(N-2)!}(At)^2 + \cdots \\ &+ \tfrac{(2N-i)!\,N!}{(2N)!\,i!\,(N-i)!}(At)^i + \cdots \\ &+ \tfrac{N!}{(2N)!}(At)^N\end{aligned} \tag{53}$$

and

$$\begin{aligned}\mathcal{N}(At) &= \mathrm{I} - \tfrac{(2N-1)!\,N!}{(2N)!\,(N-1)!}At \\ &+ \tfrac{(2N-2)!\,N!}{(2N)!\,2!\,(N-2)!}(At)^2 - \cdots \\ &+ (-1)^i\tfrac{(2N-i)!\,N!}{(2N)!\,i!\,(N-i)!}(At)^i + \cdots \\ &+ (-1)^N\tfrac{2N!}{(2N)!}(At)^N\end{aligned} \tag{54}$$

The matrix exponential is simply given by

$$e^{At} = \mathcal{D}^{-1}(At)\,\mathcal{N}(At) \tag{55}$$

Invertibility of $\mathcal{D}(At)$ is usually ensured by proper scaling of the matrix argument At.

Another important consideration in the Padé series is its length N. Assuming that the matrix At has been scaled such that $\|At\|_\infty$ is less than $1/2$, the parameter N can be choosen according to [12] such that

$$2^{3-2N}\frac{(N!)^2}{(2N)!\,(2N+1)!} \leq \epsilon \tag{56}$$

where ϵ is a given desired tolerance for accuracy.

With a Padé series of N terms where N is determined from above, the approximation can be thought of as the exact calculation of a matrix exponential for a "nearby" matrix $(At+E)$ where E is the error matrix with $\|E\|_\infty \leq \epsilon\|At\|_\infty$. The relative error of the approximation is bounded by the following inequality,

$$\frac{\|e^{(At+E)} - e^{At}\|_\infty}{\|e^{At}\|_\infty} \leq \epsilon\|At\|_\infty e^{\epsilon\|At\|_\infty} \tag{57}$$

Thus, reducing the $\infty-$norm of the matrix At would indeed improve the numerical accuracy of the matrix exponential. It has also been shown that methods by series approximation yield better accuracy if the matrix argument has been preconditioned. Additional improvement may therefore be gained by first preconditioning the original matrix. Another immediate benefit of lowering the ∞-norm of the matrix being exponentiated is that the actual scaling factor m needed would also be smaller; thereby resulting in a fewer number of matrix multiplications in the squaring procedure. As usual, preconditioning a matrix tends to bring singular values of that matrix closer together (i.e. lower the condition number), thus avoiding situation where scaling factor is predominantly determined by a few large singular values, and causing significant loss of precision related to the set of small singular values. The most common method used in the precondition of a matrix is the Osborne's method [14], which

minimizes the Frobenius norm of that matrix (and thus indirectly lowering its ∞-norm). However, extensive tests conducted so far seems to indicate that preconditioning of a matrix did not yield significant reduction in the ∞-norm and a smaller scaling factor to justify the added computational efforts incurred in the Osborne's method. The procedure of preconditioning a matrix is nonetheless recommended from the point of view of improved accuracy (see [15] and[17]).

In the implementation of our design algorithm for optimal low-order controller synthesis [9], a value of $\epsilon = 10^{-8}$ has been selected requiring therefore a 4-terms Padé series (i.e $N = 4$) in the evaluation of the matrices $\mathcal{X}^i(t)$, $\mathcal{L}^i(t)$ and $\mathcal{M}^i(t)$ of equations (39)-(41). Additional considerations in the implementation of the proposed method for computing $\mathcal{X}(t)$ and $\mathcal{M}(t)$ are given in sections 6 and 7.

6. Detailed Algorithm for Computing $\mathcal{X}(t)$

As seen in the previous section, the matrix $\mathcal{X}(t)$ can be evaluated in terms of a matrix exponential as shown in equation (48). Conceptually, it is a simple and straightforward procedure to compute the matrix exponential of any arbitrary matrix using the Padé series discussed in section 5. However, it becomes a nontrivial task when we try to implement an efficient algorithm that examines carefully the issues related to accuracy, speed and memory requirement. The basic difficulties lie in the fact that the matrix exponential is for an augmented matrix of a particular form,

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} t\right\} = \begin{bmatrix} e^{-At} & e^{-At}\mathcal{X}(t) \\ 0 & e^{Ct} \end{bmatrix} \tag{58}$$

where $A = C^T = F^n + \alpha^i I$ according to our problem in equations (39) and (40) for the matrices $\mathcal{X}^i(t_f)$ and $\mathcal{L}^i(t_f)$ respectively. Clearly, if the system matrix A is stable (i.e all the eigenvalues of the matrix A have negative real parts) then one could easily encounter numerical overflow when evaluating the term e^{-At} even though the matrix integrals $\mathcal{X}(t)$ and $\mathcal{L}(t)$ of interest are perfectly well-behaved. The overflow problem occurs most likely in the final squaring process. To arrive at a feasible approach in the evaluation of $\mathcal{X}(t)$, one needs to examine in details the steps taken in arriving at the matrix exponential of the original matrix starting from that of a scaled matrix (i.e in the squaring process).

Let's assume that one has scaled the input matrix A by $A\Delta t$ where Δt is a reasonably small time interval given by $\Delta t = t/n = t/2^m$. Thus, we need to first evaluate

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} \Delta t\right\} \text{ where } \Delta t = t/n = t/2^m.$$

For notation convenience, we define

$$exp\left\{\begin{bmatrix} -A & B \\ 0 & C \end{bmatrix} \Delta t\right\} = \begin{bmatrix} D & E \\ 0 & F \end{bmatrix}$$

$$= \begin{bmatrix} e^{-A\Delta t} & e^{-A\Delta t}\int_0^{\Delta t} e^{A\tau} B e^{C\tau}\, d\tau \\ 0 & e^{C\Delta t} \end{bmatrix} \tag{59}$$

Furthermore, let $W = exp(A\Delta t) = D^{-1}$. Now we can write our result as follows,

$$\begin{aligned} \mathcal{X}(t) &= W^n[D^{n-1}E + D^{n-2}EF \\ &\quad + D^{n-3}EF^2 + \cdots + EF^{n-1}], \end{aligned} \tag{60}$$

or

$$\mathcal{X}(t) = W[E + WEF + W^2EF^2 + \cdots + W^{n-1}EF^{n-1}]. \tag{61}$$

The above results are produced by performing m squarings of $\begin{bmatrix} D & E \\ 0 & F \end{bmatrix}$ and taking the appropriate submatrix for $\mathcal{X}(t)$. In our application (cf. equations (39)-(40)), the solution would therefore involve products of matrices of size $2(n + r + n')$. Close examination of equation (61) leads to the following algorithm involving only product of matrices of size $(n + r + n')$ with the final result achievable in m steps,

Step 1 :

$P_1 = W, \quad Q_1 = E, \quad R_1 = F$

Step 2 :

$P_2 = P_1^2, \quad Q_2 = Q_1 + P_1Q_1R_1, \quad R_2 = R_1^2$

$\cdots = \cdots, \quad \cdots = \cdots, \quad \cdots = \cdots$

Step m :

$P_m = P_{m-1}^2 \quad Q_m = Q_{m-1} + P_{m-1}Q_{m-1}R_{m-1}, \quad R_m = R_{m-1}^2$

Finally, $\mathcal{X}(t) = WQ_m$. It should be noted that one can "absorb" this extra factor of W $(= e^{A\Delta t})$ into the matrix Q_1 without any change to the above algorithm (i.e starting the above algorithm with $Q_1 = WE$ instead). This removes the need to retain the matrix W throughout the computation.

Finally, one notes that the terms P_i or R_i for $(i = 1, m)$ may underflow and become a null matrix for some i; in particular when the scaling factor is large (i.e m large). When this situation happens, one can simply truncate the series calculation for $\mathcal{X}(t)$ up to the i^{th} step in the above algorithm since all of the significant (and nonzero) terms have already been accumulated into the matrix Q_i.

7. Detailed Algorithm for Computing $\mathcal{M}(t)$

Here the numerical algorithm is a bit involved compared to the one given for the calculation of $\mathcal{X}(t)$. This is largely due to the increased complexity of the argument of the matrix exponential. Following the procedure described in section 6, let's perform a scaling upon the input matrix A by $A\Delta t$ such that computation of the matrix quantities $\mathcal{M}_o, H, J, P, U$ and $W = V^{-1}$ is well-behaved. These quantities are defined from the following matrix

exponential,

$$exp\left\{\begin{bmatrix} A & Be^{Ct/2} & 0 \\ 0 & -C & e^{Ct/2}D \\ 0 & 0 & E \end{bmatrix}\Delta t\right\} = \begin{bmatrix} P & He^{Ct/2} & \mathcal{M}_o \\ 0 & V & e^{Ct/2}J \\ 0 & 0 & U \end{bmatrix} \tag{62}$$

Due to the possible numerical underflow in the matrix $e^{Ct/2}$ for large t, the matrices H and J are computed directly from the following definitions,

$$H = \int_0^{\Delta t} e^{A\tau} B e^{C\tau}\, d\tau e^{-C\Delta t} \tag{63}$$

and

$$J = e^{-C\Delta t}\int_0^{\Delta t} e^{C\tau} D e^{E\tau}\, d\tau \tag{64}$$

However, the computation of $\mathcal{M}_o$ in equation (62) can still underflow due to its explicit dependence on $e^{Ct/2}$. For the calculation of the matrix $\mathcal{M}(t)$, ideally it can be obtained from m squarings of equation (62). If carried out in this manner, potential numerical overflow is eminent since, according to our equation for $\mathcal{M}^i(t_f)$ in (41), we have $A^T = C = E^T = F^i + \alpha^i I$. Hence, if the matrix C is stable, then the matrix exponential $e^{-Ct} = V^n$ will become unbounded. To bypass this difficulty, as in the calculation for $\mathcal{X}(t)$, one needs to conduct the squaring algorithm explicitly. It can be shown that the matrix $\mathcal{M}(t)$ can be computed as

$$\begin{aligned} \mathcal{M}(t) &= P^{n-1}\mathcal{M}_o + P^{n-2}\mathcal{M}_o U \\ &+ P^{n-3}\mathcal{M}_o U^2 + \cdots + P\mathcal{M}_o U^{n-2} \\ &+ \mathcal{M}_o U^{n-1} \\ &+ HW^2J + HW^3JU + PHW^3J \\ &+ PHW^4JU + P^2HW^4J + \cdots \\ &+ P^{n-2}HW^nJ \end{aligned} \tag{65}$$

This formulation no longer involves the matrix V. The above series for $\mathcal{M}$ can be distinguished into two parts—one that contains the matrix $\mathcal{M}_o$ and the other that does not. The terms involving $\mathcal{M}_o$ can be thought of as

$$[I\ \ 0]\begin{bmatrix} P & \mathcal{M}_o \\ 0 & U \end{bmatrix}^n \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{66}$$

which can be performed by m squarings. The remaining terms involving H, J, W, P, and U are computationally intensive and are of the form

$$\sum_{i=0}^{n-2}\sum_{j=0}^{n-2} P^i H W^{2+i+j} J U^j, \quad \text{where } 2+i+j \le n. \tag{67}$$

This equation, owing to the restriction $2+i+j \le n$, is not easily calculated in $\mathbf{m}(= log_2(n))$ steps. A reasonably efficient procedure for computing the final matrix $\mathcal{M}(t)$ is to merge both the easily computed portion given in equation (66) and the more difficult series in equation

(67) into a sequence of m steps, as shown in figure 3. Due to potential numerical underflow, the term W^{i-2} is not accurately obtained from the product W^iV^2 where $V = W^{-1}$. Indeed one needs to recompute the term W^{n+2-2^j} at each step of the above algorithm. This could become the major drawback of our scheme even though we have used an efficient matrix exponentiation routine to compute W^i requiring at most $2 * log_2(i)$ matrix multiplies.

If in addition W^n is zero (or effectively so), restriction on the indices i and j of $2+i+j \leq n$ in equation (67) becomes inconsequential; hence we can express

$$\begin{aligned}&\textstyle\sum_{i=0}^{n-2}\sum_{j=0}^{n-2} P^iHW^{2+i+j}JU^j \;\; = \\ &(H + PHW + P^2HW^2 + \cdots)W^2 \\ &\quad *(J + WJU + W^2JU^2 + \cdots)\end{aligned}$$

resulting in a simpler algorithm involving the following three terms:

$$(a)\;\; [I \;\; 0]\begin{bmatrix} P & \mathcal{M}_o \\ 0 & U \end{bmatrix}^n \begin{bmatrix} 0 \\ I \end{bmatrix}$$

$$(b)\;\; (H + PHW + P^2HW^2 + \cdots + P^{n-2}HW^{n-2})$$

$$(c)\;\; (J + WJU + W^2JU^2 + \cdots + W^{n-2}JU^{n-2})$$

This algorithm can again be computed in m steps as seen in figure 4, but now there is no costly evaluation of a W^k term at each step.

Further simplification of the above algorithm can be achieved if we make use of the fact that we have $A = E = C^T$ (cf. equation (41)) and therefore $U = P = W^T$. If, for some index $j < m$, W^{2^j} (and likewise U^{2^j} and P^{2^j}) is zero or nearly so, then this calculation for $\mathcal{M}(t)$ is reduced to $\mathcal{M}(t) = H_jW^2J_j$ since $M_m = 0$, $H_j = H_m$, and $J_j = J_m$.

In the following sections, we compare the usefulness of the proposed algorithm to the early algorithm presented in [9] in the design of low-order optimal controllers for two flexible mechanical systems .

8. A Simple Two-Mass-Spring Design Problem

Control of flexible mechanical systems has been of interest in recent years [18]. This problem provides us a simple design case where degeneracy in the closed-loop eigensystem can be easily illustrated. The problem is to control the displacement of the second mass by applying a force to the first mass as shown in Figure 5. At the start, it is simple to verify that the basic open-loop system has a pair of degenerate eigenvalues at the origin. Equations for the dynamic model are given below,

$$\begin{aligned} m_1\ddot{y_1} &= k(y_2 - y_1) + u + w \\ m_2\ddot{y_2} &= k(y_1 - y_2) \end{aligned} \tag{68}$$

or

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_1' \\ y_2 \\ y_2' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -k/m_1 & 0 & k/m_1 \\ 0 & 0 & 1 & 0 \\ 0 & k/m_2 & 0 & -k/m_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_1' \\ y_2 \\ y_2' \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_1 \\ 0 \\ 0 \end{bmatrix} (u+w) \tag{69}$$

where $m_1 = m_2 = k_1 = k_2 = 1$. For comparison, we have obtained an optimal second-order controller design of the form,

$$A = \begin{bmatrix} 0 & 1 \\ A_{21} & A_{22} \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [C_{11} \;\; C_{12}]; \qquad D = [D_{11}]$$

using both algorithms. The control design problem is to minimize the following H^2-norm of the closed-loop transfer function $T_{y_2 w}$ between the disturbance w and the displacement y_2 of the second mass through the controller design parameters $A_{21}, A_{22}, C_{11}, C_{12}$ and D_{11}. We start with the following *arbitrary* initial design guess of $A_{21} = -2, A_{22} = -1, C_{11} = 0, C_{12} = 0.5$, and $D_{11} = 0$. Both algorithms converge effectively to the same optimal design gains of $A_{21} = -0.8571, A_{22} = -0.9258, C_{11} = 0, C_{12} = -0.4535$ and $D_{11} = -0.2449$ and with an optimum value $\|T_{y_2 w}\|_2^2 = 7.71838215122$. A summary of the resulting closed-loop eigenvalues is given in Table 1. The main difference between the two algorithms is in the CPU time for the overall computation. Results are obtained for a VAX/VMS-Workstation DEC-3500 as follows: CPU time of $19.59 sec$ with the algorithm based on diagonalization and $97.36 sec$ using the proposed method. The increase in computational load is expected and constitutes the basic trade-off between reliability and speed of the solution algorithm. The proposed algorithm is more reliable and with this advantage does take a bit longer in computational time. With the early algorithm of [9], one cannot initiate the search for an optimal compensator design with zero gains (i.e $A_{21} = A_{22} = C_{11} = C_{12} = D_{11} = 0$) because, in this case, the closed-loop system would have two pairs of degenerate eigenvalues at the origin; one for the rigid-body mode and the other from the open-loop compensator poles. To alleviate this problem, it was suggested that one simply starts with any compensator design (stabilizing or not) that produces initially a non-degenerate closed-loop system. Even with these considerations, it was found that occasionally the algorithm could break down due to the presence of near degeneracies in the closed-loop system matrix. Thus, for a reliable design method, solution algorithm must treat degeneracies as a common occurrence. This situation is more evident in the optimal output-feedback control design for high-order structural models with closely packed flexible modes.

Future consideration would be to develop a hybrid algorithm taking advantage of the computational efficiency of diagonalization when the closed-loop system matrix is not degenerate, and turning to the current algorithm when degeneracies are detected. System degeneracy can be easily checked from the condition number of the eigenvector matrix.

In the next section, we present a design problem where degeneracies occur frequently and therefore it could pose a serious difficulty for the early design algorithm based on diagonalization.

9. The JPL Large Space Structure Control Design

In control problems for large flexible mechanical systems such as space structures, causes of eigenvalue degeneracies are usually more subtle in nature than the simple case presented in section 8 for a two-mass-spring system. The JPL large space structure has been carefully designed to simulate a lightweight, non-rigid and lightly damped structure in a weightless environment [16]. The structure itself resembles a large antenna with a central boom-dish apparatus and an extended dish consisted of hoop wires and 12 ribs (Figure 6). There are two torque actuators (labelled $HA1$ and $HA10$) on the boom and dish structure to control the two angular degrees of freedom in pointing maneuver, and force actuators at four rib root locations (labelled $RA1$, $RA4$, $RA7$ and $RA10$) for vibration control. From the point of view of control design, it is a challenging problem since the plant has many closely spaced modes and is of reasonably high order. There are a total of 30 modes in the basic structural model. The flexible modes are lightly damped with damping ratios ranging from 0.007 to 0.01. The two rigid-body modes have a damping ratio of 0.12. Our design concept is to use two available angular displacement sensors $HS1$ and $HS10$ of the boom-dish apparatus and the two torquers $HA1$ and $HA10$ collocated with these sensors for control synthesis. With this selection, 20 of the flexible modes associated primarily with the rib motion become uncontrollable and unobservable. These modes are removed by modal truncation from our plant synthesis model. Eigenvalues of the remaining 10 modes are shown in Table 2.

An optimal low-order controller is designed to dampen vibration of the antenna to external excitations. To evaluate the effectivenes of the control system, we perform the following test. The entire structure is agitated using the two boom-dish actuators for the first 6.4 seconds with an applied torque in the form of a square wave of 0.8 second in width and with an amplitude of 1 N-m. The control system is then activated right after the excitation has been removed, and responses of the excited structure at the sensors are examined. The design objective is to damp out the induced vibration as fast as possible without excessive use of controls. Note that the natural responses of the structure will take about a few minutes to decay to zero (Figure 8).

For practical implementation, the controller design is choosen to be of 6^{th} order and has the following form,

$$A = \begin{bmatrix} -50 & 0 & A_{13} & A_{14} & A_{15} & A_{16} \\ 0 & -50 & A_{23} & A_{24} & A_{25} & A_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 & 0 \\ 0 & 0 & A_{53} & A_{54} & 0 & 1 \\ 0 & 0 & A_{63} & A_{64} & A_{65} & A_{66} \end{bmatrix} \tag{70}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \\ B_{41} & B_{42} \\ B_{51} & B_{52} \\ B_{61} & B_{62} \end{bmatrix}$$

$$C = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The first two states in the controller model serve as roll-filters, limiting the control bandwidth to less than $50 rad/sec$. In the design optimization, we have a total of 28 design variables: 16 in the controller A matrix and 12 in the B matrix. The objective function for design optimization consists of a sum of weighted H^2-norms of physical response variables observed at different location of the structure. It is of the form

$$J(t_f) = \underset{t_f \to \infty}{Lim} \frac{1}{2} \left\{ \sum_{i=1}^{12} Q_i E_\alpha \left[y_i^2(t_f) \right] + \sum_{j=1}^{2} R_j E_\alpha \left[u_j^2(t_f) \right] \right\} \tag{71}$$

Note that the expectation operator $E_\alpha[-]$ is for a system destabilized by a factor α. Table 3 lists the design variables y_i and their corresponding penalty weightings Q_i. Also given in the table are the control design weightings R_j for the actuators $HA1$ and $HA10$. Responses in the above objective function are evaluated to random disturbances of unit white-noise spectra applied simultaneously at *all* the hub and rib actuators.

The design optimization begins with the following *arbitrary* initial guess on the controller matrices A and B,

$$A = \begin{bmatrix} -50 & 0 & 1 & 0 & 0 & 0 \\ 0 & -50 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -4 & -4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

A destabilization factor α of 0.071 was used to ensure that all the closed-loop eigenvalues have a real part less than -0.071. The optimization fails to converge when a destabilization factor of greater than 0.075 was selected. This difficulty seems to be in moving the modes

at 1.68 Hz under this controller configuration, implying that additional degrees of freedom must be added to the controller structure given in equation (70).

While the optimization convergence itself took 13.5 hours on a VAX/VMS Workstation DEC-3500, the proposed algorithm for the calculation of the objective function and its gradients with respect to the design parameters is robust and leads to well-behaved design convergence. The final optimal values of the A and B matrices are shown in Figure 7. Closed-loop eigenvalues are given in Table 4. Primary improvement is seen in the increased damping of two modes at 0.65 Hz.

Closed-loop responses of the sensor and control variables corresponding to this design are shown in Figure 8. The controlled responses decay to zero in about $20sec$ after the excitation has been removed. Notice that the control torques are within the desired limits of 1 N-m; the results are obtained through adjustment of the control design weights R_j in Table 3. This design example demonstrates the usefulness of a design algorithm for robust low-order controllers using parameter optimization, and the accompanying improvement of solution reliability using the algorithms described in sections 6 and 7 for degenerate systems.

10. Conclusions

Numerical algorithms for computing matrix exponentials and integrals of matrix exponentials have been developed to handle cases where the system matrix is degenerate. Numerical optimization combined with the given algorithms for the evaluation of the cost function and its gradients with respect to the controller design parameters has well-behaved convergence even when the closed-loop system becomes degenerate. These algorithms have been incorporated into a computer-aided-design package for synthesizing optimal output-feedback controllers. Reliability of the algorithm has been demonstrated using typical design problems encountered in the control of flexible structures. Clearly this algorithm when combined with a previous one based on diagonalization would enhance significantly the overall reliability of the optimal design procedure for low-order controllers, thereby providing an effective automated design environment for multivariable control synthesis.

References

[1] Stein, G. and Athans, M., "The LQG / LTR Procedure for Multivariable Feedback Control Design," *IEEE Transactions on Automatic Control*, AC-32, pp.105-114, 1987.

[2] Doyle,J. C., Glover, K. , Khargonekar, P. P. and Francis, B. A., "State-Space Solutions to Standard H_2 and H_∞-Control Problems," *IEEE Transactions on Automatic Control*, Vol. AC-34, No. 8, pp. 831-847, 1989.

[3] Doyle, J. C. and Glover, K., "State-Space Formulae for All Stabilizing Controllers that Satisfy an H_∞-Norm Bound and Relations to Risk Sensitivity," *Systems & Control Letters*, 11, pp. 167-172, 1988.

[4] Stoorvogel, A. A., "The Singular Control Problem with Dynamic Measurement Feedback," *SIAM J. Control and Optimization*, Vol. 29, No. 1, pp. 160-184, 1991.

[5] Stoorvogel, A. A. and Trentelman, H. L., "The Quadratic Matrix Inequality in Singular H_∞-Control with State Feedback," *SIAM J. Control and Optimization*, Vol. 28, No. 5, pp.1190-1208, September 1990.

[6] Levine, W.S. and Athans, M.,"On the Determination of the Optimal Constant Output Feedback Gains for Linear Multivariable Systems," *IEEE Transactions on Automatic Control*, AC-15, pp.44-48, 1970.

[7] Anderson, B. D. O. and Moore, J. B., *Linear Optimal Control*, Englewood Cliffs, Prentice Hall, Inc., New Jersey, 1971.

[8] Makila, P. M., "Computational Methods for Parametric LQ Problems–A Survey," *IEEE Transactions on Automatic Control*, AC-32, No.8, pp.658-671, August 1987.

[9] Ly, U. *A Design Algorithm for Robust Low-Order Controllers*, PhD. Thesis, Department of Aeronautics and Astronautics, Stanford University, November, 1982.

[10] Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H.,*User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming.* Technical Report SOL-86-2, Stanford university, January 1986.

[11] VanLoan, C. F., "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *SIAM Review* 20, pp.801-836, 1978.

[12] Golub, G. H. and VanLoan, C. F., *Matrix Computations.* Johns Hopkins University Press, pp 396-400, 1985.

