A Parameter Optimization Approach to Multiple-Objective Controller Design

Anton H. J. de Ruiter, *Member, IEEE* and Hugh H. T. Liu, *Member, IEEE*

***Abstract*— A parameter optimization method is presented for controller design. Rather than dealing with the multiple specifications directly, which are difficult to optimize over, a series of cost functions are defined such that an improvement in a cost function results in an improvement in the related specifications. The cost functions are defined in such a way that they and all the gradients have analytical expressions. A unified framework is presented so that controllers of any order and structure may be considered. Because the analytical expressions for the cost functions are only valid for stabilizing controllers, the use of LQ cost functions as barrier functions to enforce stability is introduced. By use of multiplier methods it is shown how to obtain initial stabilizing controllers that can stabilize p specified plants simultaneously as well as how to impose H_∞ constraints.**

***Index Terms*— control systems, linear systems, optimal control, optimization methods, robustness.**

I. Introduction

Practical controller implementations generally have a number of specifications that must be satisfied, and are often decentralized in structure. For example, aircraft systems are made up of a number of subsystems, for instance, the longitudinal system, the lateral system, the propulsion system and the actuator systems, each of which has its own control system. Aircraft control systems also have a large number of performance requirements that have to be satisfied both at the overall system level and at the subsystem level. Examples of these requirements are actuator limitations, rise and settle times and robustness to unmodeled disturbances to name a few. As such, a multiple objective controller design method that is applicable to both centralized and decentralized systems is desirable.

Many of the examples in the literature for multiple objective control problems are mixed-norm problems [1], [2], [3], [4], [5]. For example, minimizing a weighted sum of the H_2 and H_∞ norms, minimizing the l_1 norm subject to a constraint on the H_2 norm, etc. The solutions presented to these problems are very problem-specific. Liu and Mills [6] consider the multiple objective control problem where the objective functions are assumed to be convex. As shown in Boyd and Barrett [7], this assumption is reasonable since the majority of control specifications take the form of convex functions. Liu and Mills solution is to find n sample controllers, each of which satisfies at least one of the specifications. By forming a convex combination of the resulting n closed-loop systems, a controller can be derived from this combination that satisfies all specifications simultaneously. The advantage of this method is that controllers need only be designed for one specification at a time. The disadvantage is that the controller that is obtained is generally centralized.

A. de Ruiter was formerly a Ph.D. candidate at the University of Toronto Institue for Aerospace Studies. He is currently a Visiting Fellow at the Canadian Space Agency.

H. Liu is an Assistant Professor at the University of Toronto Institue for Aerospace Studies.

Multiple Objective Parameter Synthesis (MOPS) [8] provides an automated way of tuning controllers to meet performance requirements. The method is very practical in that it does not have any underlying controller synthesis technique, and as such, it allows any controller synthesis technique to be used. Each controller synthesis technique has free parameters, p, which can be adjusted to tune the controllers. In PID laws, p consists of the gain parameters. In LQR, p consists of the entries of the Q and R matrices. The p could even be as general as the matrix entries in a state-space realization of a controller for a given controller order. In MOPS, each performance objective is assigned a positive criterion $c_i(p)$, whose value is smaller the better the objective is achieved. The design criteria may be defined in terms of many things including, but not limited to, pole placements, time domain responses and frequency domain responses. Each c_i is assigned an upper bound demand value d_i. Some of the d_i are hard constraints, ie. $c_i \leq d_i$. Others are simply normalizing values so that all c_i/d_i may be compared reasonably. Defining $\phi_i(p) \triangleq c_i/d_i$, the MOPS method involves choosing a controller synthesis method a-priori, and the solving the min-max problem

$$\min_p \max_i \{\phi_i(p)\}.$$

This can be solved using any general nonlinear optimization solver. During the optimization procedure, some of the d_i may have to be relaxed to maintain feasibility of the solution. The procedure highlights which of the objectives are conflicting. The advantages of this procedure are that firstly, any controller synthesis technique may be used. Secondly, the objective functions may be defined in many ways. Thirdly, MOPS is applicable to nonlinear systems as well as linear systems. The disadvantage is that the cost functions generally cannot be calculated analytically, and must be determined by simulation. As such, MOPS may be computationally expensive to implement.