[13] Bender, C. M. and Orzag, S. A. *Advanced Mathematical Methods for Scientists and Engineers.* McGraw-Hill 1978. pp 383-389.

[14] Osborne, E. E. , "On Preconditioning of Matrices", *Journal of the Association for Computing Machinery*, Los Angeles, CA. pp 338-345. March, 1960.

[15] Parlett, B. N. and Reinsch, C., "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors", *Numerical Math*, 13, pp. 293-304 , 1969 .

[16] Vivian, H.C. , Blaire, P.E., Eldred, D.B., Fleischer, G.E., Ih, C.-H.C., Nerheim, N.M., Scheid, R.E. and Wen, J.T., *Flexible Structure Control Laboratory Development and Technology Demonstration*, JPL Publication 88-29, October 1, 1987.

[17] Westreich, D. , "A Practical Method for Computing the Exponential of a Matrix and its Integral", *Communications in Applied Numerical Methods*, Vol 6, pp. 375-380, 1990.

[18] Wie, B. and Bernstein, D.,"A Benchmark Problem for Robust Control Design," *Proceedings of the 1990 American Control Conference*, May 23-25, 1990.

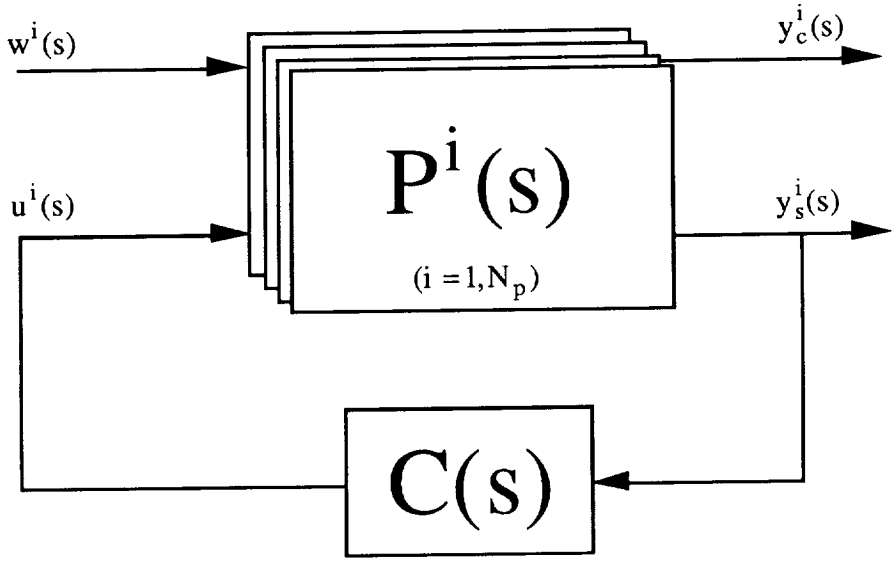


Figure 1: A Typical Closed-Loop System with a Feedback/Feedforward Controller

$$\begin{bmatrix} \dot{x}^i(t) \\ \dot{z}(t) \\ \dot{x}^i_w(t) \end{bmatrix} = \begin{bmatrix} F^i + G^i D H^i_s & G^i C & (\Gamma^i + G^i D D^i_{sw}) H^i_w \\ B(\mathrm{I} + D^i_{su} D) H^i_s & A + B C^i_{su} C & B(\mathrm{I} + D^i_{su} D) D^i_{sw} H^i_w \\ 0 & 0 & F^i_w \end{bmatrix} \begin{bmatrix} x^i(t) \\ z(t) \\ x^i_w(t) \end{bmatrix} + \begin{bmatrix} (\Gamma^i + G^i D D^i_{sw}) D^i_w \\ B(\mathrm{I} + D^i_{su} D) D^i_{sw} D^i_w \\ \Gamma^i_w \end{bmatrix} \eta^i(t)$$

Figure 2: State Model of the Closed-Loop System

Step 0 :					
$\mathcal{M}_o$	H	J	P	U	W
Step 1 :					
$\mathcal{M}_1 = P\mathcal{M}_o + \mathcal{M}_o U$	$H_1 = H$	$J_1 = J$	P^2	U^2	W^2
$+HW^n J$	$+PHW$	$+WJU$			
Step 2 :					
$\mathcal{M}_2 = P^2\mathcal{M}_1 + \mathcal{M}_1 U^2$	$H_2 = H_1$	$J_2 = J_1$	P^4	U^4	W^4
$+H_1 W^{n-2} J_1$	$+P^2 H_1 W^2$	$+W^2 J_1 U^2$			
Step 3 :					
$\mathcal{M}_3 = P^4\mathcal{M}_2 + \mathcal{M}_2 U^4$	$H_3 = H_2$	$J_3 = J_2$	P^8	U^8	W^8
$+H_2 W^{n-6} J_2$	$+P^4 H_2 W^4$	$+W^4 J_2 U^4$			
...	...	...	...	...	...
Step j :					
$\mathcal{M}_j = P^{2^{j-1}} M_{j-1} + M_{j-1} U^{2^{j-1}}$	$H_j = H_{j-1}$	$J_j = J_{j-1}$	P^{2^j}	U^{2^j}	W^{2^j}
$+H_{j-1} W^{n+2-2^j} J_{j-1}$	$+P^{2^{j-1}} H_{j-1} W^{2^{j-1}}$	$+W^{2^{j-1}} J_{j-1} U^{2^{j-1}}$			
...	...	...	...	...	...
Step m :					
$\mathcal{M}_m = P^{n/2} M_{m-1} + M_{m-1} U^{n/2}$	$H_m = H_{m-1}$	$J_m = J_{m-1}$	P^n	U^n	W^n
$+H_{m-1} W^2 J_{m-1}$	$+P^{n/2} H_{m-1} W^{n/2}$	$+W^{n/2} J_{m-1} U^{n/2}$			
$\mathcal{M}(t) = \mathcal{M}_m$					

Figure 3: An m-Step Calculation of $\mathcal{M}(t)$

Step 0 :					
$\bar{\mathcal{M}}_o$	H	J	U	P	W
Step 1 :					
$\bar{\mathcal{M}}_1 = P\bar{\mathcal{M}}_o + \bar{\mathcal{M}}_o U$	$H_1 = H$ $+PHW$	$J_1 = J$ $+WJU$	U^2	P^2	W^2
Step 2 :					
$\bar{\mathcal{M}}_2 = P^2\bar{\mathcal{M}}_1 + \bar{\mathcal{M}}_1 U^2$	$H_2 = H_1$ $+P^2 H_1 W^2$	$J_2 = J_1$ $+W^2 J_1 U^2$	U^4	P^4	W^4
Step 3 :					
$\bar{\mathcal{M}}_3 = P^4\bar{\mathcal{M}}_2 + \bar{\mathcal{M}}_2 U^4$	$H_3 = H_2$ $+P^4 H_2 W^4$	$J_3 = J_2$ $+W^4 J_2 U^4$	U^8	P^8	W^8
...	...	...	...	...	...
Step j :					
$\bar{\mathcal{M}}_j = P^{2^{j-1}}\bar{\mathcal{M}}_{j-1}$ $+\mathcal{M}_{j-1}U^{2^{j-1}}$	$H_j = H_{j-1}$ $+P^{2^{j-1}} H_{j-1} W^{2^{j-1}}$	$J_j = J_{j-1}$ $+W^{2^{j-1}} J_{j-1} U^{2^{j-1}}$	U^{2^j}	P^{2^j}	W^{2^j}
...	...	...	...	...	...
Step m :					
$\bar{\mathcal{M}}_m = P^{n/2}\bar{\mathcal{M}}_{m-1}$ $+\bar{\mathcal{M}}_{m-1}U^{n/2}$	$H_m = H_{m-1}$ $+P^{n/2} H_{m-1} W^{n/2}$	$J_m = J_{m-1}$ $+W^{n/2} J_{m-1} U^{n/2}$	U^n	P^n	W^n
$\mathcal{M} = \bar{\mathcal{M}}_m + H_m W^2 J_m$					

Figure 4: A Simplified m-Step Calculation of $\mathcal{M}(t)$

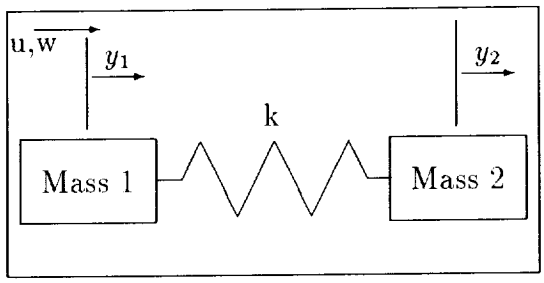


Figure 5: A Two-Mass-Spring Mass System

Eigenvalue	Damping	Freq (Hz)
-0.2290 $\pm$ 0.3397i	0.559	0.0652
-0.1553 $\pm$ 0.8480i	0.180	0.1372
-0.0786 $\pm$ 1.2950i	0.061	0.2065

Table 1: Closed-Loop Eigenvalues of the Two-Mass-Spring System

Eigenvalue	Damping	Freq (Hz)
-0.09500 $\pm$ 0.7860i	0.120	0.12600
-0.08575 $\pm$ 0.7093i	0.120	0.13704
-0.02802 $\pm$ 4.0024i	0.007	0.63701
-0.02929 $\pm$ 4.1844i	0.007	0.66598
-0.07405 $\pm$ 10.583i	0.007	1.68434
-0.07405 $\pm$ 10.583i	0.007	1.68434
-0.11310 $\pm$ 10.616i	0.007	2.57123
-0.11785 $\pm$ 16.384i	0.007	2.67929
-0.21365 $\pm$ 30.520i	0.007	4.85749
-0.21365 $\pm$ 30.520i	0.007	4.85749

Table 2: Open-Loop Modes of the Antenna Structure

Variable	Q_i	Description
$\dot{RS}1$	4100	Rib #1 root velocity
$\dot{RS}4$	3950	Rib #4 root velocity
$\dot{RS}7$	3975	Rib #7 root velocity
$\dot{RS}10$	4050	Rib #10 root velocity
$\dot{HS}1$	16500	Hub angular velocity
$\dot{HS}10$	15600	Hub angular velocity
RS1	1100	Rib #1 root displacement
RS4	1050	Rib #4 root displacement
RS7	1150	Rib #7 root displacement
RS10	1025	Rib #10 root displacement
HS1	3900	Hub angular displacement
HS10	4100	Hub angular displacement
Variable	R_i	Description
HA1	41	Hub torque actuator
HA10	40	Hub torque actuator

Table 3: Design Variables for JPL Antenna Structure

Eigenvalue			Damping	Freq (Hz)
-0.086899	$\pm$	0.6588i	0.1308	0.1058
-0.089071	$\pm$	0.7410i	0.1193	0.1188
-0.3165	$\pm$	3.624i	0.0870	0.5790
-0.2528	$\pm$	3.790i	0.0666	0.6045
-0.2162	$\pm$	4.112i	0.0525	0.6553
-0.2056	$\pm$	4.185i	0.0491	0.6669
-0.074193	$\pm$	10.58i	0.0070	1.684
-0.074589	$\pm$	10.58i	0.0070	1.684
-0.1168	$\pm$	16.15i	0.0072	2.570
-0.1253	$\pm$	16.83i	0.0074	2.678
-0.2142	$\pm$	30.52i	0.0070	4.857
-0.2143	$\pm$	30.52i	0.0070	4.857
-49.99			1.000	7.956
-49.99			1.000	7.956

Table 4: Boom-Dish-Controller Closed-Loop Modes

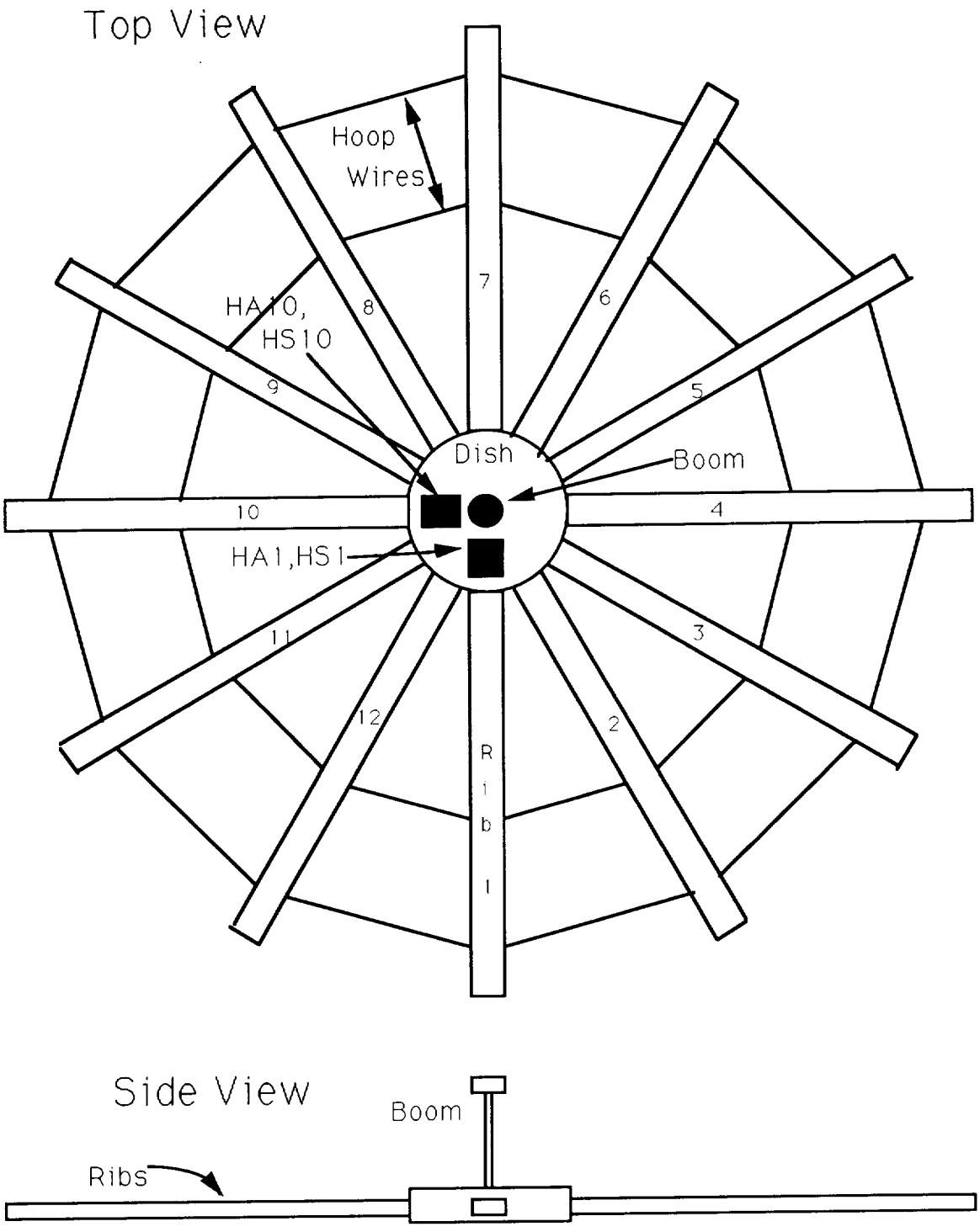


Figure 6: Antenna Structure

$$A = \begin{bmatrix} -50 & 0 & 2.874 & -2.270 & 1.7322 \times 10^{-3} & -2.2131 \times 10^{-4} \\ 0 & -50 & 1.225 & 0.7825 & 6.551 & -1.037 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -15.73 & -0.8799 & 0 & 0 \\ 0 & 0 & 1.560 & 0.2256 & 0 & 1 \\ 0 & 0 & 2.400 & -1.269 & -13.62 & -0.9810 \end{bmatrix}$$

$$B = \begin{bmatrix} 5.343 & -1.2310 \times 10^{-4} \\ 6.2118 \times 10^{-4} & 4.783 \\ 2.701 & -8.1595 \times 10^{-4} \\ 2.221 & 9.3152 \times 10^{-4} \\ -0.5147 & 5.379 \\ 1.614 & -1.252 \end{bmatrix}$$

Figure 7: Optimized Controller Matrices for LSCL Problem

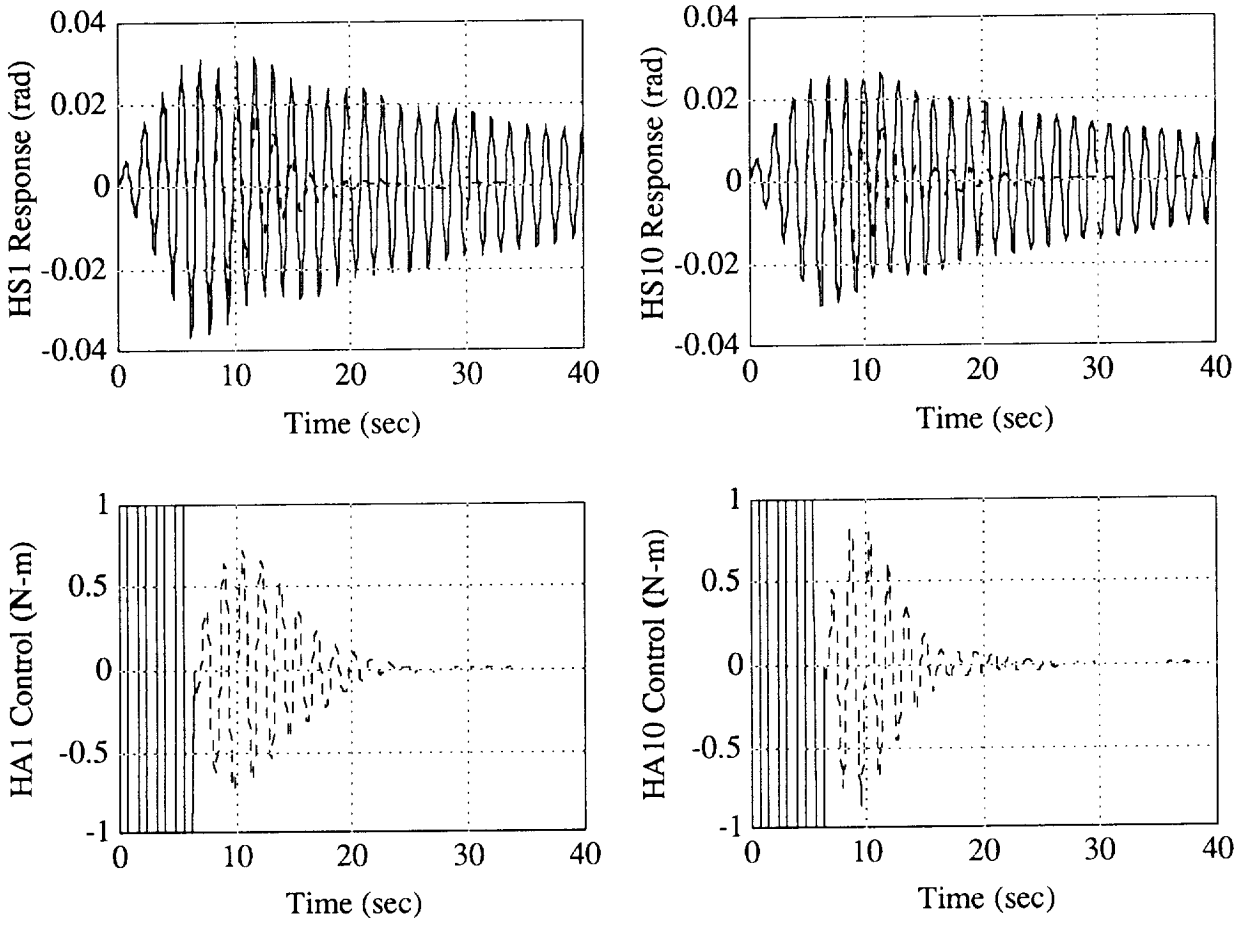


Figure 8: Open-Loop (solid curve) versus Closed-Loop (dashed) Responses

DESIGN CHALLENGES FOR THE UH-60 ROTOCRAFT

Brett VanSteenwyk and Uy-Loi Ly *
Department of Aeronautics and Astronautics, FS-10
University of Washington
Seattle, WA 98195

Abstract

High performance rotocraft controller design is characterized by having to compensate both lateral and longitudinal dynamics in a single controller design. Rather than have these modes largely decoupled in the natural state, often one needs to incorporate mode decoupling as a part of the controller design, especially with the modelling of higher frequency dynamics. There exists the usual concerns of stabilizing the behavior without excessive actuator output, however, in a high performance rotocraft such as the UH-60, good response bandwidth is as important as stabilization and decoupling.

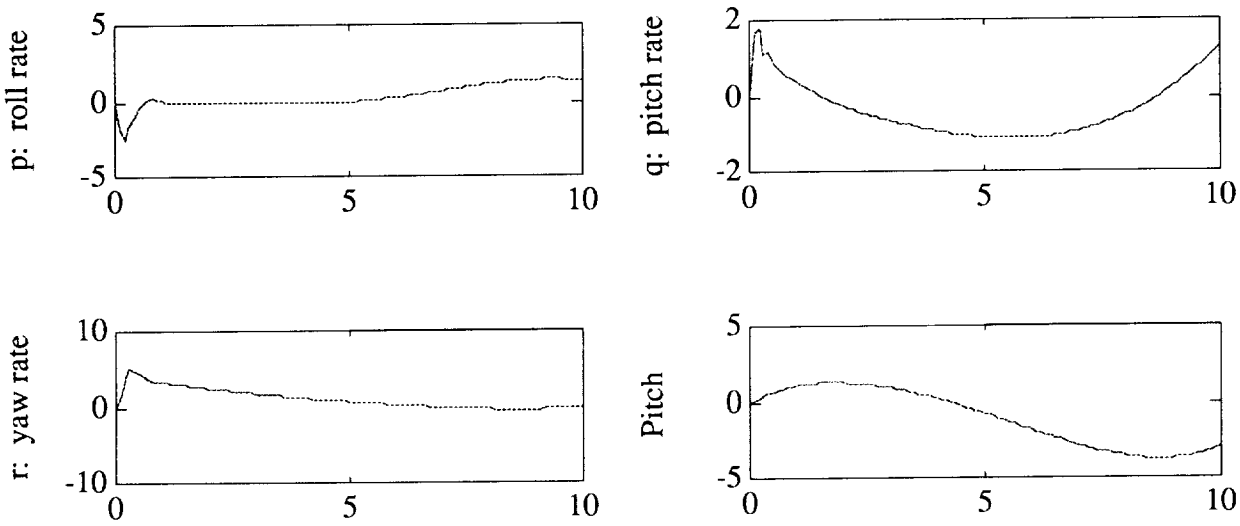
1 Introduction

The challenge is to develop a stabilizing output feedback controller using a 23 state model to simulate the UH-60 rotocraft in the hover flight condition. The essence of this challenge is to develop with a consistent set of techniques that allow a design to meet or surpass flight criteria. In addition to a stabilizing and decoupling controller, command following on ϕ (pitch), θ (roll), r (yaw rate), and w (vertical rate) is required. Sensor outputs include these terms, along with pitch and roll rates, and the forward and transverse velocities (8 sensor inputs total). Optionally, one could control the yaw angle (heading) instead of the rate—this issue is not a fundamental one insofar as a design approach is concerned. Techniques for designing controllers operating with heading should not be distinguished from those operating on yaw rate.

The UH-60 rotocraft open loop model in the hover flight condition is mildly unstable (see Table 1). The pair of unstable poles represent a phugoid-like response in the front/side velocity coupling in with the pitch/roll. In addition, there are modes that are either pure integrating (yaw from yaw rate, for instance), or are near integrating (roll). The sensor output of the model consists of the angular rates p, q, and r, corresponding to the angles θ (pitch), ϕ (roll), and ψ (yaw) which are also considered part of the sensor output. In addition, the translational velocities u, v, and w are considered sensed.

There are 4 actuator inputs: δ_0, δ_s, δ_c, and δ_{TR}. These stand for the collective (main rotor), the main rotor sine, main rotor cosine, and tail rotor pitch inputs, respectively. The response of the UH-60 to a unit collective pulse is shown in figure 1. These graphs give one a sense of the scaling and sensitivity of the various inputs and output terms.

*The work of B. VanSteenwyk and U. Ly is supported in part by NASA Ames Research Center under grant contract NAG-2-691.