For LTI systems, many performance specifications can be reformulated into linear matrix inequalities (LMIs). There are many examples in the literature where LMI's are used to solve various types of multi-objective control problems. Scherer et al. [9] show how LMI's may be obtained for the following specifications: H_∞ performance, generalized quadratic constraints,

H_2 performance, peak impulse response and settling time, bounds on the peak-to-peak gain, regional pole constraints, nominal regulation and robust regulation. Scherer then shows how the LMI's for each specification may be combined for a multi-objective problem. Dussy and El Ghaoui [10] show how a nonlinear control problem can be represented in a pseudo-linear form using a linear-fractional representation. LMI's can then be derived for the system with an associated additional non-convex constraint. The advantage is that LMI's may readily be solved using convex optimization techniques. Thus, the problem can be solved (if a solution exists). The disadvantage of the LMI approach is that transforming the specifications to LMI's introduces additional conservatism, and the controller is centralized.

Fleming [11] suggests that the use of genetic search algorithms to solve multiple objective problems is more robust in finding a solution than solving standard nonlinear programs. Fleming mentions that due to the stochastic nature of the search, genetic algorithms are capable of searching the entire solution space with a greater likelihood of finding a global optimum. The disadvantage is that genetic algorithms are generally very slow. On the other hand, genetic algorithms are capable of finding the set of Pareto-optimal solutions in a single run. This is an advantage over methods that convert multiple-objective problems into single-objective problems, for example by use of weighted sums, which only find a single Pareto-optimal point in a single run. So, although single-objective optimization techniques are faster than multiple-objective genetic algorithms, multiple runs must be performed with single-objective techniques to obtain multiple Pareto-optimal points to identify trade-offs between objectives. This can be a cumbersome procedure, so choosing suitable weights among objectives depends somewhat on the designer's understanding of the physics of the plant. The method presented in this paper is a single-objective technique.

In this paper, we develop a parameter optimization framework for the control system design for LTI systems. It is similar to the MOPS procedure in that the controller parameters are the variables of optimization. The difference is that instead of optimizing over the specifications directly (eg. rise time, settling time, overshoot), cost functions are defined such that an improvement in the cost function results in an improvement in the corresponding specifications. The cost functions are defined in such a way that they and their gradients have analytical expressions, and can therefore be calculated quickly and accurately without any approximations such as would occur if a finite-difference method were used on the specifications obtained by simulation. Since the only constraint on the controllers is that they are stabilizing, there is no added conservatism. The resulting controller design method is suitable for implementation in a CAD environment. In section II, cost functions are developed to address the specification types. In section III, the optimization problem is presented. In section IV, a framework is developed by which the optimization problem may be solved for any controller architecture for fixed order controllers. All of the tools necessary for being able to perform the optimization are developed in this section also. In particular, a method of obtaining a controller that stabilizes p plants simultaneously, and a method for imposing H_∞ constraints, is presented. In section V, the use of the method is demonstrated by a helicopter formation flying example.

II. Cost Function Definitions

There are many different types of specifications that must be satisfied in the design of a control system. This section deals with the cost function definition associated with different types of specifications. A parameter optimization to minimize a weighted sum of these cost functions is then performed over the controller parameters directly.

It is known that any dynamic output feedback controller can be transformed into a static output feedback controller by suitable redefinition of the inputs and outputs, resulting in a static controller of the form $\bar{u} = K\bar{y}$, where $\bar{u} \in R^{\bar{m}}$ and $\bar{y} \in R^{\bar{r}}$ are the redefined inputs and outputs respectively (for example, see [12]). As a result of this, the cost function and gradients are easy to calculate, and any type of controller structure (centralized, decentralized, hierarchical, etc.) may be dealt with. In what follows, the optimization framework is laid out for continuous-time systems.

We assume that the closed-loop system is given by

$$\begin{aligned} \dot{x}_{cl} &= A_{cl}x_{cl} + B_{cl}y^r + E_{cl}d + B_{clzw}w \\ y &= C_{cl}x_{cl} + D_{cl}y^r + F_{cl}d \\ u &= G_{cl}x_{cl} + H_{cl}y^r + F_{clu}d \\ z &= C_{clzw}x_{cl} + D_{clzw}w. \end{aligned} \quad (1)$$

where $x_{cl} \in R^n$ are the closed-loop states, $y^r \in R^r$ are reference commands, $y \in R^r$ are the outputs to be tracked, $d \in R^{\bar{d}}$ are external disturbances, $u \in R^m$ are the control inputs, and $w \in R^{\bar{w}}$ and $z \in R^{\bar{z}}$ are regulated inputs and outputs respectively for H_∞ design. It is also assumed that all inputs and outputs have been normalized by their maximum allowable values.

A. Tracking

It is assumed that all tracking specifications are defined in terms of a step response, (eg. percent overshoot, rise time, settling time). As mentioned, it is difficult to calculate and optimize over the tracking specifications directly. Instead, we try to match the response of a system that does satisfy the tracking specifications. To this end, assume we have a plant $G_d = \left[\begin{array}{c|c} A_d & B_d \\ \hline C_d & D_d \end{array}\right]$ which satisfies the tracking specifications and has zero steady-state error to a step response. Define the "error" $e_d \triangleq y - y_d$ to get

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}y^r \\ e_d &= \bar{C}\bar{x} + \bar{D}y^r \end{aligned}$$

where

$$\bar{A} = \begin{bmatrix} A_{cl} & 0 \\ 0 & A_d \end{bmatrix}, \ \bar{B} = \begin{bmatrix} B_{cl} \\ B_d \end{bmatrix}, \ \bar{C} = \begin{bmatrix} C_{cl} & -C_d \end{bmatrix},$$

$$\text{and } \bar{D} = D_{cl} - D_d.$$

Setting $\bar{x}(0) = 0$, and assuming $\bar{A}$ is asymptotically stable, we have the step response

$$e_d = \bar{C}e^{\bar{A}t}\bar{A}^{-1}\bar{B}y^r + \left(\bar{D} - \bar{C}\bar{A}^{-1}\bar{B}\right)y^r$$

from which we identify the transient error $\delta e_d \triangleq \bar{C} e^{\bar{A}t} \bar{A}^{-1} \bar{B} y^r$ and the steady-state error $e_{dss} \triangleq \left(\bar{D} - \bar{C}\bar{A}^{-1}\bar{B}\right) y^r$. Let $\{e_i\}$ denote the set of standard basis vectors for R^r. Define $\delta e_d^i \triangleq \bar{C} e^{\bar{A}t} \bar{A}^{-1} \bar{B} e_i$ and $e_{dss}^i \triangleq \left(\bar{D} - \bar{C}\bar{A}^{-1}\bar{B}\right) e_i$ for $i = 1, ..., r$. These are the transient and steady state errors for step responses to each individual reference command. To satisfy the tracking specifications, we wish to minimize both the transient and steady-state errors. To this end, define the following cost functions:

$$\delta J_t = \sum_{i=1}^{r} \int_0^\infty \delta e_d^{i\,T} \delta e_d^i \, dt = \text{tr}\{\bar{B}^T \bar{A}^{-T} L_o \bar{A}^{-1} \bar{B}\} \quad (2)$$

$$J_{tss} = \sum_{i=1}^{r} e_{dss}^{i\;T} e_{dss}^i = \text{tr}\{\left(\bar{D} - \bar{C}\bar{A}^{-1}\bar{B}\right)^T \left(\bar{D} - \bar{C}\bar{A}^{-1}\bar{B}\right)\} \quad (3)$$

where $L_o \geq 0$ satisfies the Lyapunov equation

$$\bar{A}^T L_o + L_o \bar{A} + \bar{C}^T \bar{C} = 0. \quad (4)$$

Note that $J_{tss} = 0$ if we include a servocompensator in the controller [13], so steady-state tracking is guaranteed in that case.