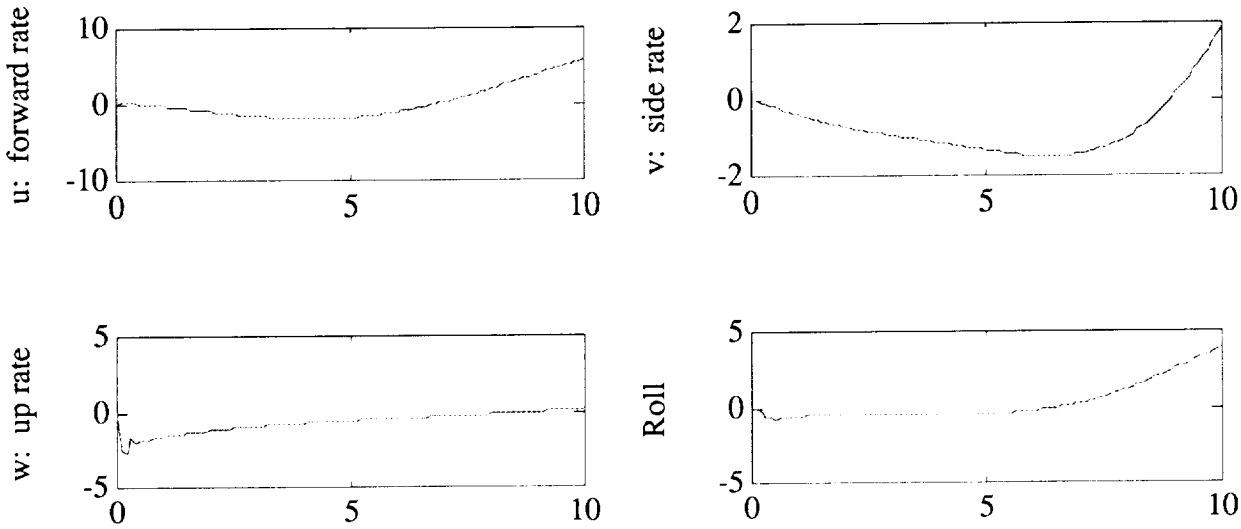


Figure 1: Unit Area Impulse Response

Eigenvalues			Damping	Freq (rad/sec)
-9.38957	±	51.2473i	0.180	52.1003
-5.74275	±	37.1461i	0.153	37.5873
-8.62407	±	24.4863i	0.332	25.9606
-24.5088	±	2.90336i	0.993	24.6801
-4.22353	±	19.6367i	0.210	20.0858
-19.3508			1.000	19.3508
-5.71403	±	5.43266i	0.725	7.88441
-3.26916	±	4.26280i	0.609	5.37205
-4.82826			1.000	4.82826
-1.35460			1.000	1.35460
0.229236	±	0.428933i	-0.471	0.486346
-0.135879	±	0.526819i	0.250	0.54406
-0.230418	±	0.011000i	0.999	0.23068
0.00000			0.000	0.00000

Table 1: Open Loop Eigenvalues of UH-60 Rotocraft

2 Basic LQ Design

What are the limits of full state feedback? Although full state LQ is not always the most satisfactory design technique, in an ideal circumstance it can define the limits of a stabilizing output feedback controller. Suppose one defines the control penalty matrix as the identity and focuses on the most effective output variables to penalize. In addition, there is the issue of controlling yaw rate, or the yaw (heading) itself. Unless noted otherwise, yaw rate is the controlled quantity (this will not change the basic conclusions).

We initially try blindly to penalize all outputs to try to stabilize the system. Given a stabilized system, a feedforward matrix is constructed to try to input commands. It quickly becomes clear that the quality of command response to pitch and roll commands drops if one is trying to stabilize the u and v responses. This seems physically reasonable (pitch forward to increase forward velocity, and so on). Given an R of I, it seems that increasing the response weighting Q will not change the least stable eigenvalues much–they go from $-0.71278 \pm 0.2257i$ to $-0.71289 \pm 0.2251i$ when Q is raised from 10*I to 1000*I. A full listing of the eigenvalues of the closed loop system is given in Table 2 for the case where Q=10*I. Although the slowest poles are not that slow, one has an essentially degenerate pair, and thus one has their persistent behavior. It is unfortunate that they do not move with increasing Q.

It turns out that one can control pitch and roll directly, and thus the ground velocities u and v as a consequence, or one can control the ground velocities with pitch and roll control derived from this. Since pitch and roll are more desired as directly commanded, and it is more reasonable to use these to control ground velocities. The penalty weighting on u and v is dropped. The resulting optimization produces a set of stabilized plant eigenvalues that have worse modes than before; however, it seems that these modes stay much less disturbable and observable from the commanded states. Increasing Q reduces the interaction with these poles without changing their values–they become more exclusively associated with the u and v states, and less with the other responses. Even so, as seen in the command responses, there is some "hangoff" even when the nominal diagonal entries for Q (except **u** and **v**) are 100,000 (see Table 3). Another way of viewing this hangoff is to observe Bode plots of pitch relative to pitch command, etc. (See figures 3 and 4) With the zero frequency dropoff in amplitude of response, one would see the dynamics manifested

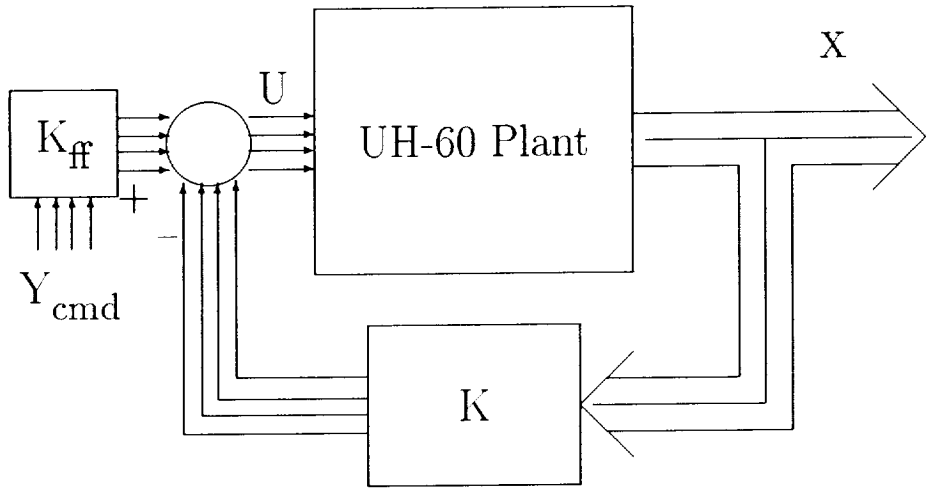


Figure 2: Basic LQ Full State Design Approach

Eigenvalues	Damping	Freq (rad/sec)
-33.3483 ± 48.5324i	0.566	58.8855
-10.9609 ± 53.7911i	0.200	54.8964
-41.2844	1.000	41.2844
-12.3645 ± 28.4117i	0.399	30.9856
-2.97880 ± 17.3773i	0.169	17.6308
-29.3151	1.000	29.3151
-23.0359 ± 4.57704	0.981	23.4862
-18.4648	1.000	18.4648
-9.25176 ± 5.21430	0.871	10.6200
-11.4372	1.000	11.4372
-9.81321	1.000	9.81321
-5.15694 ± 3.57155i	0.822	6.27295
-0.72756 ± 0.17456i	0.972	0.74821
-0.71278 ± 0.22572i	0.953	0.74767

Table 2: Closed Loop Eigenvalues of UH-60 Rotocraft, All Output Penalty, Q = 10*I

Poles			Damping	Freq (rad/sec)
-4558.04			1.000	4558.04
-1537.43			1.000	1537.43
-560.935			1.000	560.935
-104.625	±	88.1763i	0.765	136.826
-89.8398			1.000	89.8398
-74.8621	±	22.1033i	0.959	78.0570
-11.5621	±	53.4352i	0.211	54.6718
-29.4031			1.000	29.4031
-28.8883			1.000	28.8883
-2.57737	±	17.1876i	0.148	17.3798
-10.6260			1.000	10.6260
-10.2186	±	0.749891i	0.997	10.2461
-4.75180	±	5.70887i	0.640	7.42770
-1.03407			1.000	1.03407
-1.00059			1.000	1.00059
-0.965363			1.000	0.965363
-3.98898e-003	±	6.71659e-003i	0.511	7.81182e-003

Table 3: Poles in LQ Stabilized UH-60 Rotocraft System, Q=100,000

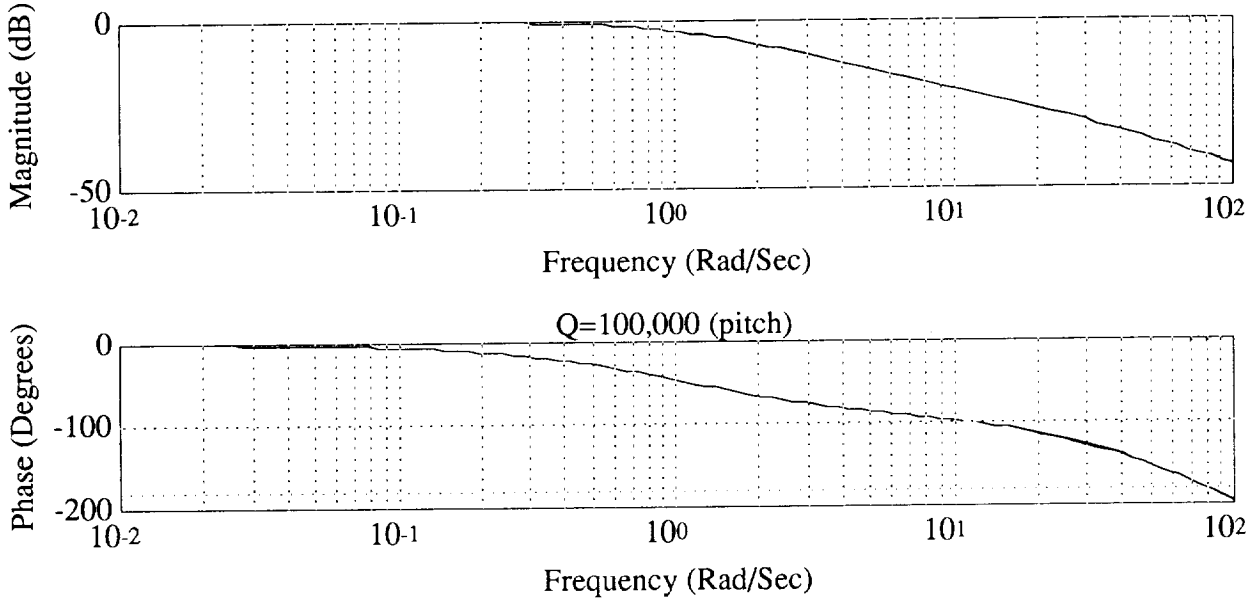


Figure 3: Bode Plot of Pitch/Pitch Command Response, LQ for Q=100,000

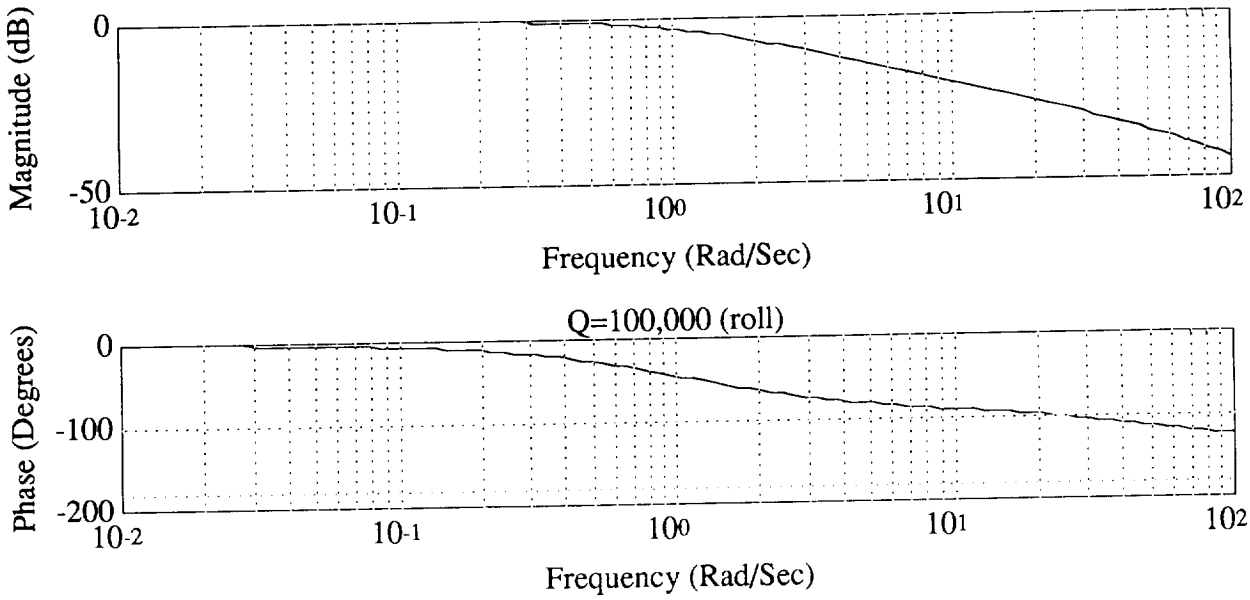


Figure 4: Bode Plot of Roll/Roll Command Response, LQ for Q=100,000

in a hangoff. Note that this problem is not very severe for the yaw and up rate commands, though noticeable.

A ready conclusion was formed–something other than straight penalty on the command outputs was needed. To get decent results, the value of Q (and subsequent actuator activity) had to be stratospheric. There were still minor, but noticeable, hangoffs in the commanded outputs even at values of 100000.

Note the following meaning with u and v being nearly integral poles. It simply provides for their control via pitch and roll commands. To attain a desired forward velocity u, one would pitch forward, allowing the rate to ramp up to a desired value, then pitch level again. It would seem intuitive, then, that it is impossible to regulate the set: $[u, v, \phi, \theta]$ as a whole. It would also seem that the existance of near-integrator poles for u and v is desirable.

3 LQ Design with Integral Control on Pitch and Roll

A ready solution to relieve the hangoff in a command response is to put an integral control state on it. Although for the purpose of synthesis we put the integrators in the plant, in reality they are more states in the controller. Thus any practical implementation will have to contend with these additional states in the controller and the additional numerical challenges they provide. Additionally, integral control is associated with lower bandwith, and thus is to be avoided if possible. It seemed that the pitch and roll commands had the worse of the hangoff problems, so integral control was applied for these commands. In addition, in the interest of eliminating another integrator, the heading state was not included (that the pilot controls heading via yaw rate commands).

The resulting system had 25 states: 23 model states, and 2 integrator states. The existance of integral states allowed the blending of proportional, integral, and derivative states for both pitch and roll to form zero locations that would serve as closed loop system pole attractors. In a frequency

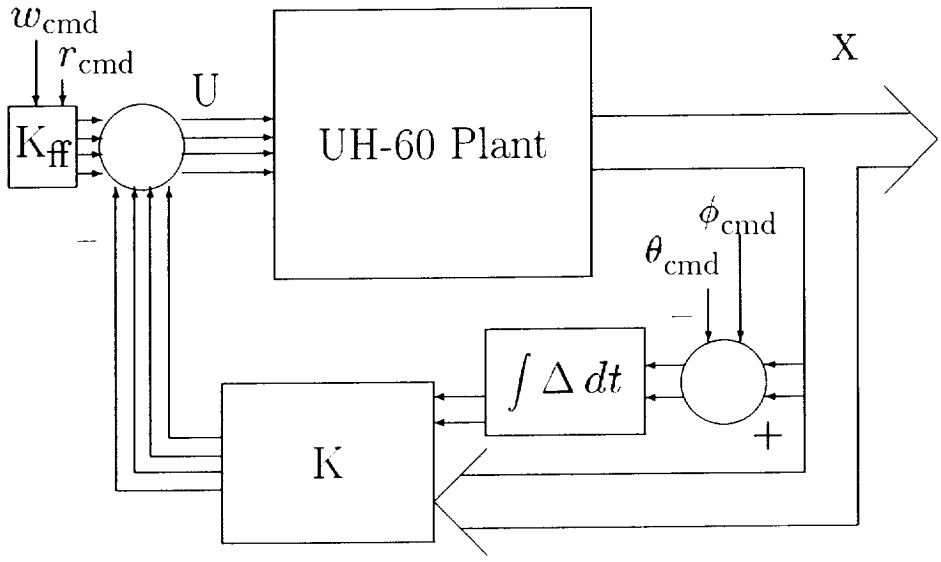


Figure 5: LQ Full State Design with Two Integral Controls

domain sense, one had:

$$a_1 * \int \Delta\theta + a_2 * \Delta\theta + a_3 * \dot{\theta} = \frac{\theta}{s} * \left(a_1 + a_2 s + a_3 s^2\right)$$

To keep the scaling relative to the error constant, one should leave a_2 (the proportional term) as 1. Thus, a_1 and a_3 need to be adjusted so that the zeros associated with this criterion (penalty) will be at desired locations. For the pitch criterion, $a_1 = 1$, $a_2 = 1$, and $a_3 = 0.25$. For the roll criterion, $a_1 = 1$, $a_2 = 1$, and $a_3 = 0.2$. For Q = diag[1,0.1,1.6,2] (θ, ϕ, r, and w), and R = diag[10,1,1,1] (δ_0, δ_s, δ_c, and δ_{TR}), one achieved very satisfactory results, revealed in part by the behavior of the eigenvalues (see table 4). The Bode and time series responses tell the rest of the story (figures 6 to 13).

4 Loop Transfer Recovery

Given the selection of LQ design as the basic approach for the full state case, the use of LTR to select appropriate observer dynamics (full or reduced order) largely presumes an existing set of feedback gains. The degrees of freedom reside with the estimator portion (be it full state or otherwise). The preferred norms and criteria for recovering the full state performance from an output feedback controller should be kept consistent. Thus, since we have started by using H_2, we shall continue to do so.

Poles			Damping	Freq (rad/sec)
-9.38043	±	51.2638i	0.180	52.1149
-6.11002	±	37.2574i	0.162	37.7550
-8.72514	±	24.6011i	0.334	26.1025
-24.5081	±	2.91104i	0.993	24.6804
-4.20761	±	19.5369i	0.211	19.9849
-19.3829			1.000	19.3829
-5.98385	±	5.24203i	0.752	7.95521
-3.75254	±	4.67419i	0.626	5.99413
-6.51433			1.000	6.51433
-4.65048			1.000	4.65048
-1.87052	±	1.37995i	0.805	2.32446
-1.48464	±	0.60051i	0.927	1.60149
-1.70573	±	0.10905i	0.998	1.70922
-3.98898e-003	±	6.71659e-003i	0.511	7.81182e-003

Table 4: Poles for 2 Integral LQ Stabilized System

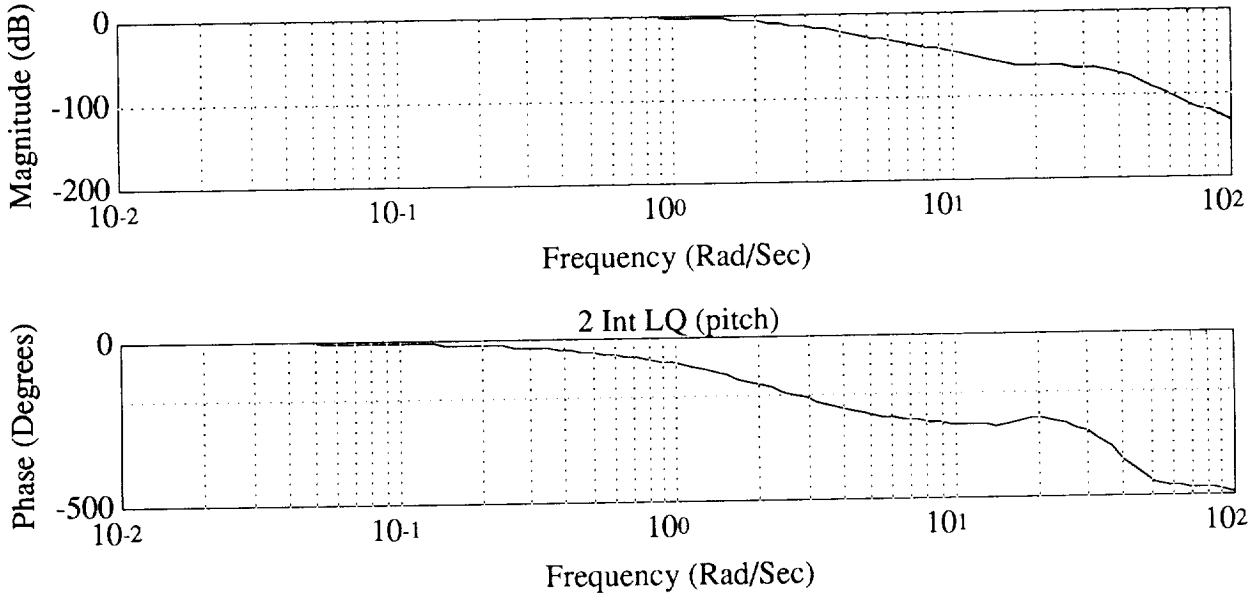


Figure 6: Bode Plot of Pitch/Pitch Command Response, LQ 2 Integral Control

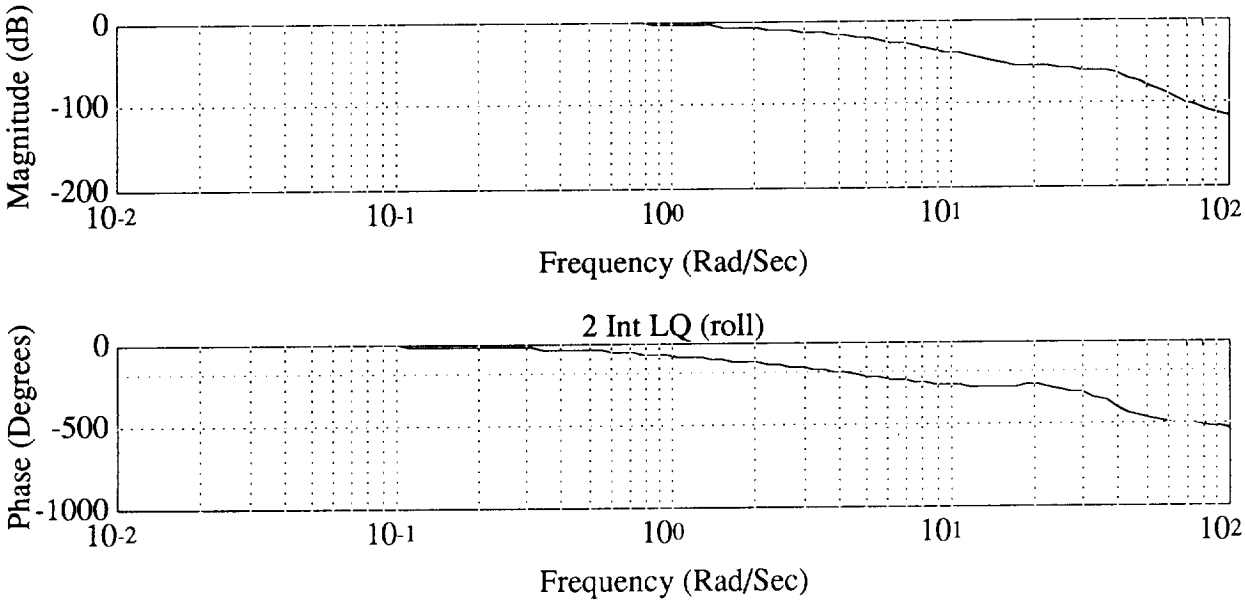


Figure 7: Bode Plot of Roll/Roll Command Response, LQ 2 Integral Control

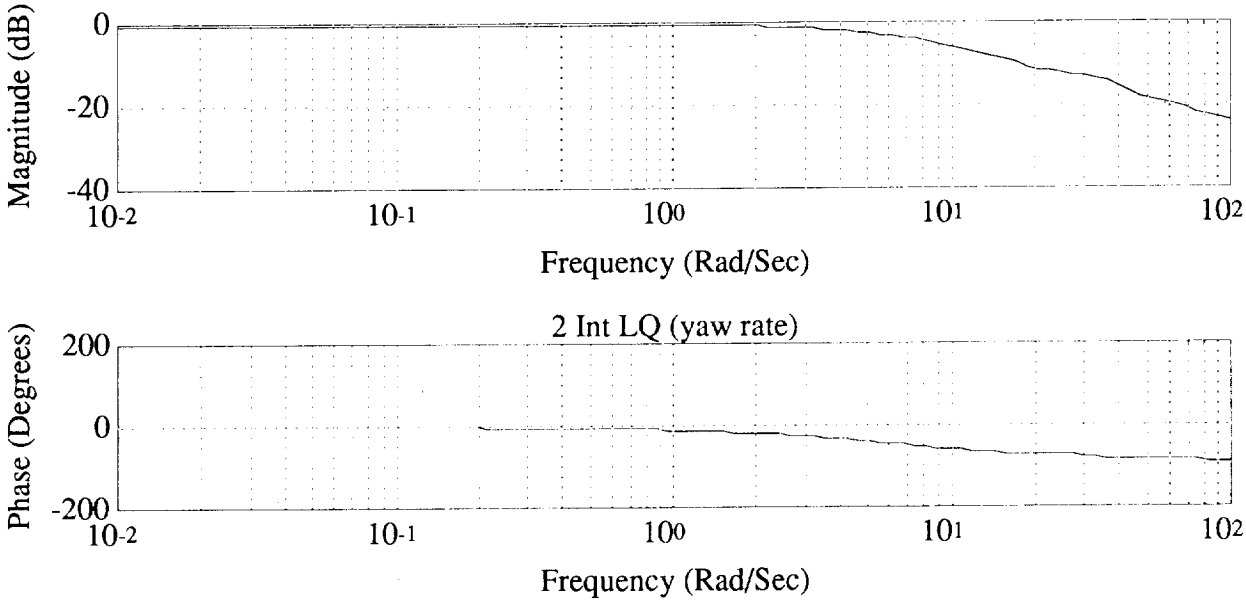


Figure 8: Bode Plot of Yaw/Yaw Rate Command Response, LQ 2 Integral Control

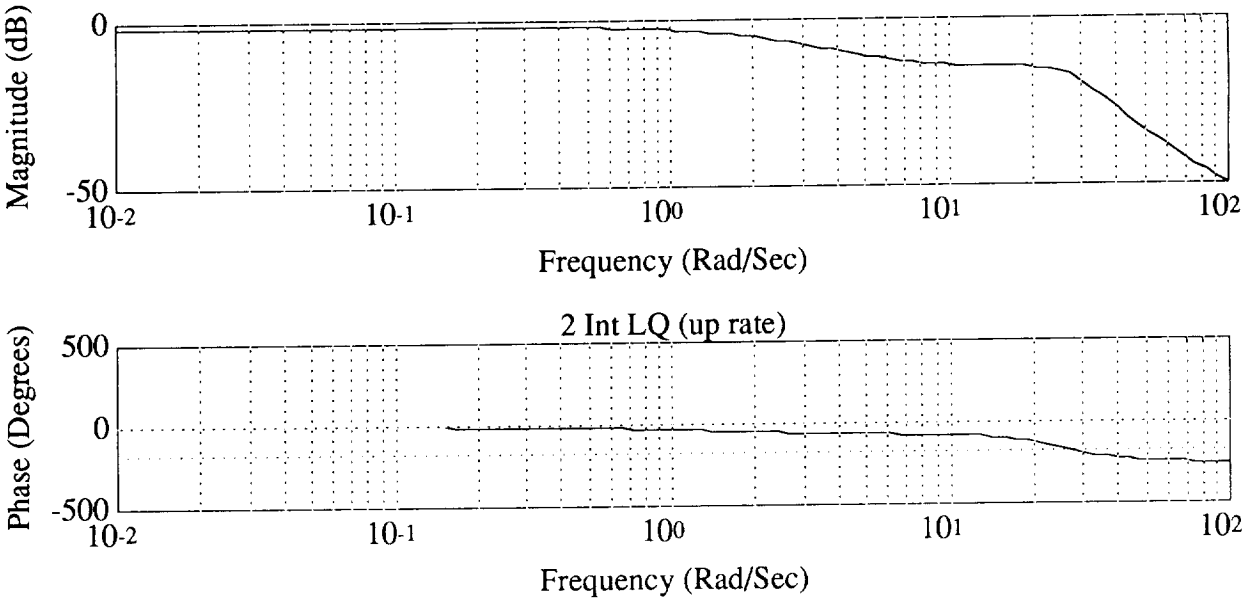


Figure 9: Bode Plot of Up/Up Rate Command Response, LQ 2 Integral Control

The premise of CLTR is simple. Assume the system:

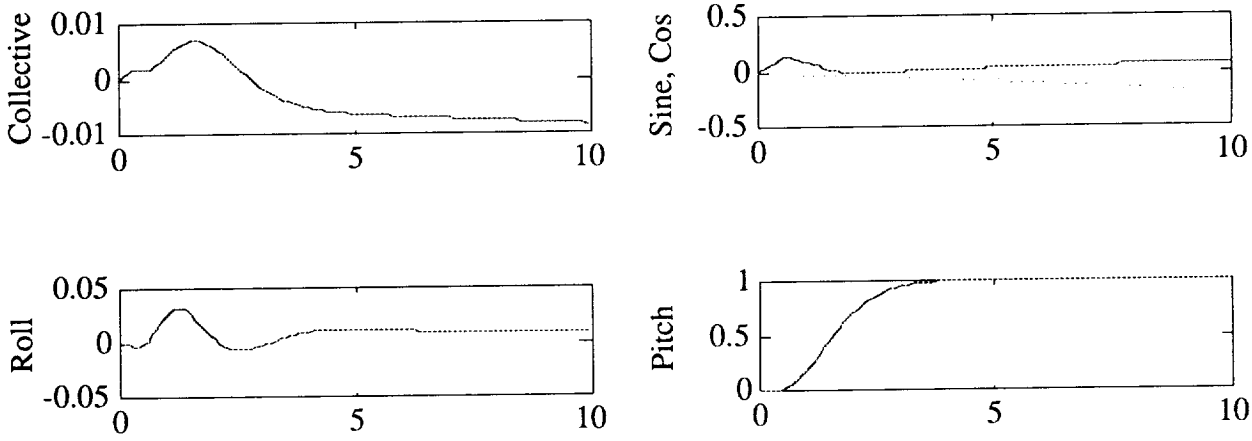
$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \quad \text{(dynamics)} \\ z &= C_1 x + D_{11} w + D_{12} u \quad \text{(criteria)} \\ y &= C_2 x + D_{21} w + D_{22} u \quad \text{(sensors)} \end{aligned}$$

where w is the command/disturbance input and u is the control input. A state feedback matrix F is found that gives the system desired properties. Since state feedback represents only the achievable, but not the actual performance due to noise and limited sensor output, the focus is to add dynamics to synthesize individual or combinations of the necessary states. Within the context of this problem, the disturbance input w is the commanded output (pitch roll, yaw, or vertical velocity), with u remaining as the full set of actuator controls. The objective is to minimize the difference in criteria responses to disturbance (command) inputs for the full state versus output feedback systems (see [4]):

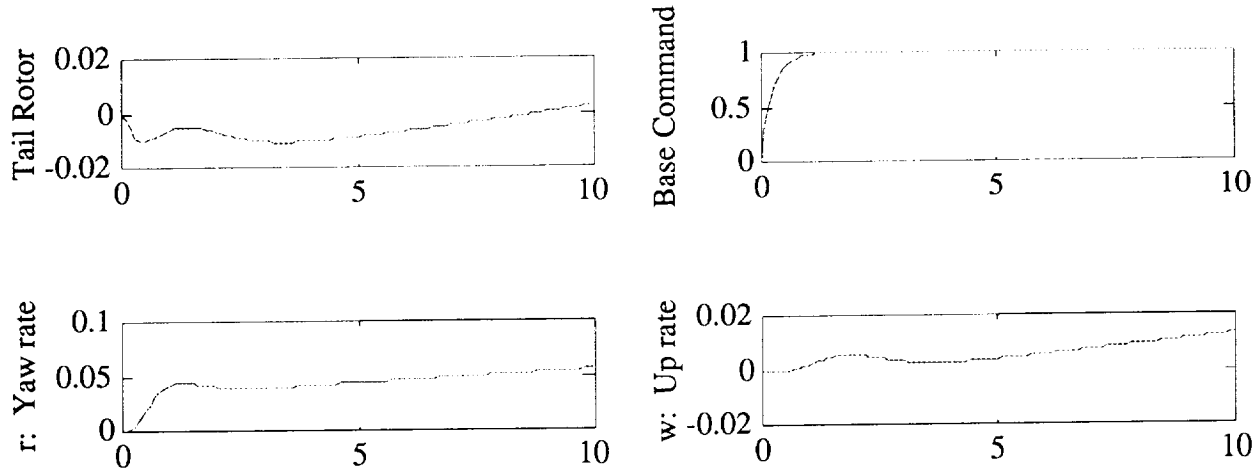
$$T_{zw}(s) - T_{0zw}(s) = E_f(s) = T_{zu}(s) M_f(s)$$

This is generally approached by minimizing $||M_f(s)||_2$ in Ricatti-based methods, while the error function itself can be minimized using direct optimization schemes.

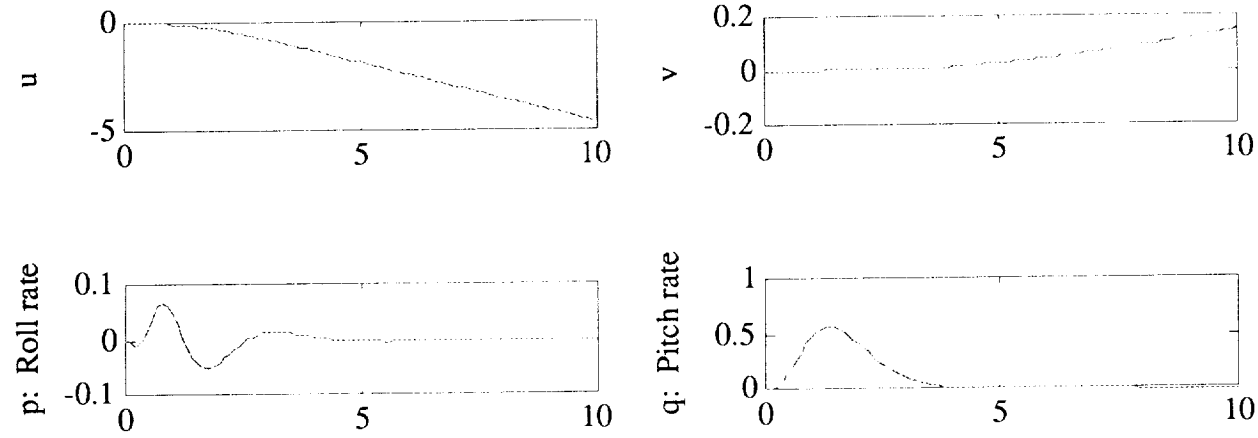
In addition, the direct optimization does allow one to alter some of the state feedback gains. In this case, it was initially held that those states associated with ϕ and θ and their error integrals are held constant as these nominally would not be affected by changes in the rest of the controller. However, for the states associated with the rest of the outputs (p, q, r, u, v, and w), the gains of the feedback matrix are allowed to float. Although the integrators on the command error are treated as part of the plant here, it is understood that they will be a part of the controller. The formal numerical approach induces one to leave it as part of the plant during the design process so as to use the full state feedback matrix as is.



Responses to Pitch Command



Responses to Pitch Command



Responses to Pitch Command

Figure 10: LQ 2 Integral Control

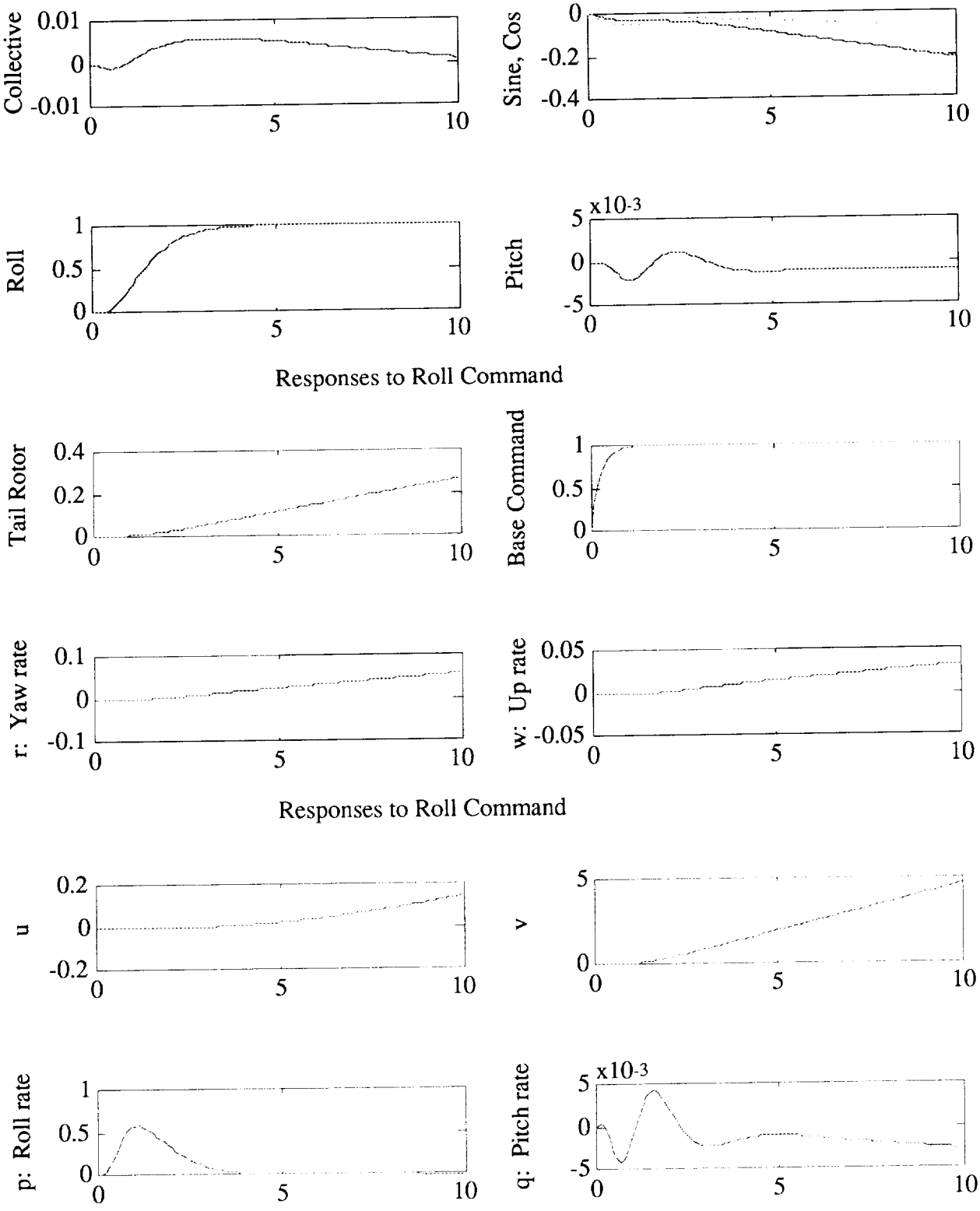
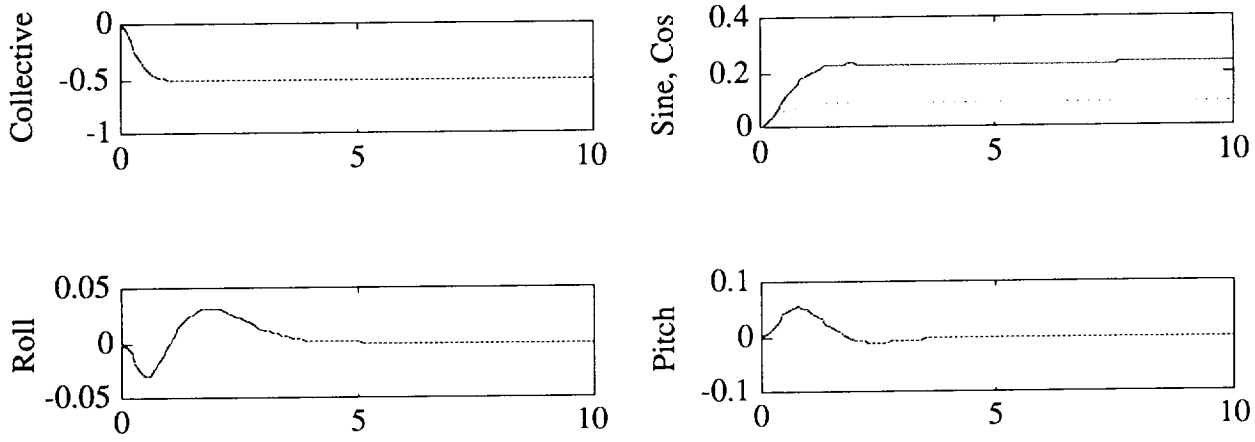
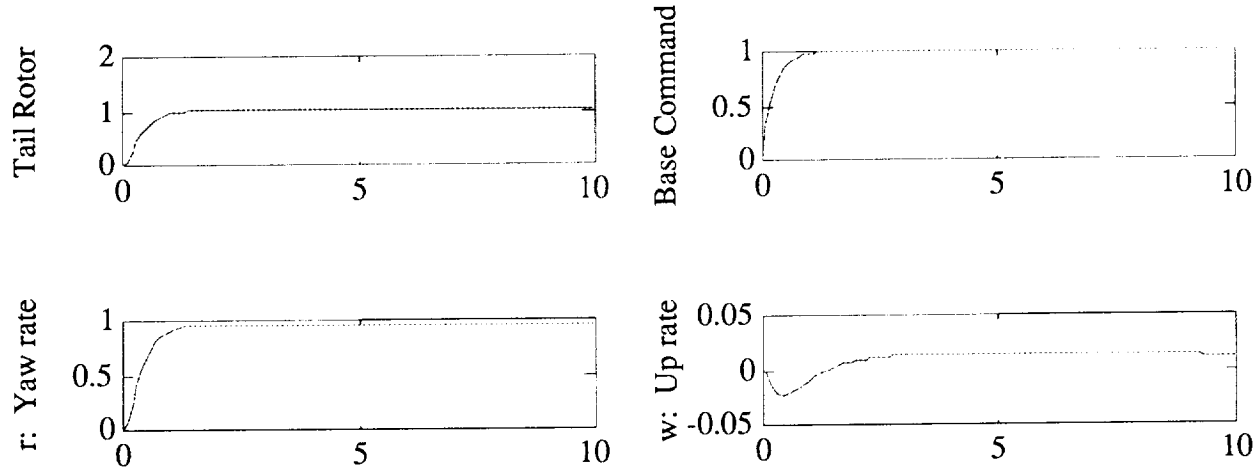


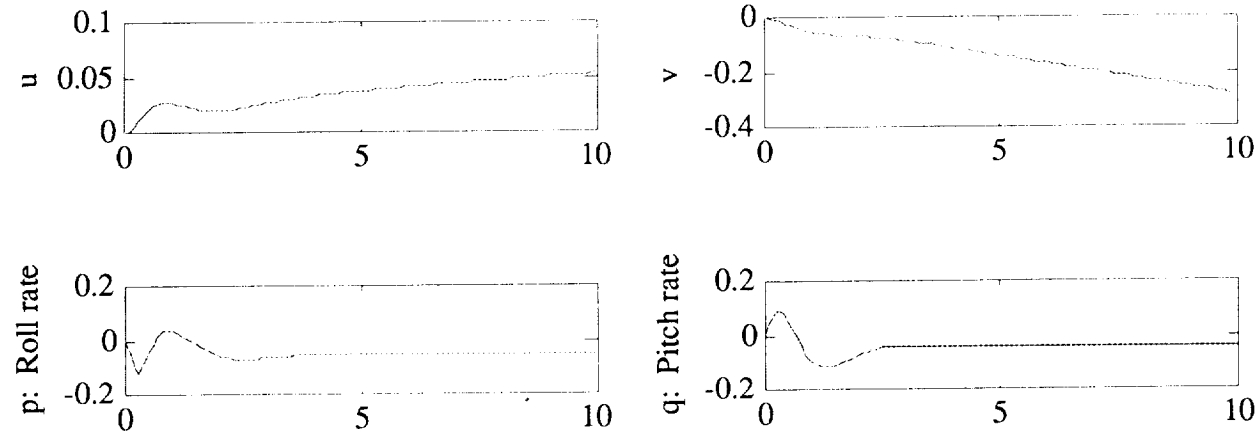
Figure 11: LQ 2 Integral Control



Responses to Yaw Rate Command

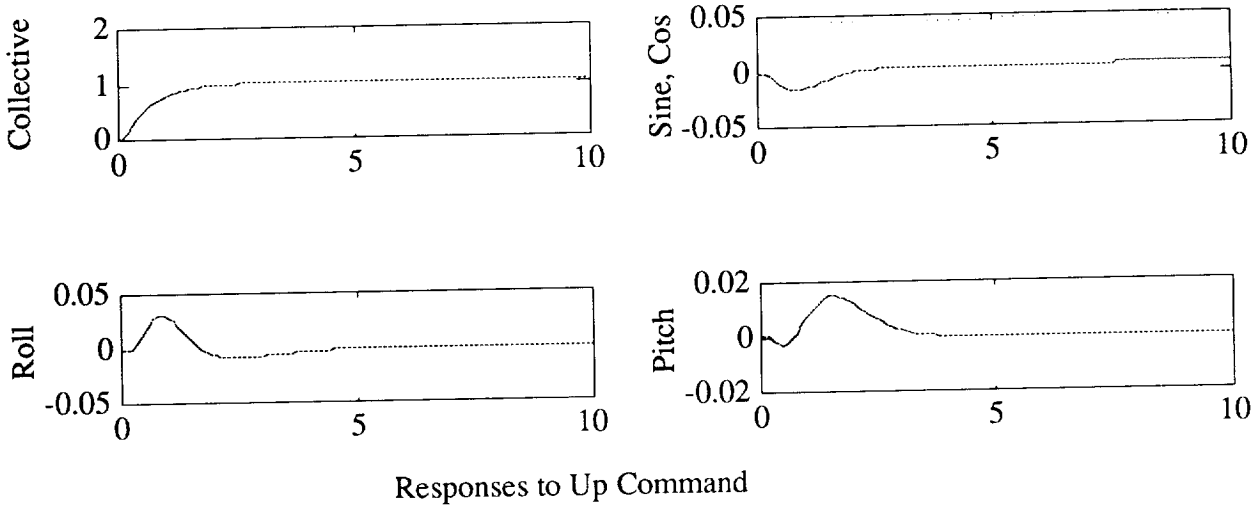


Responses to Yaw Rate Command

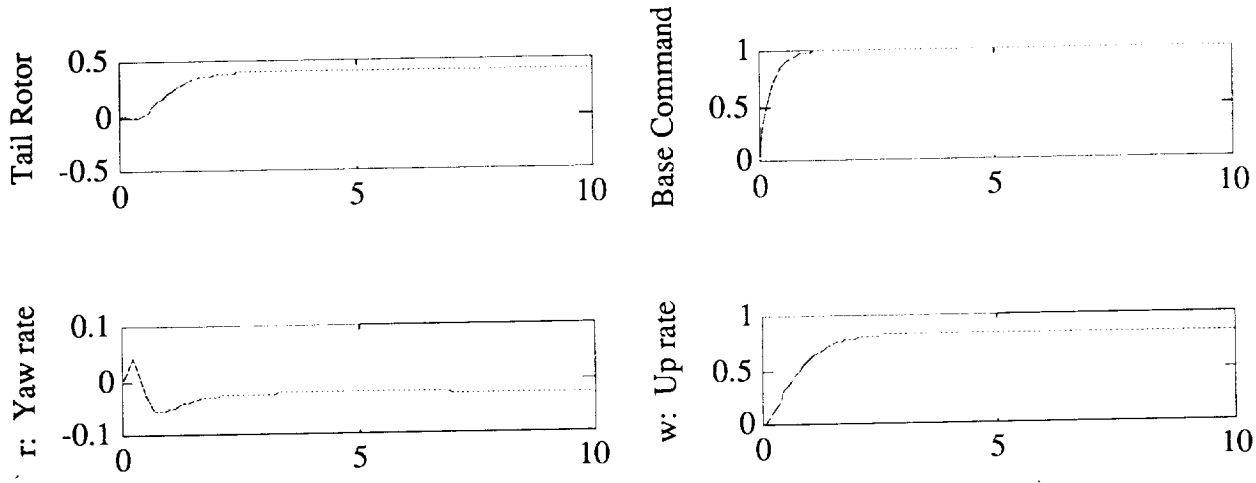


Responses to Yaw Rate Command

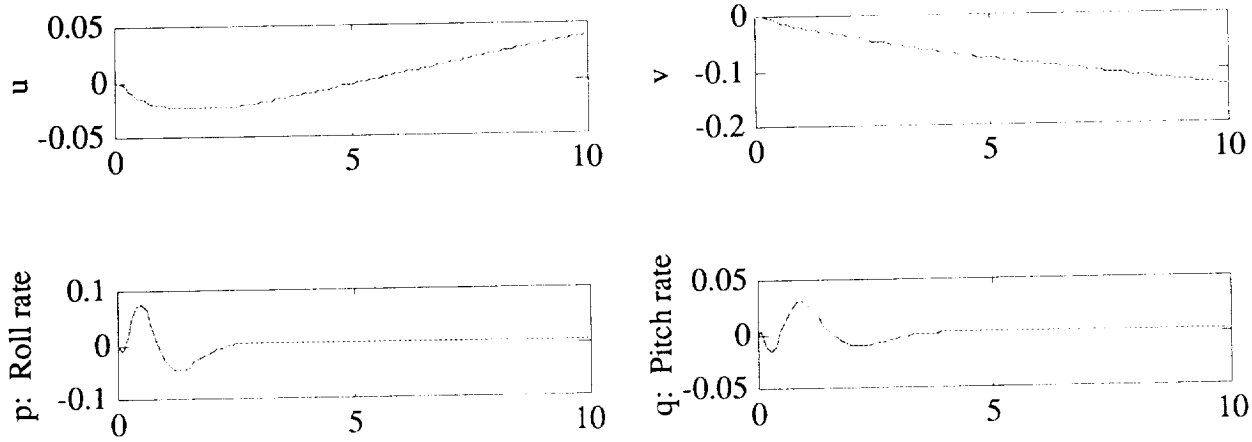
Figure 12: LQ 2 Integral Control



Responses to Up Command



Responses to Up Command



Responses to Up Command

Figure 13: LQ 2 Integral Control

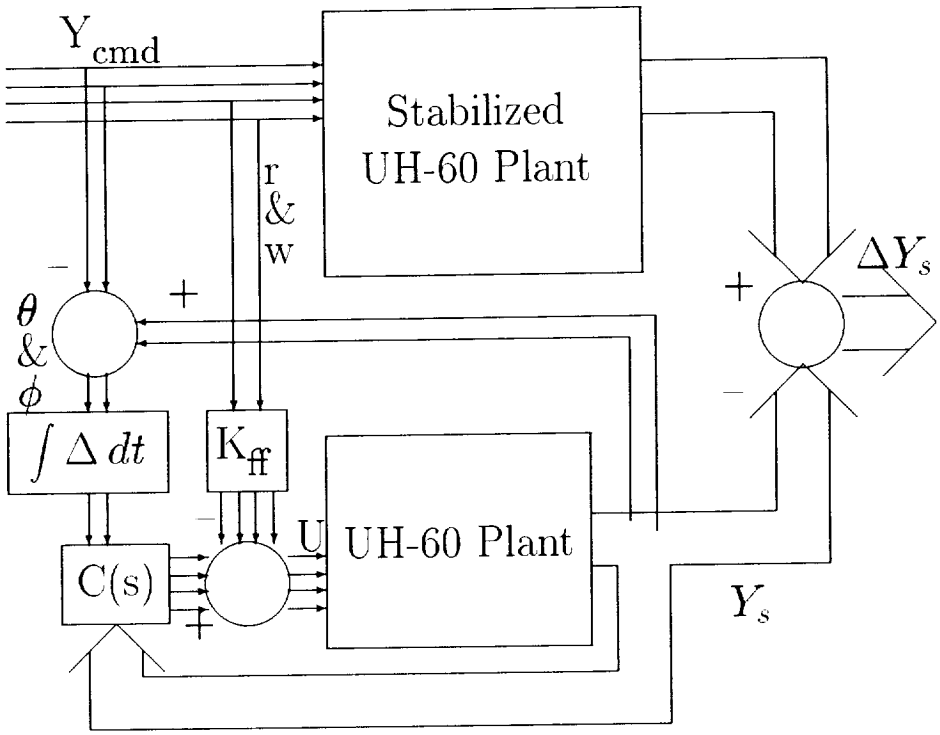


Figure 14: LTR for Output Feedback Controller C(s)

5 Output Feedback Designs from CLTR: The 2 Integral State System

These arise by starting with the full state feedback matrix (or relevant parts thereof) and moving to reduce the error difference between the trial output feedback system and the baseline design produced in Section 3. What results can one expect? Looking at the S.C.B. [3] of the basic plant from the control input to the measured output, we find that it is left invertible and include 4 infinite zeros. There are no invariant zeros. The infinite zeros are probably from the integral control states and combinations of u, v, pitch, and roll. Note that the original state feedback matrix K is preferable for constructing a command feedforward for r and w. While exact recovery is not possible due to some infinite zeros, asymptotic recovery is. The results from the full order and reduced order observer designs are expected to be good. Not much else offhand can be said for the general compensators.