B. Controller Limitations

Here, we try to minimize the control response to step commands. The response of u to a step command is given by

$$u = G_{cl} e^{A_{cl} t} A_{cl}^{-1} B_{cl} y^r + \left(H_{cl} - G_{cl} A_{cl}^{-1} B_{cl}\right) y^r. \quad (5)$$

The transient control effort is defined as $\delta u \triangleq G_{cl} e^{A_{cl} t} A_{cl}^{-1} B_{cl} y^r$ and the steady-state control effort as $u_{ss} \triangleq \left(H_{cl} - G_{cl} A_{cl}^{-1} B_{cl}\right) y^r$. The control responses to individual step commands are defined as $\delta u^i \triangleq G_{cl} e^{A_{cl} t} A_{cl}^{-1} B_{cl} e_i$ and $u_{ss}^i \triangleq \left(H_{cl} - G_{cl} A_{cl}^{-1} B_{cl}\right) e_i$ for $i = 1, ..., r$. To satisfy the limitations on control effort, we wish to minimize both the transient and steady-state control efforts, as well as the control rate. Thus, the following cost functions are defined:

$$\delta J_u = \sum_{i=1}^{r} \int_0^\infty \delta u^{iT} \delta u^i \, dt = \text{tr}\{B_{cl}^T A_{cl}^{-T} L_u A_{cl}^{-1} B_{cl}\} \quad (6)$$

$$\begin{aligned} J_{uss} &= \textstyle\sum_{i=1}^r u_{ss}^{i\;T} u_{ss}^i \\ &= \text{tr}\{\left(H_{cl} - G_{cl} A_{cl}^{-1} B_{cl}\right)^T \left(H_{cl} - G_{cl} A_{cl}^{-1} B_{cl}\right)\} \end{aligned} \quad (7)$$

where $L_u \geq 0$ satisfies the Lyapunov equation

$$A_{cl}^T L_u + L_u A_{cl} + G_{cl}^T G_{cl} = 0. \quad (8)$$

Differentiating (5) gives the control rate $\dot{u} = G_{cl} e^{A_{cl} t} B_{cl} y^r$. Defining $\dot{u}^i \triangleq G_{cl} e^{A_{cl} t} B_{cl} e_i$, we define the cost function for penalizing the control rate as

$$J_{\dot{u}} = \sum_{i=1}^{r} \int_0^\infty \dot{u}^{iT} \dot{u}^i \, dt = \text{tr}\{B_{cl}^T L_u B_{cl}\}. \quad (9)$$

C. Disturbance Rejection

Since we are dealing with the closed-loop transfer function from disturbance to output, we set $y^r = 0$. In section IV-E, a method is presented where by H_∞ constraints may be added to the optimization problem. It is based on Algebraic Riccati Inequalities. However, it must be noted that the use of an H_∞ constraint does not account for disturbances not in $\mathcal{L}_2$ such as polynomial and sinusoidal type disturbances, which can occur in practice [14]. Thus, we define a cost function to limit the effect of those types of disturbances. Hence, we consider disturbances generated by $d = C_p \eta$, where $\dot{\eta} = A_p \eta$, $\eta \in R^{n_p}$ and the eigenvalues of A_p are in the closed right-half plane. Let the characteristic equation of A_p be given by $\lambda^{n_p} + \alpha_1 \lambda^{n_p - 1} + ... + \alpha_{n_p} = 0$. With successive integrations by parts, and the Cayley-Hamilton theorem, it can be shown that with A_{cl} stable, the closed-loop response to a disturbance with initial condition $\eta(0) = \eta_0$ is

$$y = C_{cl} \Gamma_d^{-1} e^{A_{cl} t} \Sigma_d \eta_0 + (D_{cl} C_p - C_{cl} \Gamma_d^{-1} \Sigma_d) e^{A_p t} \eta_0, \quad (10)$$

where $\Gamma_d \triangleq A_{cl}^{n_p} + \alpha_1 A_{cl}^{n_p - 1} + ... + \alpha_{n_p} I$ and $\Sigma_d \triangleq A_{cl}^{n_p - 1} B_{cl} C_p + \sum_{j=1}^{n_p - 1} (A_{cl}^{n_p - 1 - j} B_{cl} C_p A_p^j + \alpha_j \sum_{i=1}^{n_p - j} A_{cl}^{i-1} B_{cl} C_p A_p^{n_p - j - i})$. Note that since A_{cl} is stable, Γ_d is invertible. The transient and steady-state responses are defined as $\delta y \triangleq C_{cl} \Gamma_d^{-1} e^{A_{cl} t} \Sigma_d \eta_0$ and $y_{ss} \triangleq \left(D_{cl} C_p - C_{cl} \Gamma_d^{-1} \Sigma_d\right) e^{A_p t} \eta_0$, and we define the corresponding cost functions

$$\begin{aligned} \delta J_d &= \text{tr}\{\textstyle\int_0^\infty \Sigma_d^T e^{A_{cl}^T t} \Gamma_d^{-T} C_{cl}^T C_{cl} \Gamma_d^{-1} e^{A_{cl} t} \Sigma_d \, dt\} \\ &= \text{tr}\{\Sigma_d^T P_d \Sigma_d\} \end{aligned} \quad (11)$$

and

$$J_{dss} = \text{tr}\{\left(D_{cl} C_p - C_{cl} \Gamma_d^{-1} \Sigma_d\right)^T \left(D_{cl} C_p - C_{cl} \Gamma_d^{-1} \Sigma_d\right)\} \quad (12)$$

where $P_d \geq 0$ satisfies $A_{cl}^T P_d + P_d A_{cl} + \Gamma_d^{-T} C_{cl}^T C_{cl} \Gamma_d^{-1} = 0$. Note that as for the tracking specifications results, if a servocompensator is included in the controller, $J_{dss} = 0$.

D. Gain Limitations

In the case that we wish to ensure that the controller gains do not become too large, we may penalize them in the cost function by adding

$$J_{gain} = \text{tr}\{K^T K\}. \quad (13)$$

E. Pole Location Constraint

The simplest pole location constraint is the stability constraint. It may be desirable to constrain the closed-loop eigenvalues to a region inside the left-half plane. The case the method in this paper can deal with is where the closed-loop poles are desired to lie to the left of $-\delta$, $\delta \geq 0$. Thus, instead of enforcing the stability of A_{cl}, we enforce the stability of $A_{cl} + \delta I$ (for stability only, set $\delta = 0$). The following LQ cost function ensures that the cost forms a natural boundary for the stability region:

$$J_{PL} = E\left[\int_0^\infty x^T Q x \, dt\right] = \text{tr}\{P_{PL} M_o\} \quad (14)$$

where $Q = Q^T > 0$, $x(t)$ is the unforced response of $\dot{x} = (A_{cl} + \delta I)x$ subject to a nonzero initial condition. Thus, P_{PL} is the unique solution of

$$(A_{cl} + \delta I)^T P_{PL} + P_{PL}(A_{cl} + \delta I) + Q = 0$$

where $E[\cdot]$ denotes the expectation operator, and $M_o = E[x(0)x(0)^T] > 0$.