5.1 LTR Design I: Full Order Observer Design

In regards to reducing this error,

$$T^f_{zu}(s)M_f(s) = E_f(s) = T^f_{zw}(s) - T_{0zw}(s),$$

one can compute the error $M_f(s)$ directly in terms of the plant matrices and the estimator matrices:

Plant Structure			Observer Structure		
$\dot{x}$	=	$Ax + B_1 w + B_2 u$	$\dot{\hat{x}}$	=	$(A - KC_2)\hat{x} + B_2 u + Ky$
z	=	$C_1 x + D_{11} w + D_{12} u$	u	=	$-F\hat{x}$
y	=	$C_2 x + D_{21} w + D_{22} u$			

With K is the unknown estimator gain matrix,

$$\begin{aligned} M_f(s) &= F(sI - A + KC_2)^{-1}(B_1 - KD_{21}) \\ M_f^T(s) &= (B_1^T - D_{21}^T K^T)(sI - A^T + C_2^T K^T)^{-1} F \end{aligned}$$

This corresponds to a dual system:

$$\begin{aligned} \dot{\bar{x}} &= A^T \bar{x} + C_2^T u + F^T w \\ z &= B_1^T \bar{x} + D_{21}^T u \text{ where:} \\ u &= -K^T \bar{x} \end{aligned}$$

Minimizing z above in an H_2 sense with respect to K^T is most easily performed through LQ. This is in general a problem with coupled weighting between the states $\bar{x}$ and controls u, and is usually singular.

$$\min_K \int_0^\infty z^T z \, dt = \min_K \int_0^\infty [x \;\; u] \begin{bmatrix} B_1 * B_1^T & B_1 * D_{21}^T \\ D_{21} * B_1^T & D_{21} * D_{21}^T \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt$$

The issue is to deal with the singular part of $D_{21} * D_{21}^T$. If one does the Singular Value Decomposition on D_{21}:

$$D_{21} \rightarrow U_D \Sigma_D V_D^T; \;\; D_{21} * D_{21}^T = U_D \Sigma_D \Sigma_D^T U_D^T$$

where:

$$\Sigma_D \Sigma_D^T = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \cdots & & \\ 0 & \sigma_2^2 & 0 & \cdots & & \\ \vdots & & \ddots & & & \\ 0 & \cdots & & \sigma_n^2 & \cdots & 0 \\ 0 & \cdots & & & 0 & \\ 0 & \vdots & & & & \ddots \end{bmatrix}$$

One must supplement the singular part of this matrix with a small magnitude matrix. The singular part, in practice, is usually not well defined: the singularity is usually relative to the maximum singular value, not just an element being zero. Suppose one chooses an ϵ equal to $0.000001 * \sigma_1^2$. If this magnitude is greater than σ_m^2 for some m, then the matrix to add to $D_{21} * D_{21}^T$ would be:

$$0.000001 * \sigma_1^2 * U_D * \begin{bmatrix} [0]_{(m-1)\times(m-1)} & 0 \\ 0 & [\mathrm{I}] \end{bmatrix} U_T^T$$

One examines the asymptotic properties as $\epsilon \rightarrow 0$. If the open loop system can recover the state feedback properties exactly, then the gain matrix K^T will converge. If not, usually some of these gains will approach infinity.

For this design, K will not converge as ϵ goes to 0. What was done here was to observe the recovery error as a function of ϵ and figure a tradeoff between decrease in error and increase in controller activity. It became simpler than that when it was observed that the response to the yaw rate and vertical rate commands became worse at a point. This seemed to correspond to where the recovery error no longer improved a great deal with decreasing ϵ (the responses to the pitch and roll commands were still improving at this point). A value at 0.0025 was settled upon. The following shows the recovery error for some values of ϵ.

ϵ	Recovery Error
0.01	0.58031
0.005	0.48800
0.0025	0.44232
0.00125	0.43586

The resulting eigenvalues for the chosen design are in Table 5. These tend not to be very sensitive to the value of ϵ chosen and hence do not tell very much. What does tell a lot is a plot of the

Poles			Damping	Freq (rad/sec)
-251.746			1.000	251.746
-33.5097	±	49.0920i	0.564	59.4384
-10.1434	±	52.6143i	0.189	53.5832
-9.38043	±	51.2638i	0.180	52.1149
-46.5182			1.000	46.5182
-6.11002	±	37.2574i	0.162	37.7550
-10.4599	±	27.1390i	0.360	29.0849
-8.72514	±	24.6011i	0.334	26.1025
-5.92371	±	19.6943i	0.288	20.5659
-4.20761	±	19.5369i	0.211	19.9849
-24.4609	±	3.59786i	0.989	24.7241
-24.5081	±	2.91104i	0.993	24.6804
-19.3829			1.000	19.3829
-20.0000			1.000	20.0000
-20.0000			1.000	20.0000
-11.1077			1.000	11.1077
-5.98385	±	5.24203i	0.752	7.95521
-3.84041	±	5.56737i	0.568	6.76345
-3.75254	±	4.67419i	0.626	5.99413
-6.51433			1.000	6.51433
-3.56047	±	3.39555i	0.724	4.92003
-4.70424			1.000	4.70424
-4.65048			1.000	4.65048
-1.87052	±	1.37995i	0.805	2.32446
-1.48464	±	0.60051i	0.927	1.60149
-1.70573	±	0.10905i	0.998	1.70922
-0.16719	±	0.49056i	0.323	0.51827
-0.23179	±	0.41152i	0.491	0.47231
-1.28631			1.000	1.28631
-4.26079e-3	±	6.65864e-3i	0.539	7.90518e-3

Table 5: Full Order Observer Design Eigenvalues, 2 Integral Control

maximum singular value of both the controller activity and of the error with respect to frequency. These are shown for the various values of ϵ in figures 15 and 16.

The design does not end here. Note that this recovery process operates on a matrix M, not the actual recovery error. Perhaps much of the controller activity has gone into regulating high frequency components of the error that will not pass through the plant anyway (remember: $E_f(s) = T^f_{zu}(s)M_f(s)$) Suppose one can augment the transposed system associated with M to

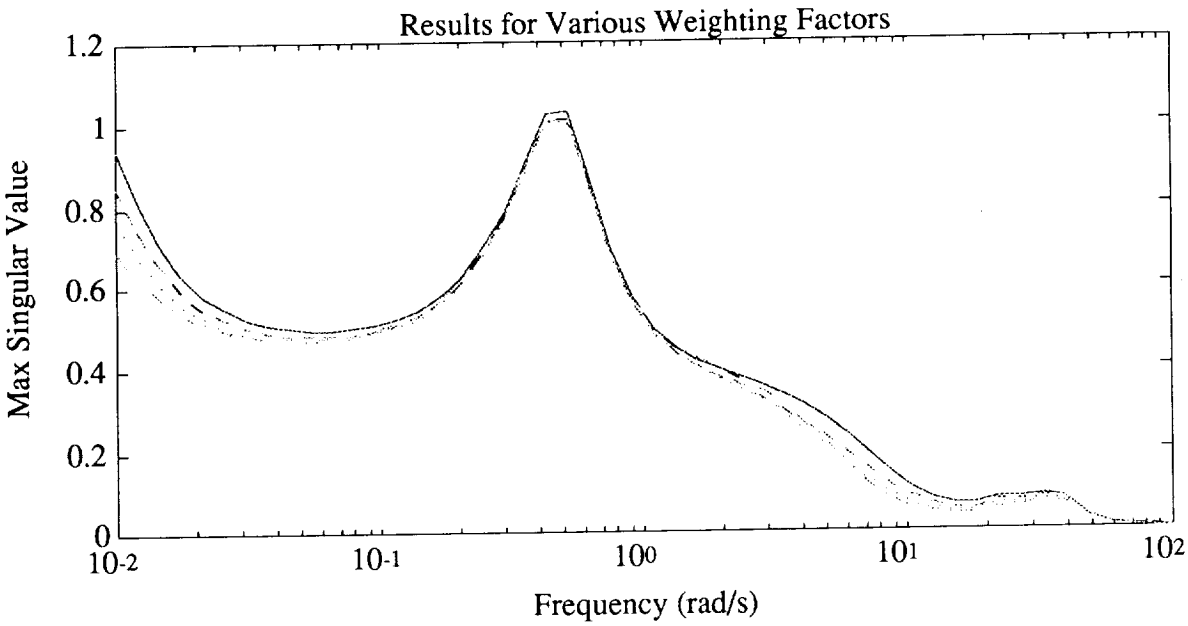


Figure 15: Recovery Error, Full Order Observer

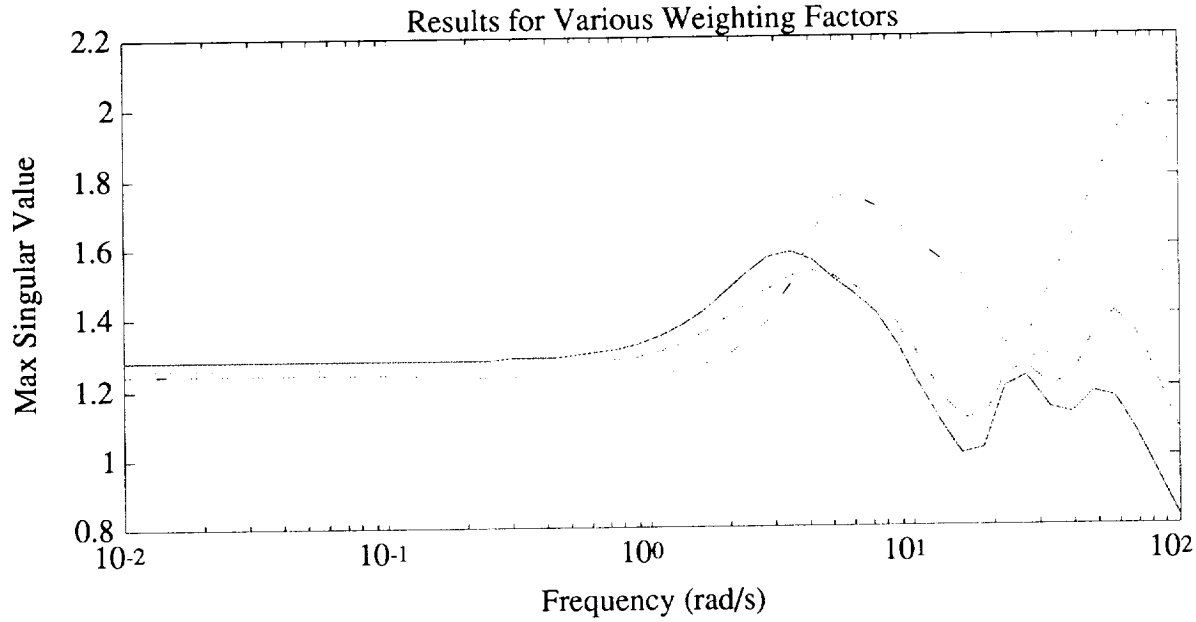


Figure 16: Controller Activity, Full Order Observer

reduce weighting of the subsequent LQ process on the higher frequency components. Thus, the recovery design will do better at the low frequency end at the expense of the high frequency end. Intuition would indicate that putting the corners of these first order filters on the outputs at 8 radians/s for the pitch and roll components and 4 radians/s on the yaw rate and up rate components would make sense. Actually, some testing was done, and a corner of 0.5 radians/s was used for all filters. Because of this, weighting for more controller activity (lower values of ϵ) were called for. Some overall results can be summarized as follows:

ϵ	Recovery Error
0.0001	0.3629
0.00001	0.2777
0.000001	0.2746
0.0000001	0.4136

Poles			Damping	Freq (rad/sec)
-208.040			1.000	208.040
-114.649			1.000	114.649
-47.6094	±	52.9427i	0.669	71.2010
-10.3964	±	52.4826i	0.194	53.5025
-9.38043	±	51.2638i	0.180	52.1149
-62.2535			1.000	62.2535
-62.2535			1.000	62.2535
-6.11002	±	37.2574i	0.162	37.7550
-13.5964	±	27.2780i	0.446	30.4788
-8.72514	±	24.6011i	0.334	26.1025
-4.20761	±	19.5369i	0.211	19.9849
-7.14504	±	18.1857i	0.366	19.5390
-24.5250	±	3.77065i	0.988	24.8131
-24.5081	±	2.91104i	0.993	24.6804
-19.3829			1.000	19.3829
-12.0730			1.000	12.0730
-5.98385	±	5.24203i	0.752	7.95521
-3.19825	±	6.34656i	0.450	7.10687
-3.75254	±	4.67419i	0.626	5.99413
-6.51433			1.000	6.51433
-3.96063	±	2.27644i	0.867	4.56823
-4.88859			1.000	4.88859
-4.65048			1.000	4.65048
-1.87052	±	1.37995i	0.805	2.32446
-1.48464	±	0.60051i	0.927	1.60149
-1.70573	±	0.10905i	0.998	1.70922
-0.66682	±	0.39932i	0.858	0.77724
-0.23421	±	0.46426i	0.450	0.51999
-1.39861			1.000	1.39861
-4.26079e-3	±	6.65864e-3i	0.539	7.90518e-3

Table 6: Full Order Observer Recovery, 2 Integral Control, Weighted

As one can see, this design is much more favorable. The errors at higher frequencies for M_f are not significant in the overall error. Note that now there is a definite minimum at 0.000001 (the chosen ϵ). The eigenvalues are seen in table 6, with plots of the recovery error and controller activity as seen in figures 17 and 18.

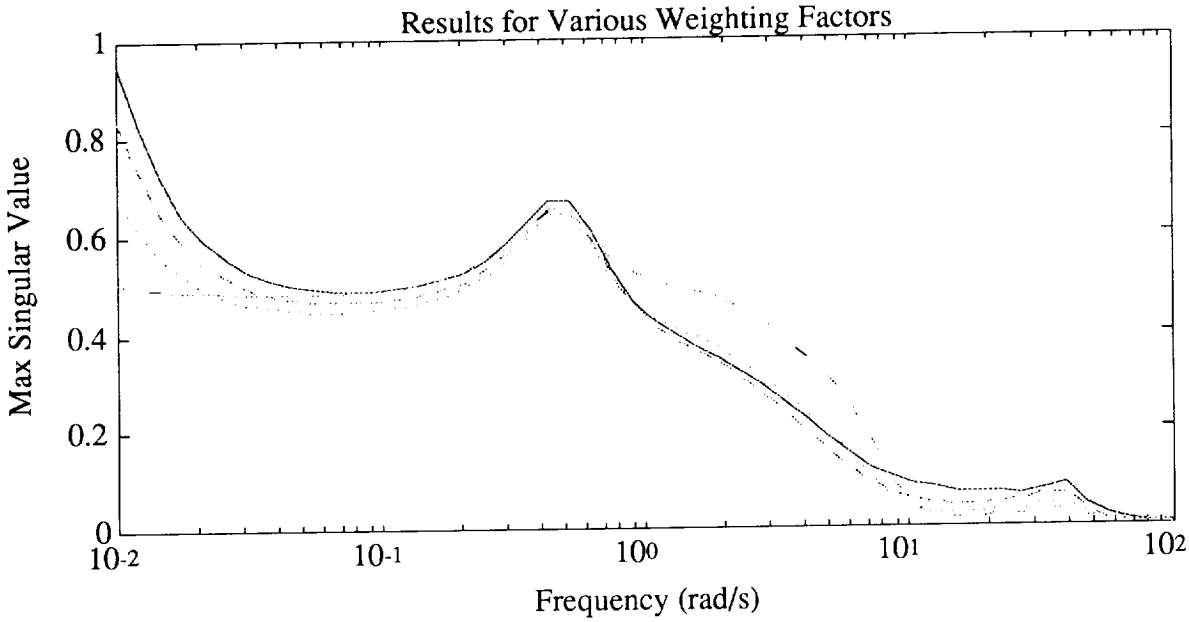


Figure 17: Recovery Error, Shaped Full Order

5.2 LTR Design II: Reduced Order Observer Design

If one were to reconstruct all the states assuming that the measurements were relatively noise-free, then a Luenberger Observer with a minimal number of estimated states would be appropriate. The variables: $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \phi,$ **and** θ are all assumed to be measured (along with the integrals of ϕ and θ), and the remaining 15 states, those of inflow, lag and flap, had to be estimated or otherwise accounted for. Thus, in the general form for the Luenberger Observer:

$$\begin{aligned} \dot{v} &= Lv + G_1 y + G_2 u \\ \hat{x} &= Pv + Jy \\ u &= -F\hat{x} \end{aligned}$$

with:

$$QA - LQ = G_1 C_2, \quad G_2 = QB_2, \text{ and: } \quad JC_2 + PQ = I_{23}.$$

A clever set of transformations exists that will move this problem into the form of a full order observer and a corresponding dual system (see [4], section 4.2). Fortunately for this problem, little has to be done to achieve this form. The direct feedthrough term from the controls to the sensor output is assumed zero. The first step involves a transformation of the original system. The measured outputs become states (unless there is a direct feedthrough from the disturbances to

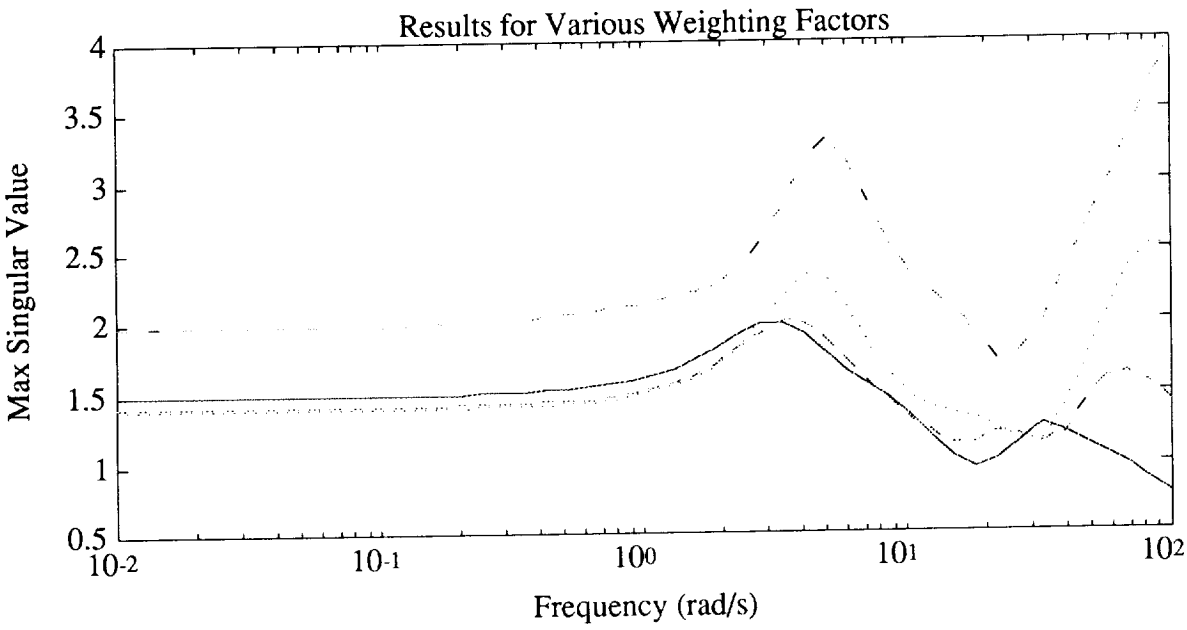


Figure 18: Controller Activity, Shaped Full Order

them). The rest of the states along with the direct feedthrough terms form the rest of the system:

Original System:

$$\begin{aligned} \dot{x}' &= A'x' + B_1'w + B_2'u \\ z &= C_1'x' + D_{11}'w + D_{12}'u \\ y &= C_2'x' + D_{21}'w \end{aligned}$$

becomes:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{1,1} \\ B_{1,2} \end{bmatrix} w + \begin{bmatrix} B_{2,1} \\ B_{2,2} \end{bmatrix} u \\ z &= C_1 x + D_{11} w + D_{12} u \\ \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} &= \begin{bmatrix} 0 & C_{2,02} \\ I_{p-m_0} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_{21,0} \\ 0 \end{bmatrix} w \end{aligned}$$

The UH60 system has neither a D_{21} nor a D_{22} term, so the x_2 states are soley represented by the non-output states. Needless to say, $C_{2,02}$ is non-existant (as is y_0), and no transformation on the state variables is needed since the existing outputs are states already. Defining an observer gain matrix K_r, with partitions $[K_{r0},\ K_{r1}]$ corresponding to $[y_0,\ y_1]$, a few more transformations will yield the Luenberger form:

First define:

$$A_{or} = A_{22} - K_{r0}C_{2,02} - K_{r1}A_{12}$$

Then:

$$\begin{aligned} \dot{v} &= A_{or}v + (B_{2,2} - K_{r1}B_{21})\,u + K_{r0}y_0 \\ &\quad + (A_{21} - K_{r1}A_{11} + A_{or}K_{r1})\,y_1 \end{aligned}$$

$$\begin{aligned} \dot{\hat{x}} &= \begin{bmatrix} 0 \\ I_{n-p+m_0} \end{bmatrix} v + \begin{bmatrix} I_{p-m_0} \\ K_{r1} \end{bmatrix} y_1 \\ u &= -F\hat{x} \end{aligned}$$

Partitioning F along the lines of x: $F = [F_1,\ F_2]$, and with the usual definition of recovery error:

$$E_r(s) = T_{zu}(s) M_r(s),$$

it works out that:

$$M_r(s) = F_2 \left(sI - A_r + K_r C_r\right)^{-1} \left(B_r - K_r D_r\right)$$

with:

$$A_r = A_{22},\ \ B_r = B_{1,2},\ \ C_r = \begin{bmatrix} C_{2,02} \\ A_{12} \end{bmatrix},\ \ D_r = \begin{bmatrix} D_{21,0} \\ B_{1,1} \end{bmatrix}$$

Minimizing M_r is equivalent to finding a state feedback matrix of the transposed system to minimize the responses—the same form as for the full order estimator.

A straightforward LQ/Ricatti solution yields the appropriate observer gain matrix which then must be plugged in to the above equations to generate a dynamic compensator.

ϵ	Recovery Error
500	0.9555
250	0.9007
125	0.9135
62.5	1.0012

Again, looking at these values of the recovery error, as well as plots of the yaw rate command response (which does not necessarily improve with decreasing ϵ), the value of 250 was settled upon. Note that for less than this value, the recovery error does indeed start to rise from this point. Consider that the recovery error presented in the table is no longer just the magnitude of the $M(s)$ matrix. The eigenvalues of the closed loop system corresponding to the $\epsilon = 250$ design are shown in Table 7. Although the order of the system has been reduced by 10, the performance is somewhat dismal. More detail can be seen in the controller activity and recovery error singular value plots (see figures 19 and 20. One can see from these the large (and perhaps unnecessary) increase in activity for higher frequencies. This occurs even with the large values of ϵ used.

Thus, again we seek a way of frequency weighting the error in the transposed system used to generate the observer gains. A fair amount of trial and error led to using a corner of 0.2 rad/s on all criteria for this system. This is probably much lower than what intuition would lead one to. However, it is not very surprising in light of the experience with the frequency weighting on the full order observer. However, the recovery errors are slightly better:

ϵ	Recovery Error
0.5	0.7437
0.25	0.6915
0.125	0.6987
0.0625	0.7421

Still, we have relatively large values of ϵ compared to a weighted full order system, though, in line with intuition, they are somewhat smaller than before. The recovery error for the $\epsilon = 0.25$ was best, though the improvement does not get one near the residual errors for a shaped full state system. In fact, it does not do better than an unweighted full order system. The corresponding eigenvalues for this system are shown in Table 8. Singular value plots of the controller and the recovery error are shown in figures 21 and 22.

Overall there seems to be a fairly high price to going to this reduced order observer structure.

Poles			Damping	Freq (rad/sec)
-7.83040	±	51.6865i	0.150	52.2762
-9.38043	±	51.2638i	0.180	52.1149
-12.4905	±	41.0027i	0.291	42.8630
-6.11002	±	37.2574i	0.162	37.7550
-12.7108	±	25.2826i	0.449	28.2979
-8.72514	±	24.6011i	0.334	26.1025
-24.5081	±	2.91104i	0.993	24.6804
-24.5195	±	2.27973i	0.996	24.6252
-3.71934	±	19.6099i	0.186	19.9595
-4.20761	±	19.5369i	0.211	19.9849
-19.3829			1.000	19.3829
-17.5160			1.000	17.5160
-6.90951	±	4.91729i	0.815	8.48063
-5.98385	±	5.24203i	0.752	7.95521
-6.51433			1.000	6.51433
-3.75254	±	4.67419i	0.626	5.99413
-5.79405	±	0.39518i	0.998	5.80751
-4.65048			1.000	4.65048
-1.87052	±	1.37995i	0.805	2.32446
-1.70573	±	0.10905i	0.998	1.70922
-1.48464	±	0.60051i	0.927	1.60149
-4.26079e-3	±	6.65864e-3i	0.539	7.90518e-3

Table 7: Reduced Order Observer Recovery, 2 Integral Control

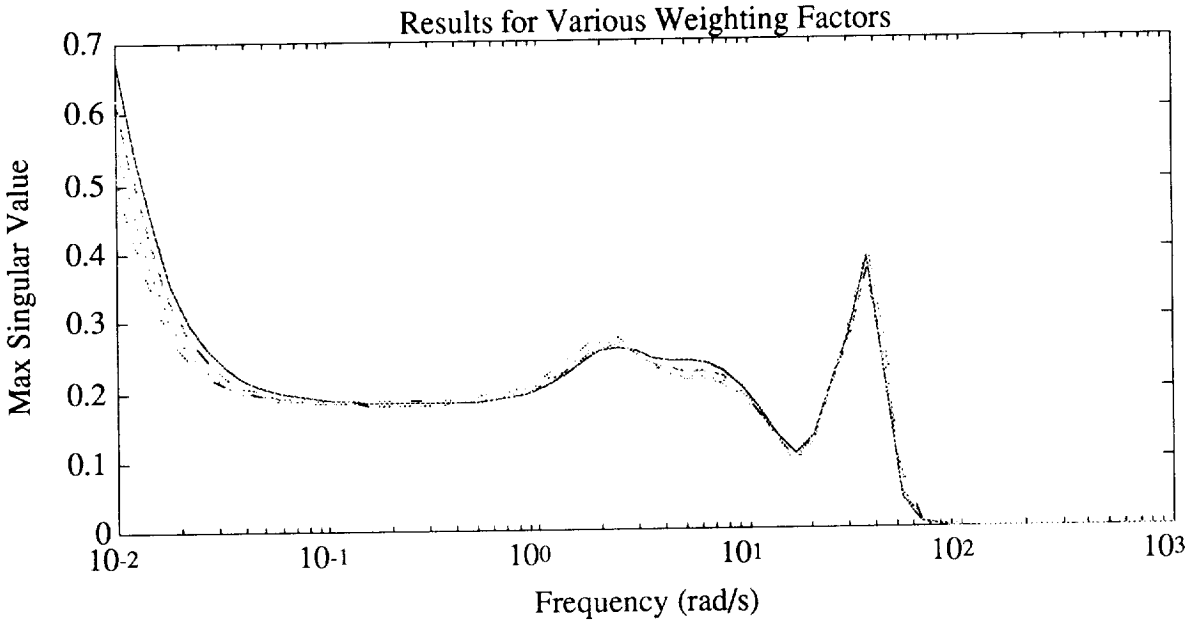


Figure 19: Recovery Error, Reduced Order Observer

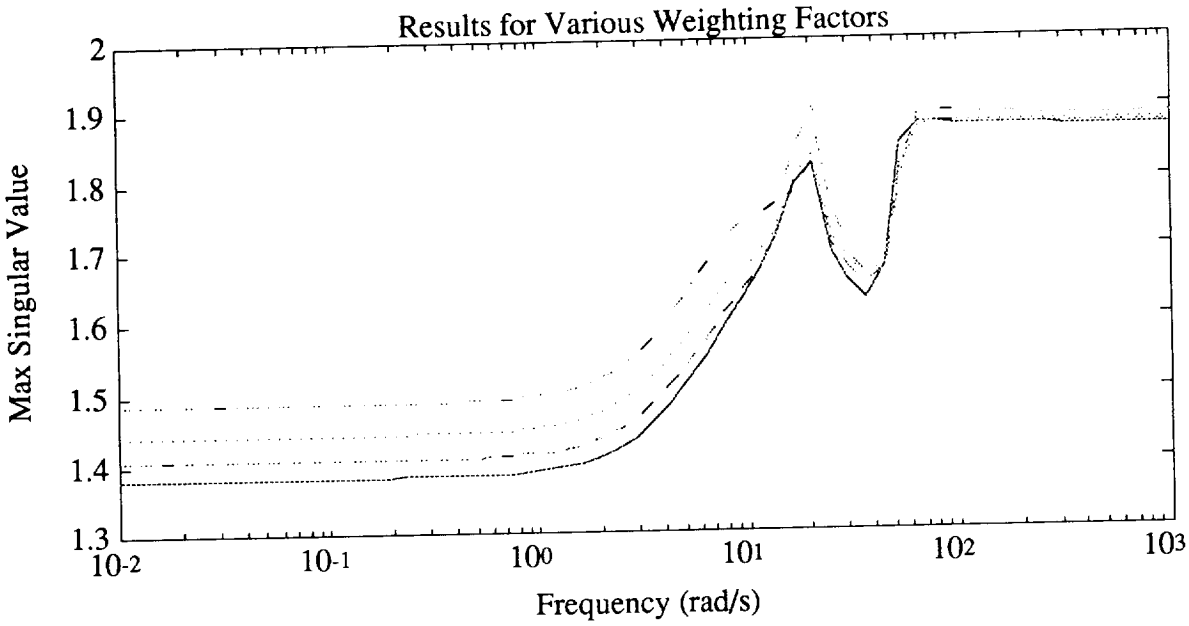


Figure 20: Controller Activity, Reduced Order Observer

Poles			Damping	Freq (rad/sec)
-8.80979	±	52.3960i	0.166	53.1315
-9.38043	±	51.2638i	0.180	52.1149
-16.1812	±	35.1029i	0.419	38.6528
-6.11002	±	37.2574i	0.162	37.7550
-14.0871	±	24.0717i	0.505	27.8907
-8.72514	±	24.6011i	0.334	26.1025
-24.7999	±	0.55088i	1.000	24.8060
-24.5081	±	2.91104i	0.993	24.6804
-20.5883	±	1.42024i	0.998	20.6373
-2.25247	±	20.2789i	0.110	20.4037
-4.20761	±	19.5369i	0.211	19.9849
-19.3829			1.000	19.3829
-5.98385	±	5.24203i	0.752	7.95521
-6.54135	±	0.58733i	0.996	6.56766
-6.51433			1.000	6.51433
-3.75254	±	4.67419i	0.626	5.99413
-4.72891			1.000	4.72891
-4.65048			1.000	4.65048
-1.87052	±	1.37995i	0.805	2.32446
-1.70573	±	0.10905i	0.998	1.70922
-1.48464	±	0.60051i	0.927	1.60149
-4.26079e-3	±	6.65864e-3i	0.539	7.90518e-3

Table 8: Reduced Order Observer Recovery, 2 Integral Control, Weighted

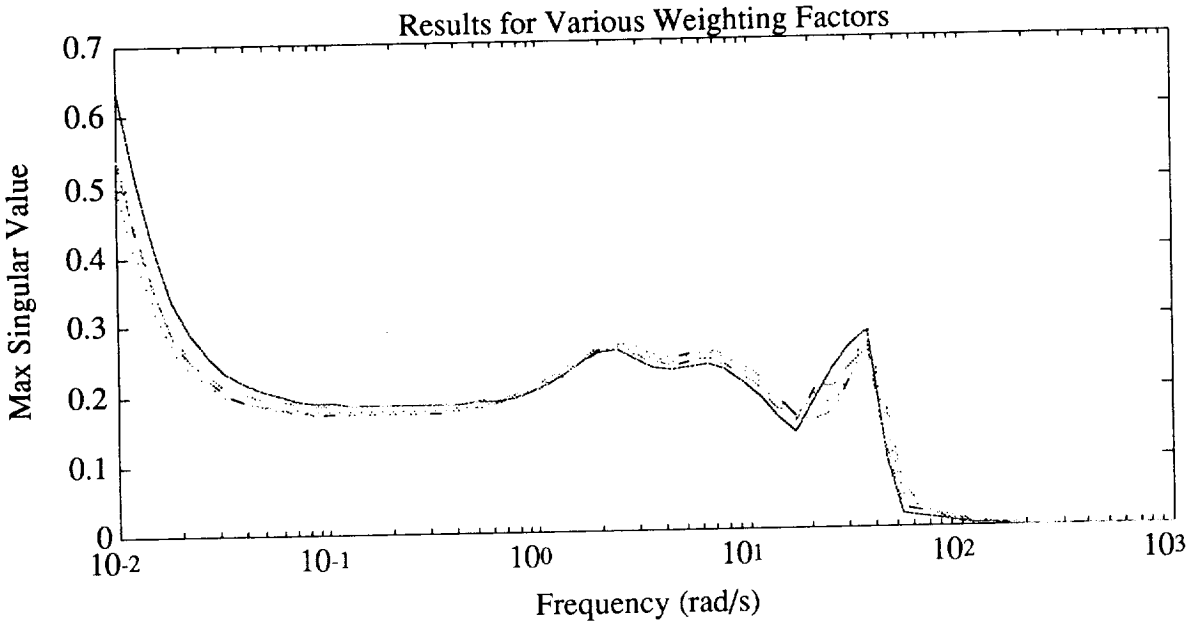


Figure 21: Recovery Error, Shaped Reduced Order Observer

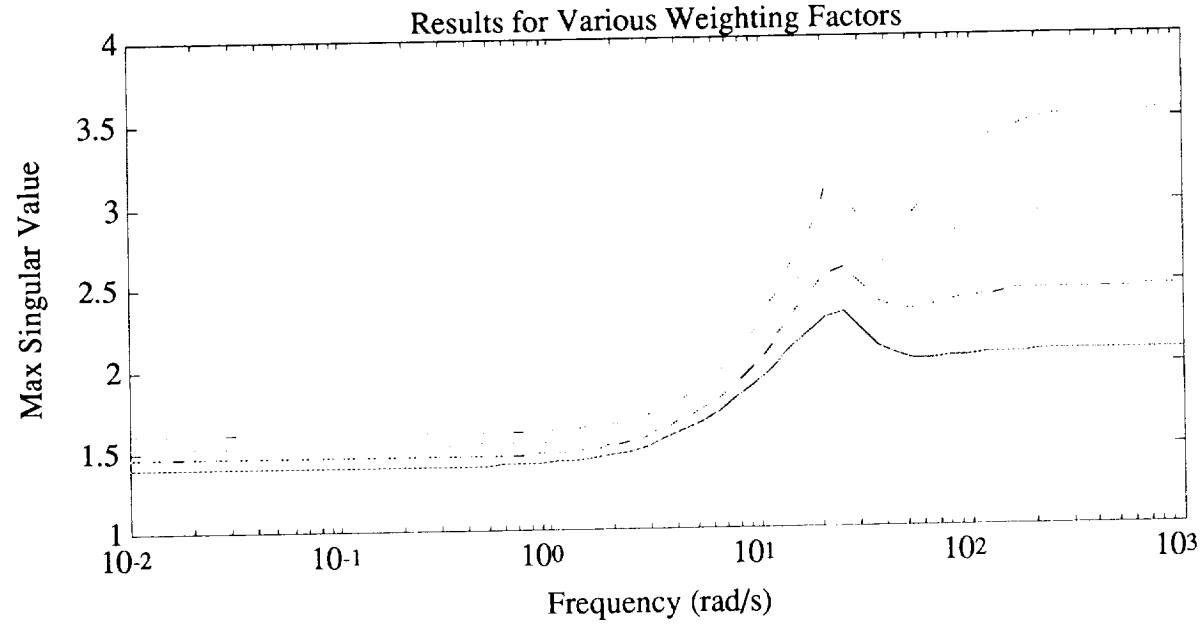


Figure 22: Controller Activity, Shaped Reduced Order Observer

5.3 Numerical LTR I: 8th Order Design

Aside from the structured observers, one can formulate a compensator with dynamics whose states do not necessarily have any analogues with observer states. This direct optimization, run through the program SANDY [1], will optimize a controller design given a general structure for it. In this case, we specify that it be 8th order in a series of 4 uncoupled 2nd order systems. The initial values for the direct feedthrough matrix on this controller correspond to those corresponding columns of the full state feedback matrix. The general control structure is as follows:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{65} & A_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{87} & A_{88} \end{bmatrix};$$

$$B_c = \begin{bmatrix} B_{1p} & B_{1q} & B_{1r} & B_{1u} & B_{1v} & B_{1w} & 0 & 0 & 0 & 0 \\ B_{2p} & B_{2q} & B_{2r} & B_{2u} & B_{2v} & B_{2w} & 0 & 0 & 0 & 0 \\ B_{3p} & B_{3q} & B_{3r} & B_{3u} & B_{3v} & B_{3w} & 0 & 0 & 0 & 0 \\ B_{4p} & B_{4q} & B_{4r} & B_{4u} & B_{4v} & B_{4w} & 0 & 0 & 0 & 0 \\ B_{5p} & B_{5q} & B_{5r} & B_{5u} & B_{5v} & B_{5w} & 0 & 0 & 0 & 0 \\ B_{6p} & B_{6q} & B_{6r} & B_{6u} & B_{6v} & B_{6w} & 0 & 0 & 0 & 0 \\ B_{7p} & B_{7q} & B_{7r} & B_{7u} & B_{7v} & B_{7w} & 0 & 0 & 0 & 0 \\ B_{8p} & B_{8q} & B_{8r} & B_{8u} & B_{8v} & B_{8w} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_c = \begin{bmatrix} C_{\delta_0 1} & C_{\delta_0 2} & C_{\delta_0 3} & C_{\delta_0 4} & C_{\delta_0 5} & C_{\delta_0 6} & C_{\delta_0 7} & C_{\delta_0 8} \\ C_{\delta_c 1} & C_{\delta_c 2} & C_{\delta_c 3} & C_{\delta_c 4} & C_{\delta_c 5} & C_{\delta_c 6} & C_{\delta_c 7} & C_{\delta_c 8} \\ C_{\delta_s 1} & C_{\delta_s 2} & C_{\delta_s 3} & C_{\delta_s 4} & C_{\delta_s 5} & C_{\delta_s 6} & C_{\delta_s 7} & C_{\delta_s 8} \\ C_{\delta_{tr} 1} & C_{\delta_{tr} 2} & C_{\delta_{tr} 3} & C_{\delta_{tr} 4} & C_{\delta_{tr} 5} & C_{\delta_{tr} 6} & C_{\delta_{tr} 7} & C_{\delta_{tr} 8} \end{bmatrix}$$

$$D_c = \begin{bmatrix} K_{\delta_0 p} & K_{\delta_0 q} & K_{\delta_0 r} & K_{\delta_0 u} & K_{\delta_0 v} & K_{\delta_0 w} & K_{\delta_0 \theta} & K_{\delta_0 \phi} & K_{\delta_0 \int \theta} & K_{\delta_0 \int \phi} \\ K_{\delta_c p} & K_{\delta_c q} & K_{\delta_c r} & K_{\delta_c u} & K_{\delta_c v} & K_{\delta_c w} & K_{\delta_c \theta} & K_{\delta_c \phi} & K_{\delta_c \int \theta} & K_{\delta_c \int \phi} \\ K_{\delta_s p} & K_{\delta_s q} & K_{\delta_s r} & K_{\delta_s u} & K_{\delta_s v} & K_{\delta_s w} & K_{\delta_s \theta} & K_{\delta_s \phi} & K_{\delta_s \int \theta} & K_{\delta_s \int \phi} \\ K_{\delta_{tr} p} & K_{\delta_{tr} q} & K_{\delta_{tr} r} & K_{\delta_{tr} u} & K_{\delta_{tr} v} & K_{\delta_{tr} w} & K_{\delta_{tr} \theta} & K_{\delta_{tr} \phi} & K_{\delta_{tr} \int \theta} & K_{\delta_{tr} \int \phi} \end{bmatrix}$$

Of course, the values marked with a 0 remain 0. The rest are allowed to optimize including the state feedback gains noted in the D_c matrix. This still amounts to a hefty number of parameters: 104 of them. It turns out that the feedthrough gains associated with states that turn out to be directly measured (i.e., pitch, roll, etc.) have a significant impact on the recovery. Optimization of these gains from their state feedback values provides for a significant amount of the recovery. Thus, an analogy from the Ricatti-based solutions does not hold here—we are indeed changing the supposed state feedback gain matrix K. The recovery error is 0.16374, which is the best recovery of all systems. This is remarkable, considering the simple dynamics structure. Later on one will see the limits to optimizing the direct feedthrough gains associated with just the observed terms. This optimization perhaps could be improved upon as some analysis indicates that some further improvement could be made, however some current algorithmic difficulties have prevented it. The full state feedforward gains were used, as attempts to structure around their use did not seem to work (see figure 23). It would have seemed that recovery error could have been improved by the additional degrees of freedom, but the results seemed to indicate otherwise. This optimization

Zero			Damping	Freq (Hz)
-9.37520	±	49.7002i	0.185	50.5767
-47.5601			1.000	47.5601
-6.58311	±	37.7641i	0.172	38.3336
-10.4636	±	27.2017i	0.359	29.1448
-20.6436	±	12.9692i	0.847	24.3795
-23.4432	±	2.81882i	0.993	23.6121
-3.74904	±	19.5247i	0.189	19.8814
-11.2690	±	11.8710i	0.688	16.3680
-15.9453			1.000	15.9453
-4.30385	±	6.15915i	0.573	7.51387
-5.44168	±	4.74486i	0.754	7.21980
-6.95416	±	1.78549i	0.969	7.17971
-5.61014			1.000	5.61014
-2.05374	±	1.28164i	0.848	2.42084
-1.63643	±	0.73012i	0.913	1.79192
-1.50904	±	0.12381i	0.997	1.51411
-4.25281e-003	±	6.68477e-003i	0.537	7.92292e-003

Table 9: Poles of 8 State LTR Optimized Controller and Plant

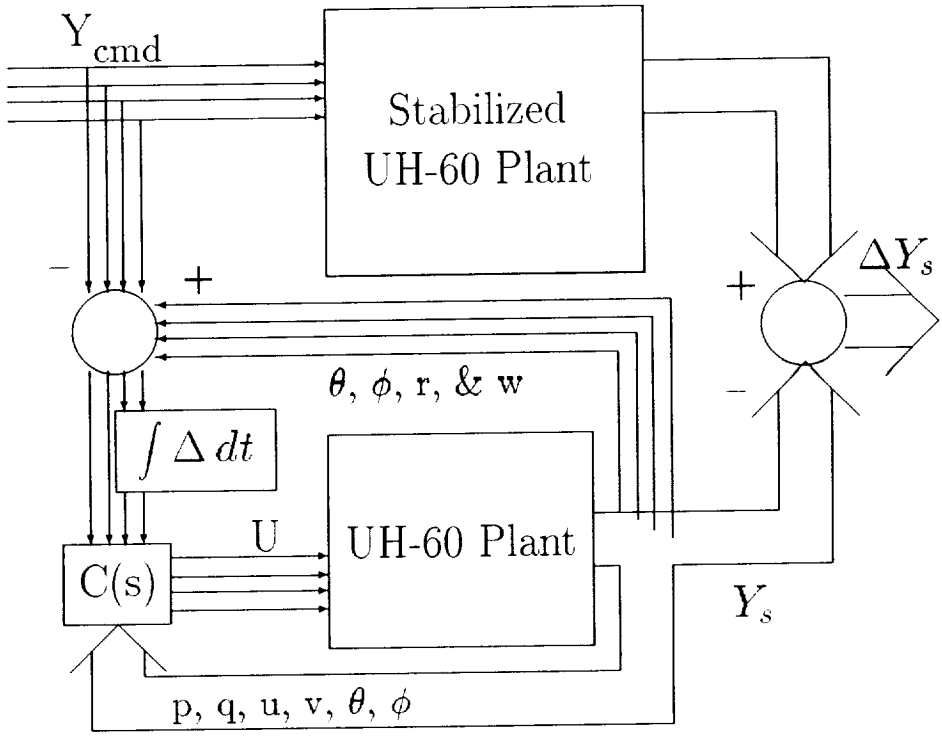


Figure 23: LTR for Output Feedback Controller C(s), No $K_f f$ (not used)

proceeded with filtered disturbance inputs. The pitch and roll command disturbance inputs were integrated, then passed through a filter with a time constant of 8 rad/sec. The vertical rate and yaw rate command inputs were passed through a filter with a time constant of 4 rad/sec. These filters were also applied in the subsequent **SANDY** based LTR optimizations.

There is no control penalty **R** on an LTR-based direct optimization. Since it stands to reason that one is trying to recover a state feedback design, one needs to be able to freely vary the necessary gains. Ricatti-based LTR theory indicates a need for some gains to go to infinity, Fortunately most optimization schemes will detect an extreme lack of sensitivity of the cost function value to a parameter change: the parameter will probably never move very far before the optimization completes and terminates. Such has been the observed behavior here.

The disturbance response penalty affected the mean-squared values. Note that here, as opposed to the closed form Ricatti solutions, one could minimize the actual error rather than a matrix analogous to some error multiplier matrix *M(s)*. The value for the response penalty matrix Q, for all direct LTR optimizations was:

$$Q = diag\{0.1, 0.05, 2, 1, 1, 2, 1, 0.2, 1, 0.2\}$$

The entries corresponded to the outputs {p, q, r, u, v, w, ϕ, θ, $\int\theta$, and $\int\phi$ }.