F. Robustness and Fault Tolerance

As mentioned in the section on disturbance rejection, H_∞ type constraints may be added using the method presented in section IV-E. An alternative (or additional) method of considering robustness, is to define a finite set of plants which reflects the allowable deviations of the actual plant from the nominal plant. For example, if the center of mass (COM) of an aircraft is known to lie between x_{min} and x_{max}, with nominal value x_{nom}, then three such models could correspond to the aircraft with COM at x_{nom}, x_{min} and x_{max} respectively. Similarly, if the form of the unmodeled dynamics are known and are linear, then they can be appended to one of the models. This is done in the example in section V. Let these closed-loop plants be given by

$$\begin{array}{rcl} \dot{x}_{cl} & = & A_{cl}^i x_{cl} + B_{cl}^i y^r + E_{cl}^i d \\ y & = & C_{cl}^i x_{cl} + D_{cl}^i y^r + F_{cl}^i d \\ u & = & G_{cl}^i x_{cl} + H_{cl}^i y^r + H_{clu}^i d \end{array} \qquad (15)$$

be the closed-loop systems for $i = 1, ..., p$ where p is the number of finite set of plant deviations that have been defined. For robust stability, we simply add LQ cost functions using (14) with A_{cl}^i for each i. For robust performance, we can add which ever we desire from tracking, controller limitations and disturbance rejection cost functions as calculated by (2), (3), (6), (7), (9), (11) and (12) using (15) for each i.

The approach to fault tolerance is exactly the same as for robustness, except that the plants in (15) now refer to the finite set of closed-loop systems corresponding to allowable component failures.

G. Definition of the Desired System G_d

Consider SISO systems of the form

$$G_{di}(s) = \frac{d_{ni}^2}{s^2 + 2\zeta_i d_{ni} s + d_{ni}^2}. \qquad (16)$$

These systems have the property that $G_{di}(0) = 1$, so that asymptotic tracking to a step response automatically occurs if the system is asymptotically stable. The poles are given by $s = -\sigma \pm j d_d$ where $\sigma = \zeta_i d_{ni}$ and $d_d = d_{ni}\sqrt{1-\zeta_i^2}$. ζ_i is called the damping ratio and d_{ni} is the undamped natural frequency. It is well known that for $0 \leq \zeta \leq 1$, the tracking specifications are given by rise-time $t_r \approx \frac{1.8}{d_{ni}}$, overshoot $M_p = e^{-\pi\zeta_i/\sqrt{1-\zeta_i^2}}$ and settling time $t_s = \frac{4.6}{\zeta_i d_{ni}}$. Thus, it is very easy to define a second order system with the desired tracking properties. Defining $G_d \triangleq \text{diag}\{G_{di}\}$, we have a system that satisfies the tracking specifications and is decoupled.

III. Multiple-Objective Controller Design

Having defined all of the cost functions and constraints, the controller is designed by solving the single-objective optimization problem

$$\begin{array}{rcl} \min_K J & = & w_{\delta t}\delta J_t + w_{tss}J_{tss} + w_{\delta u}\delta J_u + w_{uss}J_{uss} \\ & & +w_{\dot{u}}J_{\dot{u}} + w_{\delta d}\delta J_d + w_{dss}J_{dss} + w_{PL}J_{PL} \\ & & +w_{gain}J_{gain} + W, \end{array} \qquad (17)$$

where each $w \geq 0$, W is the weighted sum of any additional cost functions added for robustness and K are the controller parameters. The parameters available for the control system designer to vary are the weights w and the damping ratios and the undamped natural frequencies (ζ_i and d_{ni}) in the desired system G_d. By finding a local minimum for (17), a Pareto optimal solution is found with respect to the cost functions, though not necessarily with respect to the original specifications. Having said this, experience showed that by varying the weights and G_d for the above problem, it was easy to identify trade-offs between them.

IV. A Unified Framework for Optimization

In the following development, we assume that a plant has already been transformed into a static decentralized output feedback problem. To do this, the controller order must be fixed. Clearly, it is desirable to have the controller of least order that gives acceptable performance. One method is to start with the simplest controller (static output feedback), and then successively increase the order until satisfactory performance is achieved.