5.4 Numerical LTR II: 4th Order Design

How much worse do the results get if the dynamics in the controller is lowered to 4th order? In a word, not much worse—a recovery error of 0.17105. With the overall structure otherwise the same, an interaction term between the pair of second order systems was allowed. This was intended to

Zero			Damping	Freq (Hz)
-464.184			1.000	464.184
-8.95655	±	51.1120i	0.173	51.8908
-6.90773	±	37.0939i	0.183	37.7316
-7.07229	±	25.5506i	0.267	26.5114
-24.1159	±	2.94300i	0.993	24.2948
-4.51612	±	19.8662i	0.222	20.3731
-20.2468			1.000	20.2468
-5.75452	±	5.30307i	0.735	7.82541
-2.82475	±	5.70997i	0.443	6.37048
-4.12753	±	5.42571i	0.605	6.81724
-6.75926			1.000	6.75926
-2.12814	±	2.73388i	0.614	3.46455
-1.38437	±	1.13227i	0.774	1.78844
-1.72118			1.000	1.72118
-1.57803	±	0.61632i	0.931	1.69411
-1.12528			1.000	1.12528
-4.2532e-003	±	6.68453e-003i	0.537	7.92290e-003

Table 10: Poles of 4 State LTR Optimized Controller and Plant

make the dynamics a little richer to see if this would allow results nearly as good as those for the 8th order system. Because the size of this controller was minimal it was felt that some extra effort needed to be made to see if this controller would achieve a level as good as any other system. These

interaction terms–from the output of one second order system to the input of the other made the design of the controller matrix look like:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A_{21} & A_{22} & \underline{A_{23}} & 0 \\ 0 & 0 & 0 & 1 \\ \underline{A_{41}} & 0 & A_{43} & A_{44} \end{bmatrix}$$

Since the cost (needing to optimize only 2 more variables) was low, this was done.

5.5 Numerical LTR III: 0th Order Design

Finally, one asks, is there a need for dynamics at all? What about feeding the output directly back? This time the direct feedthough matrix is all there is, so one will begin with using the state feedback gains and try to optimize it further. Analyzing results with just state feedback gains applied to the measured output, we see what one can expect in the least. The recovery error is 6.1807, with the accompanying suite of system eigenvalues as seen in table 11. While the performance is not horrible, it does not keep the response within any sort of acceptable tolerance. The eigenvalues of the system (Table 11) still indicate a well stabilized system. If one optimizes this D matrix via

Zero			Damping	Freq (Hz)
-9.49008	±	51.5146i	0.181	52.3815
-5.48053	±	33.7356i	0.160	34.1778
-7.55594	±	24.5267i	0.294	25.6642
-24.3200	±	2.76144i	0.994	24.4763
-4.68312	±	20.9348i	0.218	21.4522
-19.7090			1.000	19.7090
-10.4723			1.000	10.4723
-5.46663	±	5.15352i	0.728	7.51284
-3.40089	±	5.46878i	0.528	6.44000
-2.20360	±	2.52614i	0.657	3.35220
-1.80756			1.000	1.80756
-1.47456	±	0.83193i	0.871	1.69305
-1.34444	±	0.50772i	0.936	1.43711
-4.10885-003	±	6.72672e-003i	0.521	1.25451e-003

Table 11: Poles of UH-60 Stabilized With Selected State Feedback Gains

SANDY, the recovery error improves dramatically to 1.2489. While this is still not as good as the optimized LTR designs with dynamics, it is not so far off of a Luenberger LTR design. Technically there are dynamics even with this design in the form of the integrators.

Looking at the plot of recovery error maximum singular values (figure 24), one will see that the error at lower frequency is actually better for the 0th order controller. It is believed that this is a function of how well the optimization did. Because there were many fewer parameters to move, the relative strength of the optimization was probably better. Hopefully future work will improve technique to allow opitmization to the same degree on the 8th and 4th order designs. Because of the lack of internal dynamics, the error was worse than the 8th and 4th order designs for frequencies in the 1-10 rad/sec range.

Zero			Damping	Freq (Hz)
-9.78679	±	51.4210i	0.187	8.33081
-2.09033	±	35.1811i	0.059	5.60912
-7.35643	±	24.4792i	0.288	4.06810
-24.2160	±	3.31615i	0.991	3.89007
-5.59656	±	19.3922i	0.277	3.21233
-20.0660			1.000	3.19360
-3.54972	±	6.16242i	0.499	1.13186
-5.86258	±	5.18050i	0.749	1.24515
-6.71038			1.000	1.06799
-2.87716	±	4.32472i	0.554	0.82671
-1.15676	±	0.77973i	0.829	0.22202
-1.10283	±	0.67672i	0.852	0.20593
-1.70616			1.000	0.27154
-4.10885-003	±	6.72672e-003i	0.521	1.25451e-003

Table 12: Poles of 0 State LTR Optimized Controller and Plant

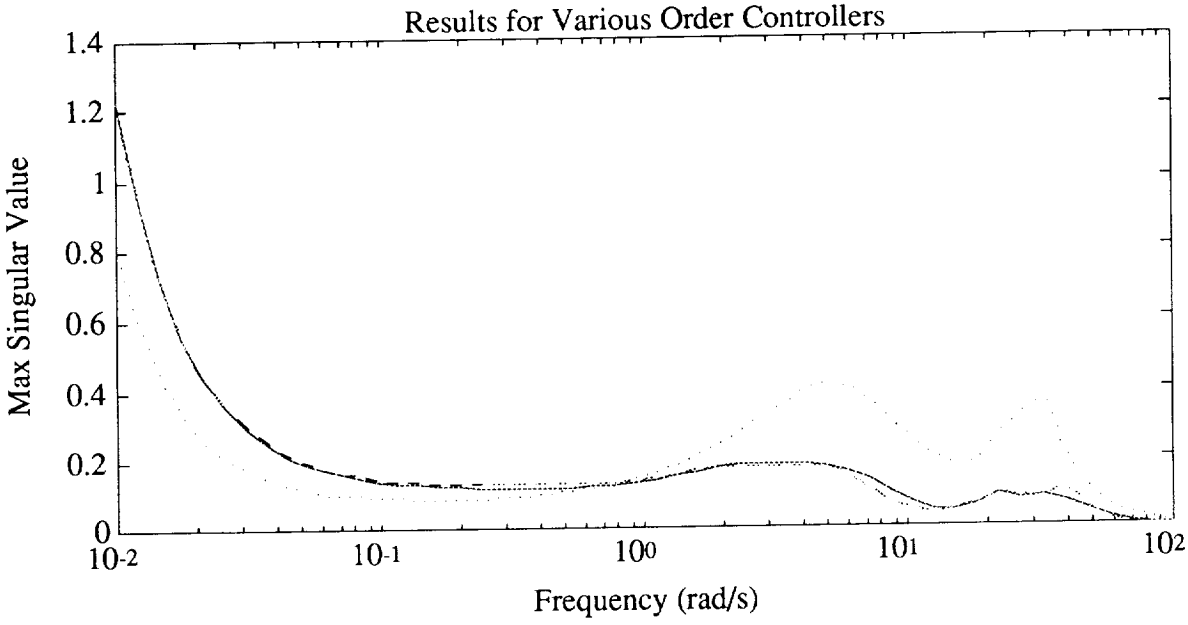


Figure 24: Recovery Error, 8th, 4th and 0th Order

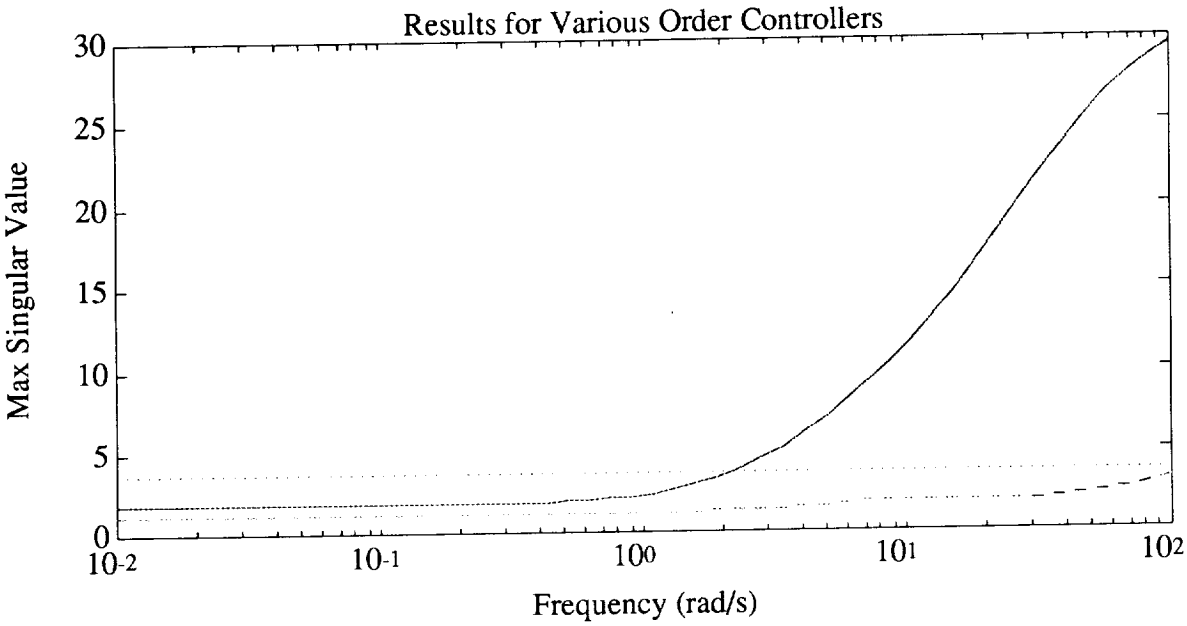


Figure 25: Controller Activity, 8th, 4th and 0th Order

6 Direct Optimization for a 2 Integral State System Controller

What is the difference between LTR and a direct optimization? Formwise, the compensator structure is identical to that of the 4th order LTR design:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A_{21} & A_{22} & \underline{A_{23}} & 0 \\ 0 & 0 & 0 & 1 \\ \underline{A_{41}} & 0 & A_{43} & A_{44} \end{bmatrix}$$

$$B_c = \begin{bmatrix} B_{1p} & B_{1q} & B_{1r} & B_{1u} & B_{1v} & B_{1w} & 0 & 0 & 0 & 0 \\ B_{2p} & B_{2q} & B_{2r} & B_{2u} & B_{2v} & B_{2w} & 0 & 0 & 0 & 0 \\ B_{3p} & B_{3q} & B_{3r} & B_{3u} & B_{3v} & B_{3w} & 0 & 0 & 0 & 0 \\ B_{4p} & B_{4q} & B_{4r} & B_{4u} & B_{4v} & B_{4w} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_c = \begin{bmatrix} C_{\delta_0 1} & C_{\delta_0 2} & C_{\delta_0 3} & C_{\delta_0 4} \\ C_{\delta_c 1} & C_{\delta_c 2} & C_{\delta_c 3} & C_{\delta_c 4} \\ C_{\delta_s 1} & C_{\delta_s 2} & C_{\delta_s 3} & C_{\delta_s 4} \\ C_{\delta_{tr} 1} & C_{\delta_{tr} 2} & C_{\delta_{tr} 3} & C_{\delta_{tr} 4} \end{bmatrix}$$

$$D_c = \begin{bmatrix} K_{\delta_0 p} & K_{\delta_0 q} & K_{\delta_0 r} & K_{\delta_0 u} & K_{\delta_0 v} & K_{\delta_0 w} & K_{\delta_0 \theta} & K_{\delta_0 \phi} & K_{\delta_0 \int \theta} & K_{\delta_0 \int \phi} \\ K_{\delta_c p} & K_{\delta_c q} & K_{\delta_c r} & K_{\delta_c u} & K_{\delta_c v} & K_{\delta_c w} & K_{\delta_c \theta} & K_{\delta_c \phi} & K_{\delta_c \int \theta} & K_{\delta_c \int \phi} \\ K_{\delta_s p} & K_{\delta_s q} & K_{\delta_s r} & K_{\delta_s u} & K_{\delta_s v} & K_{\delta_s w} & K_{\delta_s \theta} & K_{\delta_s \phi} & K_{\delta_s \int \theta} & K_{\delta_s \int \phi} \\ K_{\delta_{tr} p} & K_{\delta_{tr} q} & K_{\delta_{tr} r} & K_{\delta_{tr} u} & K_{\delta_{tr} v} & K_{\delta_{tr} w} & K_{\delta_{tr} \theta} & K_{\delta_{tr} \phi} & K_{\delta_{tr} \int \theta} & K_{\delta_{tr} \int \phi} \end{bmatrix}$$

The disturbance inputs, while filtered as before, will influence the dynamics in a significantly different way. The feedforward matrix normally used is based on the full state feedback design—these are the gains used to translate a vertical rate command into commands on the actuators. With this design, the yaw rate and vertical rate disturbances are summed into the signals going from the

sensors to the controller itself—no "recovery" to a set of full state feedback gains is implied (see figure 26). The pitch and roll commands, as before, go into the integrators (no direct feedthrough

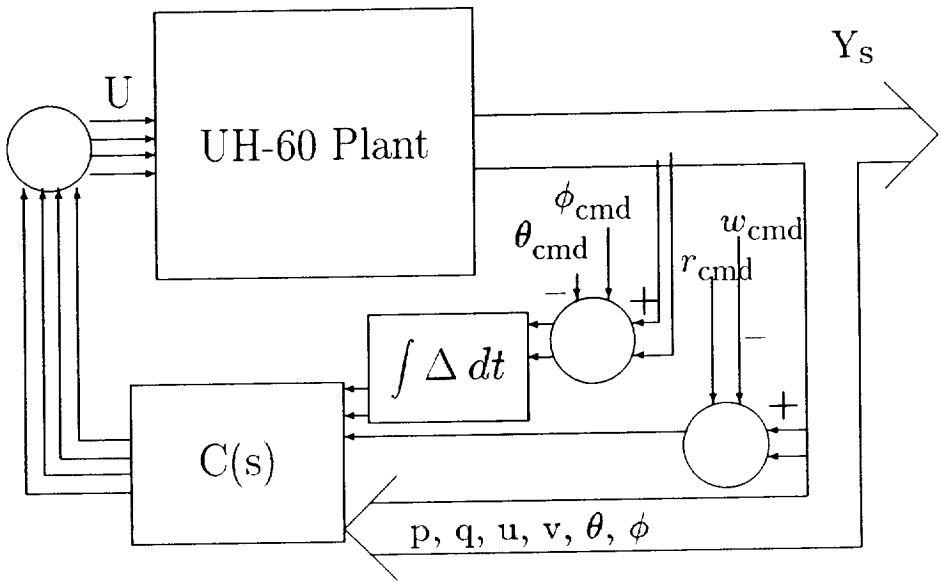


Figure 26: Direct Optimization for Output Feedback Controller C(s)

gains around these integrators). Because there is no direct comparison with the original state feedback design feedforward, one cannot look at residuals between the two systems in a meaningful way. The overall recovery process appears somewhat sensitive to the feedforward gains. One can, however, look at the basic responses of this closed loop system and judge how well the optimization went. In fact, because there exists a 4th order LTR design with the same structure a comparison to it will be useful.

The largest source of difficulty was in choosing the relative weighting of control versus criterion. Classically, one worked for the fastest adequate response time versus having unreasonable gains or a large overshoot. The weighting matrices had values as shown in tables 13 and 14. Considering

Q for State Feedback	Q for Optimized Output Feedback	Factor Weighted
1	10	Blend of: $0.25q + \theta + \int \theta\, dt$
0.1	6	Blend of: $0.2p + \phi + \int \phi\, dt$
1.6	16	Yaw Rate r
2	30	Vertical Rate w

Table 13: Diagonal Values of Q and Corresponding Factors

how this design had some high frequency components and overshoot, the difference in weights from the LQ to the direct design is significant. The resulting eigenvalues are shown in Table 15. From previous experience, one can see that the eigenvalues do not say much by themselves. Looking at

R for State Feedback	R for Optimized Output Feedback	Factor Weighted
10	0.4	Collective δ_0
1	1	Sine δ_s
1	1	Cosine δ_c
1	1	Tail Rotor δ_{TR}

Table 14: Diagonal Values of R and Corresponding Factors

Zero			Damping	Freq (Hz)
-470.046			1.000	470.046
-115.173			1.000	115.173
-9.39606	±	51.4281i	0.180	52.2794
-2.85548	±	35.6182i	0.080	35.7325
-30.9068			1.000	30.9068
-8.59788	±	26.2384i	0.311	27.6112
-24.2882	±	3.56569i	0.989	24.5486
-5.44497	±	19.9205i	0.264	20.6512
-17.0312			1.000	17.0312
-1.23462	±	8.68975i	0.141	8.77702
-3.91763	±	6.12717i	0.539	7.27255
-2.88111	±	3.14284i	0.676	4.26359
-3.07029	±	1.37917i	0.912	3.36583
-3.74960			1.000	3.74960
-1.34781	±	0.58055i	0.918	1.46753
-1.46438			1.000	1.46438
-0.33784			1.000	0.33784
-3.96107e-003	±	6.69608e-003i	0.509	7.77995e-003

Table 15: Poles of 4 State Direct Optimized Controller and Plant

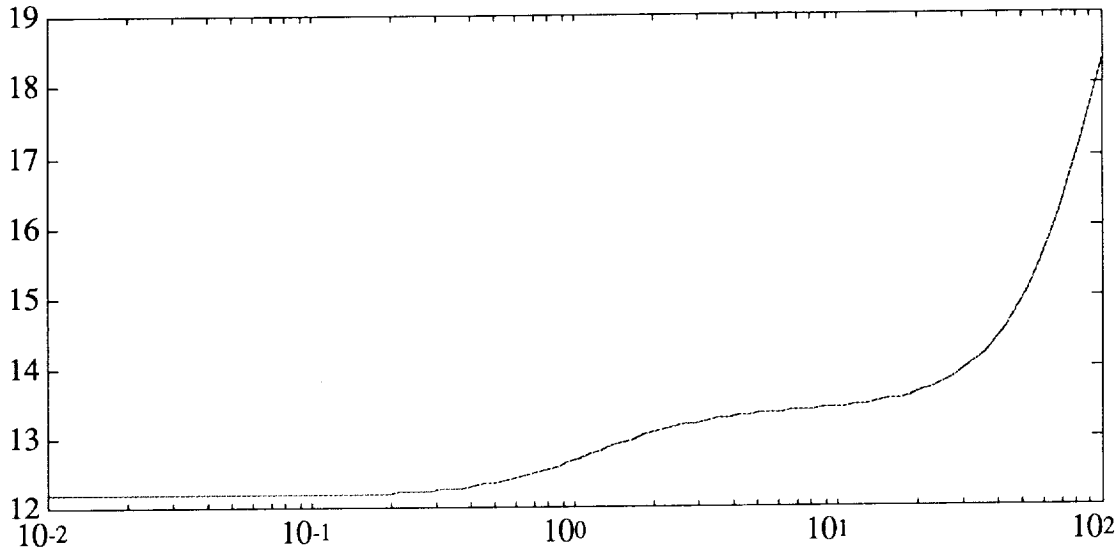


Figure 27: Controller Activity, non-LTR Optimization

the controller activity (see Figure 27), one can see that it starts out with a fairly large gain at low frequencies and increases a modest *relative* amount as one goes up in frequency. Considering the command response plots (figures 28 to 31, this extra controller activity is apparent in the faster response times (and overshoots) for pitch and roll along with the appearance of high frequency components in the responses. Other than this, there seem to be other differences with the LTR design, in that the direct optimization performed better in keeping the yaw rate and up rate near zero in the pitch and roll commands, as well as mitigating actuator use in the yaw rate and up rate commands. Note that while the rise time of the direct optimization response for the pitch and roll commands was better, the delay time was the same.

The Bode plots do not add much to the insights already learned aside from reinforcing the fact that the rolloff is lower on the direct design. Because the phase also moves more slowly the difference in robustness may only be slight.

7 Conclusions

Though one could directly optimize a controller, the LTR procedure is in itself useful combined with the numerical optimization as the process derives a best case controller (versus full-state feedback) as well as a feel for a particular plant model behavior. Though Q and R will almost always be dealt with as diagonal matrices, developing the weightings for a proper controller design is one of the more difficult parts of the design—this development is sped up considerably given the availability of a full state design with which to explore the various effects.

Recovering to a design (though it be to an unrealizable full state one in general) is well defined within a numerical LTR. Though the weighting matrix on the controls is zero, the gains do not tend to blow up (an optimization process would be hard pressed to do this). Having a full state design to recover to (as opposed to a direct optimization to the best possible controller) represents

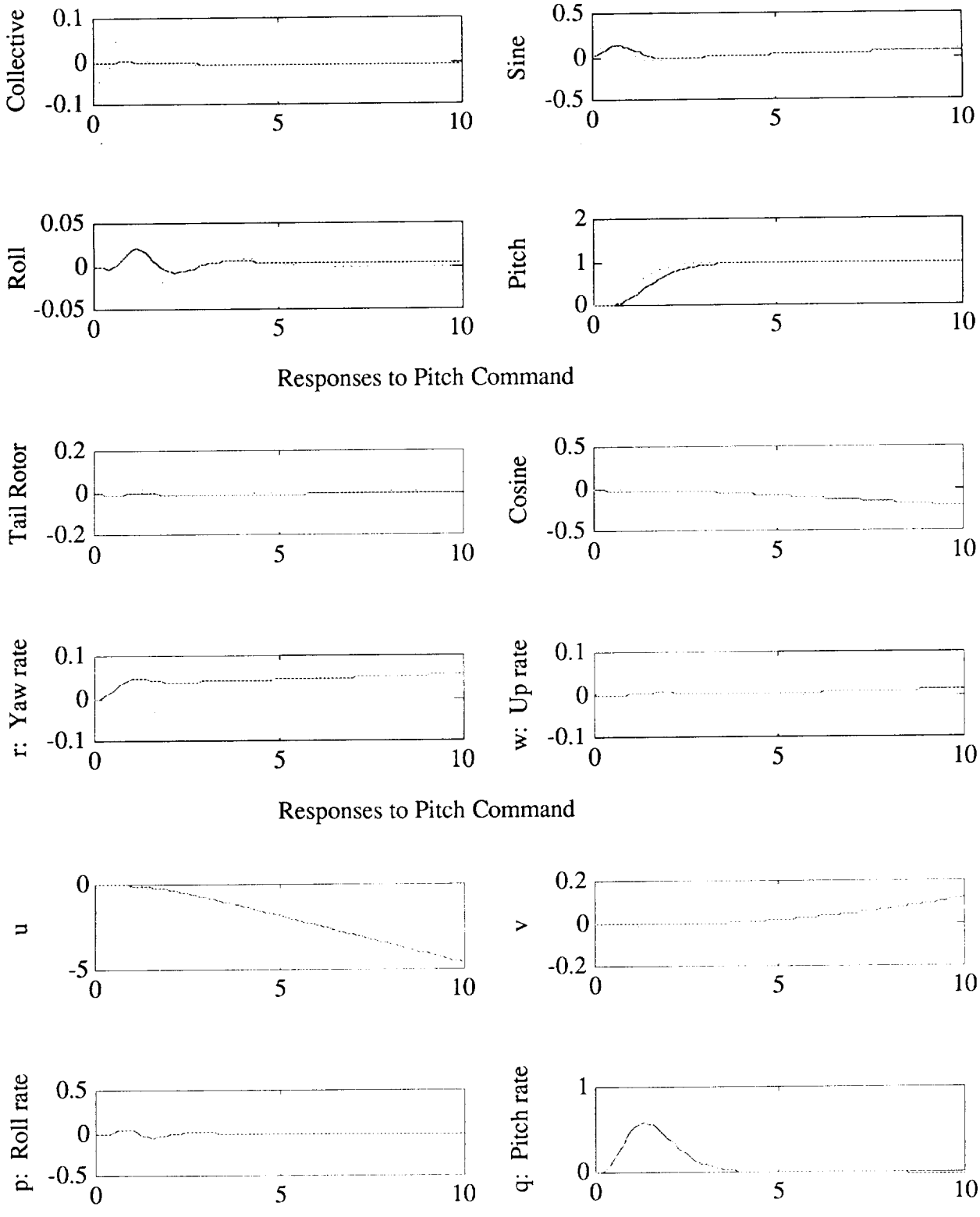


Figure 28: LTR and Direct Comparison

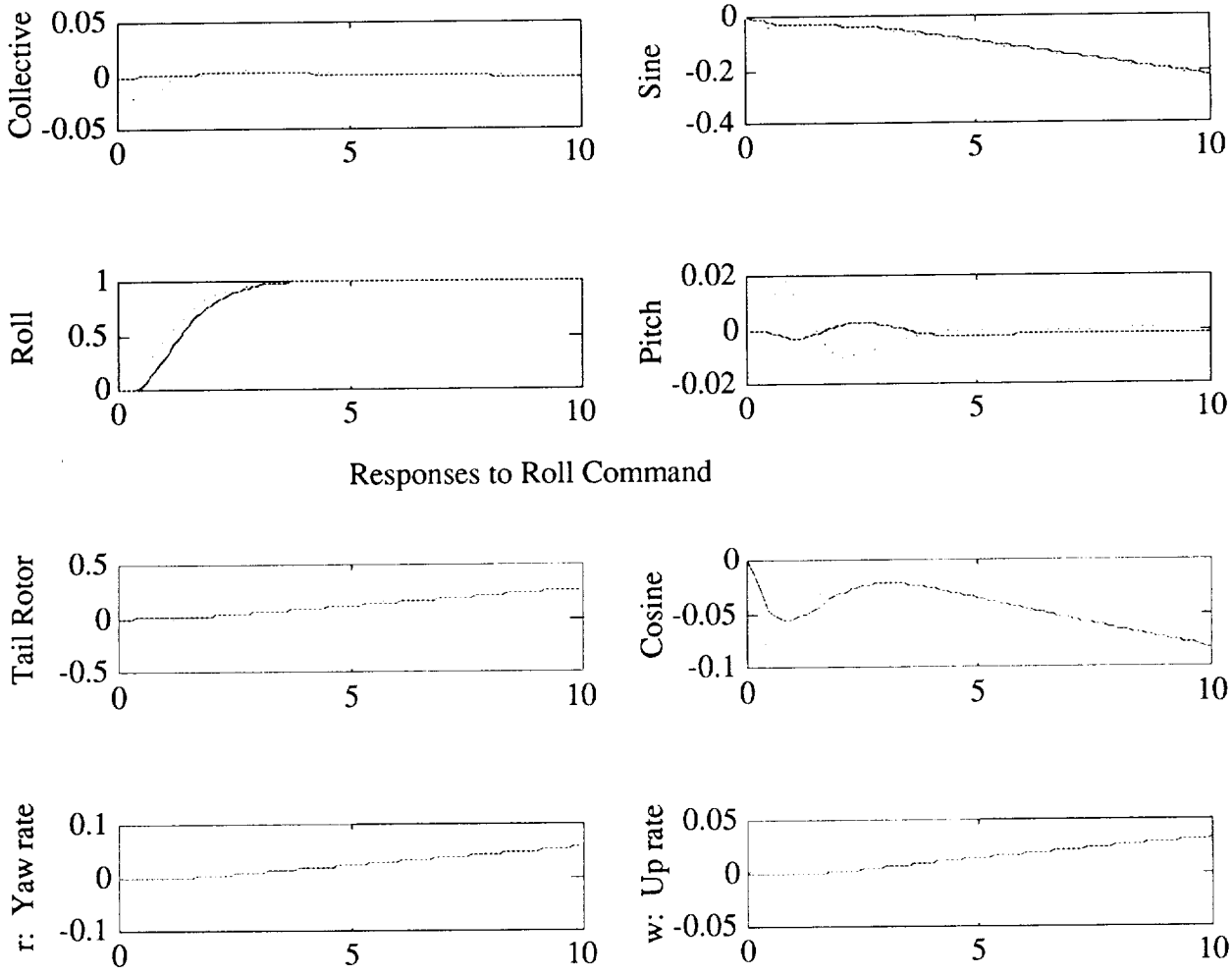
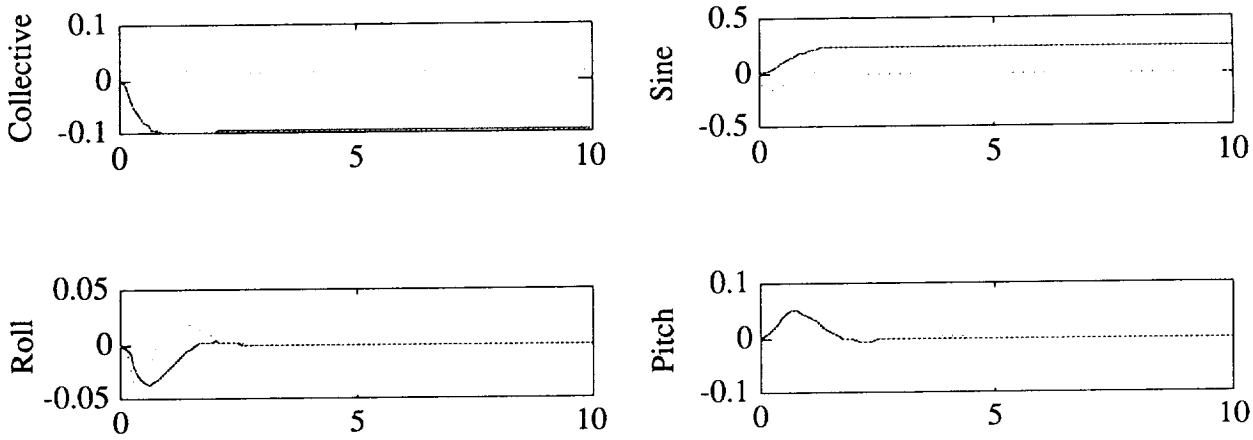
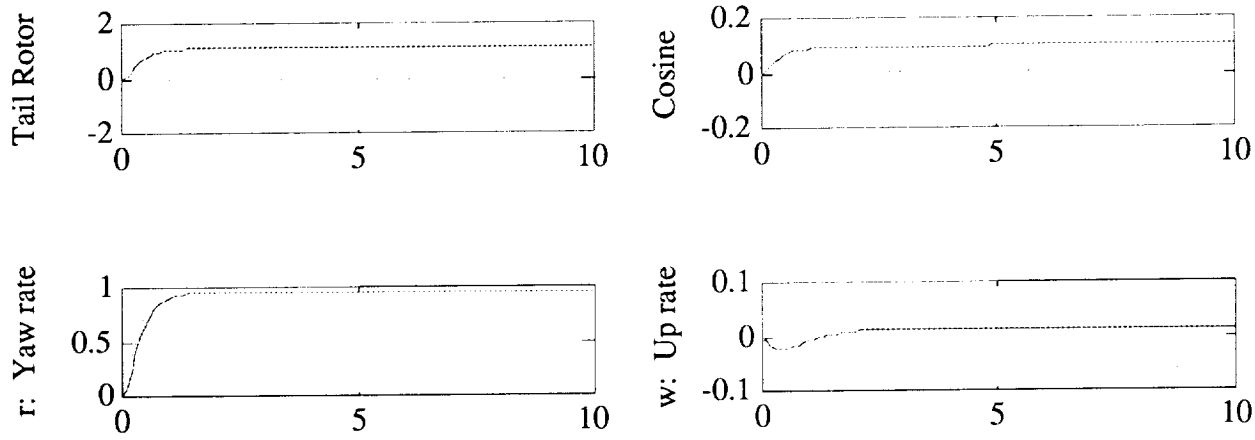


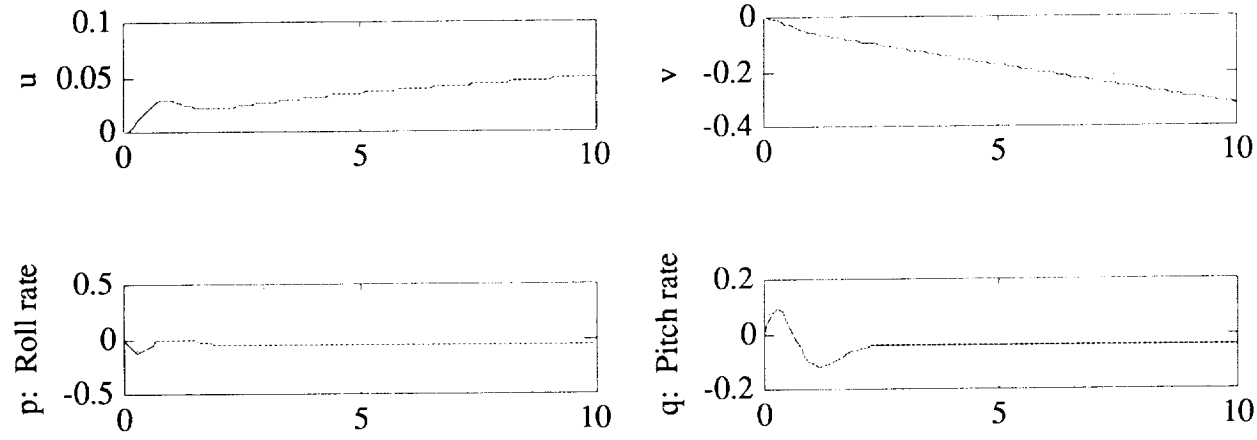
Figure 29: LTR and Direct Comparison



Responses to Yaw Rate Command



Responses to Yaw Rate Command



Responses to Yaw Rate Command

Figure 30: LTR and Direct Comparison

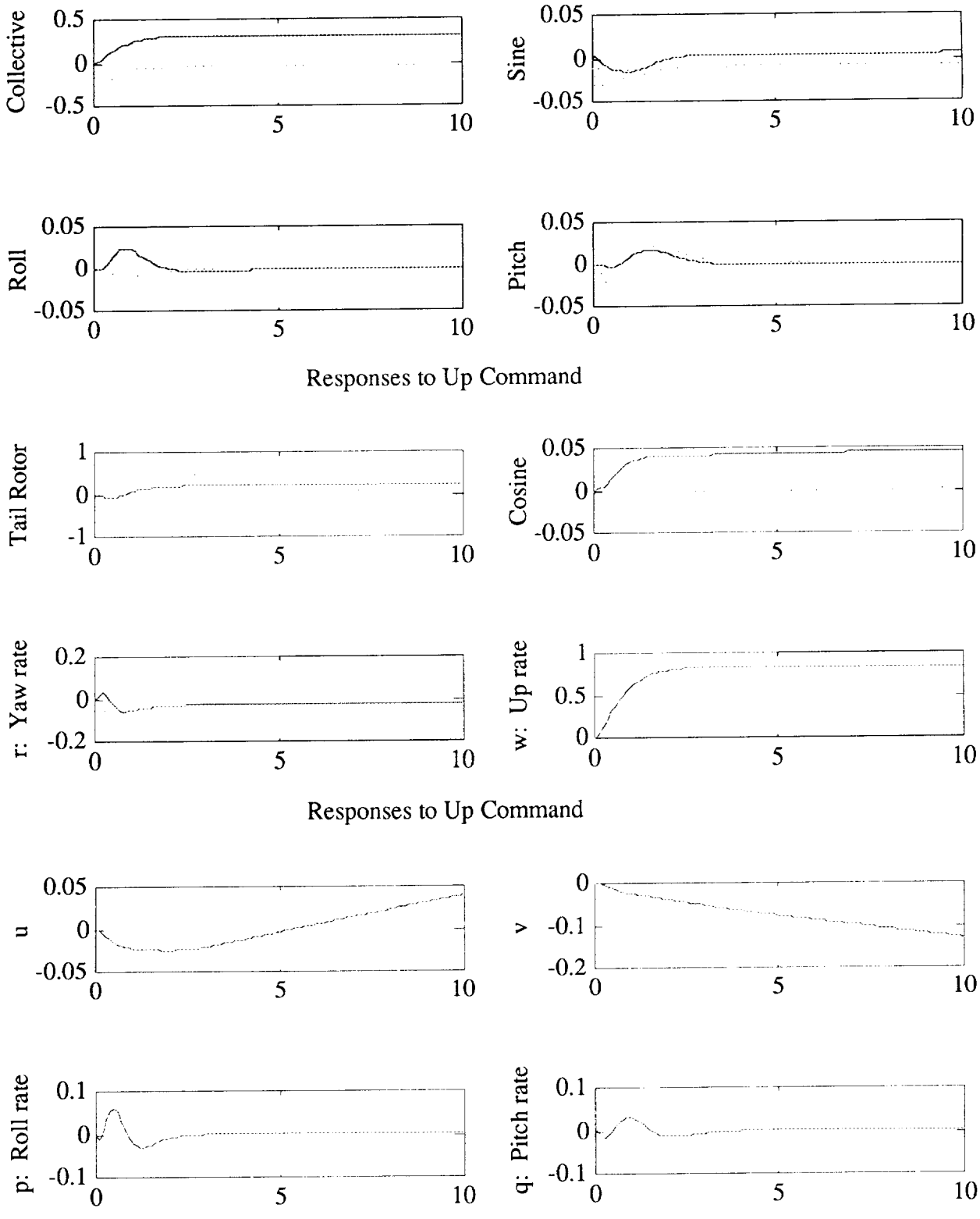


Figure 31: LTR and Direct Comparison

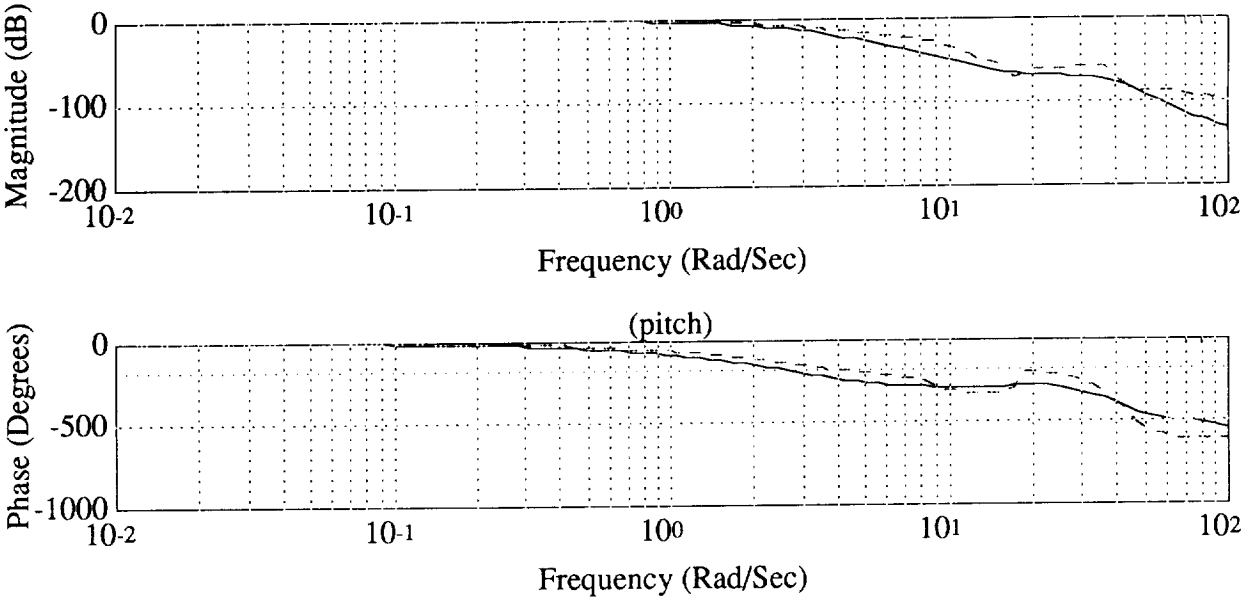


Figure 32: Pitch/Pitch Command Bode Plot, LTR and Direct

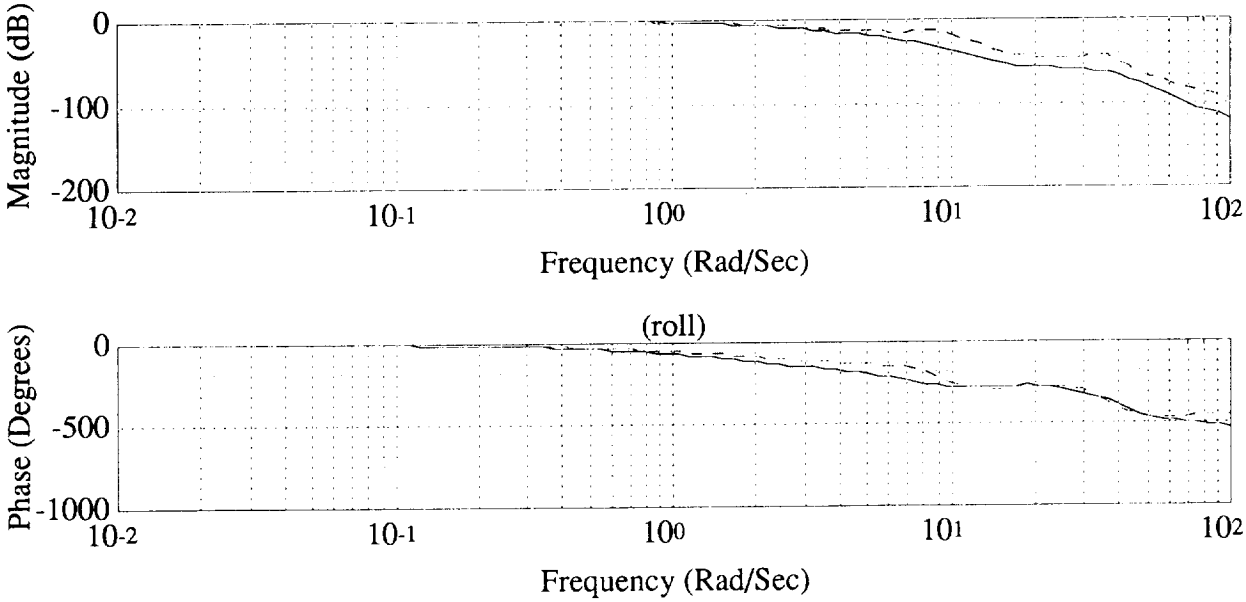


Figure 33: Roll/Roll Command Bode Plot, LTR and Direct

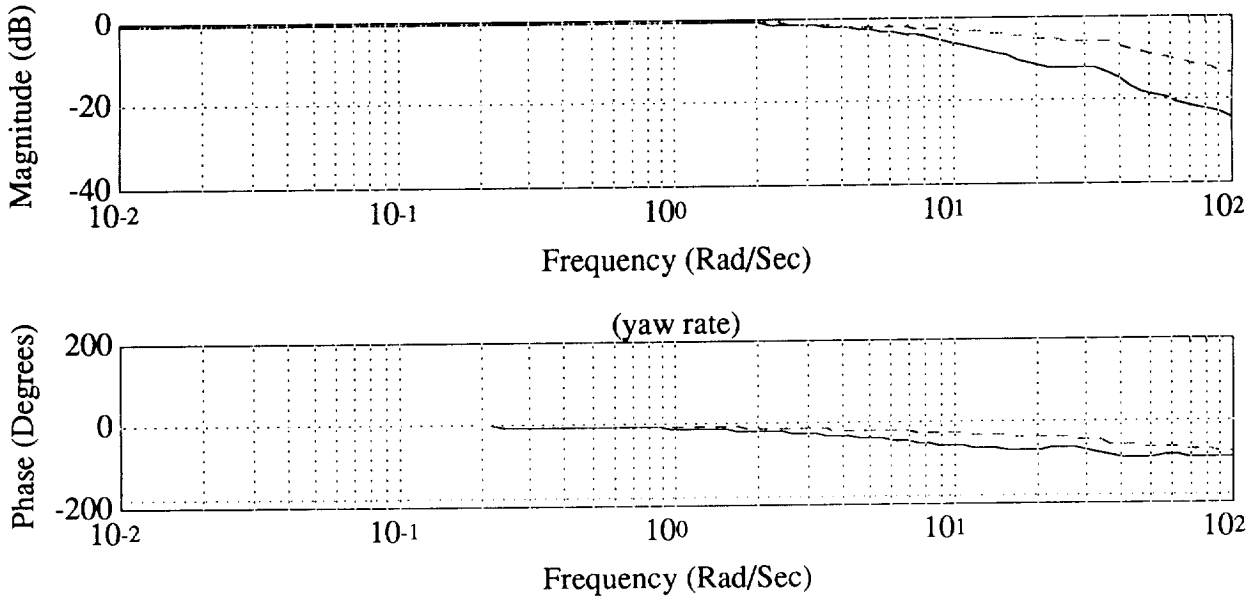


Figure 34: Yaw/Yaw Commanded Rate Bode Plot, LTR and Direct

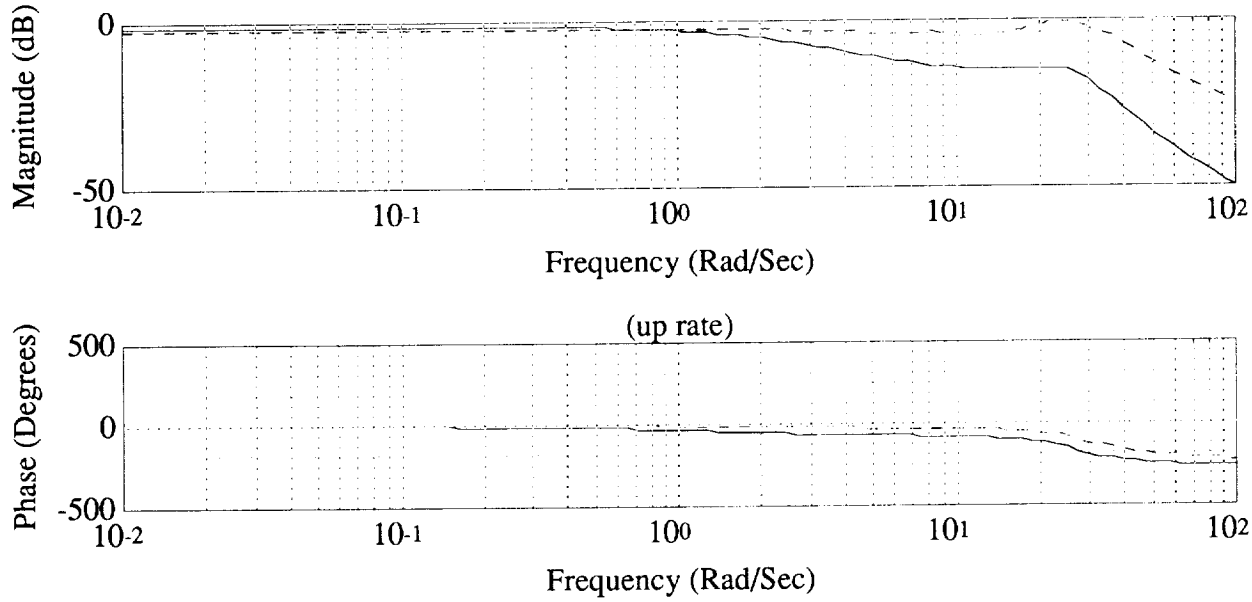


Figure 35: Up/Up Commanded Rate Bode Plot, LTR and Direct

METHOD	ERROR
Full State Observer (Unweighted)	0.4359
Full State Observer (Weighted)	0.2746
Reduced Order Observer (Unweighted)	0.9007
Reduced Order Observer (Weighted)	0.6915
8th Order Numerical LTR	0.1637
4th Order Numerical LTR	0.1711
0th Order Numerical LTR	1.2489

Table 16: Overall List of Recovery Error versus Method

a continual design check—a sort of quality control. This is necessary as one may need to restart the optimizer to avoid some local minimum. In addition the LTR numerical solution may provide a good starting point for the direct optimization.

Using the appropriate entries of the full state feedback gain matrix for the feedforward from the yaw rate command and up rate command to the actuators may be restrictive enough to affect the resulting controller. It appears that LTR does not work if these commands were inserted between the sensor output and the controller input as in the direct optimization (see Figure 26). Fortunately this is not an issue with the pitch and roll commands as the feed into the error integrators is the same in both LTR and the direct case.

What would be the advantages and disadvantages of integral control on r and w? The process of recovery may actually be made easier; however, the full state feedback design would probably not have as good a response time as with a non-integral design. This sort of tradeoff may have to be made in making the decision as to what kind of mission one intends with this helicopter.

Other future exploration should include a test of the robustness of this controller to deviations about the hover flight condition. Ideally the gain scheduling from flight condition to flight condition should result in a fairly even performance across the envelope–especially at a midpoint between flight conditions.

Further improvements to the numerical algorithms to combine robustness to defective matrices and the speed of the diagonal algorithms is also pending. Routine use of a robust form of optimization gradient calculation may improve the numerical results seen in this paper. The LTR process tries to recover a state feedback design with an output feedback design—therefore it is inevitable that various eigenvalues, one set from the closed loop state feedback design, one set from the recovery design, will overlap. Chances are this overlap will result in a defective set of eigenvalues. Currently this roadblock is overcome only by going to the much slower robust form.

References

[1] Ly, U. *A Design Algorithm for Robust Low-Order Controllers*, PhD. Thesis, Department of Aeronautics and Astronautics, Stanford University, November, 1982.

[2] Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H., *User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming*. Technical Report SOL-86-2, Stanford university, January 1986.

[3] P. Sannuti, A. Saberi, "A Special Coordinate Basis of Multivariable Linear Systems–Finite and Infinite Zero Structure, Squaring Down, and Decoupling", *Int. J. Contr.*, Vol. 45, No. 5, pp. 1655-1704, 1987.

[4] B.M. Chen, A. Saberi, and U. Ly, "Closed Loop Transfer Recovery with Observer-Based Controllers—Part 1: Analysis".

[5] B.M. Chen, A. Saberi, and U. Ly, "Closed Loop Transfer Recovery with Observer-Based Controllers—Part 2: Design".