Thus, we consider the system

$$\begin{array}{rcl} \dot{x} & = & \bar{A}x + \bar{B}\bar{u} + \Gamma y^r + \bar{E}d + E_z w \\ \bar{y} & = & \bar{C}x + \bar{D}\bar{u} + \bar{D}_r y^r + \bar{F}d + \bar{F}_z w \\ u & = & H\bar{u} \\ y & = & Cx + D\bar{u} + Fd \\ z & = & C_z x + D_z \bar{u} + F_z w, \end{array} \qquad (18)$$

with control law

$$\bar{u} = K\bar{y}. \qquad (19)$$

With (18) and (19), the closed-loop matrices can be readily calculated. For example (the remaining expressions can be found in [12]),

$$A_{cl} = \bar{A} + \bar{B}\left(I - K\bar{D}\right)^{-1} K\bar{C} \qquad (20)$$

A. Calculation of Gradients

Based on the notation of (18) and (19), the cost function gradients are readily obtained. For example (the remaining expressions can be found in [12]),

$$\begin{array}{rcl} \nabla_{K_f}\delta J_t & = & (I - K\bar{D})^{-T}\left[2\bar{B}^T A_{cl}^{-T}(L_{o11}A_{cl}^{-1}B_{cl} + L_{o12} \right. \\ & & \times A_d^{-1}B_d)(\bar{D}_r^T - B_{cl}^T A_{cl}^{-T}\bar{C}^T) + 2(\bar{B}^T L_{o11} \\ & & +D^T C_{cl})\Delta\bar{C}^T + (\bar{B}^T L_{o12} - D^T C_d)\Sigma^T\bar{C}^T\left.\right] \\ & & \times(I + K^T(I - K\bar{D})^{-T}\bar{D}^T) \end{array}$$

where

$$\begin{array}{rcl} A_{cl}\Delta + \Delta A_{cl}^T + A_{cl}^{-1}B_{cl}B_{cl}^T A_{cl}^{-T} & = & 0, \\ A_{cl}\Sigma + \Sigma A_d^T + 2A_{cl}^{-1}B_{cl}B_d^T A_d^{-T} & = & 0. \end{array}$$

The subscript $_f$ represents the fact that all entries in K are allowed to be varied. In general, we will impose a decentralized controller structure. Thus, we let

$$K = \begin{bmatrix} K_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & K_m \end{bmatrix}.$$

The gradients for the decentralized controller ($\nabla_K J$) are simply the block diagonal part of the full gradients ($\nabla_{K_f} J$).

B. Hestenes Multiplier Method

Suppose we want to find a variable X such that X satisfies the constraints

$$\Gamma(X) < 0 \quad (21)$$

$$\Phi(X) = 0 \quad (22)$$

and that we have a family of approximations to X that are given by $X_i = P_i(X),\ i = 1, ..., a,\ a \geq 1$, each of which satisfy (22). Suppose also that we have a cost function $\bar{J}(X)$ that is bounded below such that $\bar{J}(X) < \infty$ if and only if $\Gamma(X) < 0$. Then a solution to (21) and (22) may be found by using the Hestenes multiplier method (see [15] and [16] for details). Specifically, we try to solve

$$\min \bar{J} \text{ subject to } \Phi(X) = 0.$$

Clearly, the solution to this satisfies (21) and (22). The usefulness of the Hestenes method is that it does not require a feasible initial point. Defining the augmented Lagrangian to be

$$\hat{J}(X, \Lambda) \triangleq \bar{J}(X) + tr[\Lambda^T \Phi(X)] + \frac{1}{2} tr[\Phi^T(X)\Phi(X)], \quad \delta > 0, \quad (23)$$

where Λ is a Lagrange multiplier, the Hestenes multiplier method is:

1. Initialize $\Lambda = 0$. Choose a constant $\beta_1 \geq 1$ and set

$$\delta \geq 2\beta_1 \bar{J}(X)/tr[\Phi^T(X)\Phi(X)].$$

2. Holding Λ and δ fixed, minimize the augmented Lagrangian $\hat{J}(X, \Lambda)$.
3. Check (21) for each approximation X_i. If (21) is satisfied for some X_i, stop. Otherwise,
4. Using the Hestenes updating rule, update Λ and δ by

$$\Lambda_{new} = \Lambda + \delta\Phi(X)$$

$$\delta_{new} = \beta_2 \delta,\ \beta_2 \geq 1$$

5. Return to 2.

Remark In general, the function $\bar{J}(X)$ is nonconvex, so at best we can find a local minimum. It is shown in [16] that convergence is guaranteed if $\bar{J}(X)$ satisfies the second order sufficient conditions for a constrained minimum, ie. positive definiteness of the Hessian of the Lagrangian along the feasible directions at the solution, and linear independence of the gradients of the constraints. For both cases in this paper, the linear independence condition is satisfied. However, without knowing the solution to the constrained problem (this is what we are trying to find), it is difficult to verify the positive definiteness condition. Having said this, experience shows that the method converges even for very complex problems. The convergence in section IV-D was quite fast, while the convergence in section IV-E was slow, although it did converge.

C. Determination of an Initial Stabilizing Controller

It is clear that the cost functions as defined in the previous section are only well-defined if the controller is stabilizing. Thus, before any optimization can be performed, an initial stabilizing controller of the required decentralized structure must be found. There are many methods by which stabilizing output feedback controllers may be found. The most useful method for the situation in this paper is the method developed by Wenk and Knapp [17] since a stabilizing controller of arbitrary structure may be found. This method essentially begins with a full stabilizing controller found by state feedback, and then uses the Hestenes multiplier method to gradually enforce the decentralized structure, while maintaining closed-loop stability of the system.

D. Simultaneous Structured Output Feedback Stabilization of p Plants

For robust stability, we may require the controller to stabilize a number of perturbed plants simultaneously. Thus, to perform the resulting optimization, we need an initial controller that stabilizes those plants simultaneously. Assume that we are given p plants, G_{pi}, of the form (18) with p stabilizing controllers $\bar{u}^i = K_i \bar{y}^i$ where the K_i have the required structure. Let A^i_{cl} correspond to the closed-loop A-matrix for the system G_{pi} with controller K_i. Referring to section IV-B, the Hestenes multiplier method may be used by defining $X = \text{col}_i\{K_i\}$, $\Gamma(X) \triangleq \max\{Re[sp(A^i_{cl})], i = 1, ..., p\}$, the constraint

$$\Phi(X) = \begin{bmatrix} K_1 - K_2 \\ K_2 - K_3 \\ \vdots \\ K_{p-1} - K_p \end{bmatrix}.$$

Since there are p plants, we define p approximations to X by

$$X_i = P_i(X) \triangleq \begin{bmatrix} K_i \\ \vdots \\ K_i \end{bmatrix}, \quad i = 1, ..., p,$$

i.e. for each i, we set $K_1 = \cdots = K_i = \cdots = K_p$. As cost, we take $\bar{J} = \sum_{i=1}^{p} J_i(K_i, G_{pi})$ where for each plant G_{pi}, J_i is defined by (17). Having found a suitable approximation X_i using the Hestenes method, a controller that stabilizes all plants simultaneously is given by K_i.

E. H_∞ Constraints

For the purposes of H_∞ control design, we have added a regulated output z and an external input w to the plant. We consider the system

$$T_{zw} = \left[\begin{array}{c|c} A_{cl} & B_{clzw} \\ \hline C_{clzw} & D_{clzw} \end{array}\right]. \quad (24)$$

Thus, the methodology covers the cases of tracking, disturbance rejection and system robustness. For tracking, simply let w contain y^r and let z contain the tracking errors. For disturbance rejection, let w contain d and let $z = y$. For robustness, it is well known from the small gain theorem [18] that if $||T_{zw}||_\infty < \gamma$, and the system uncertainty is given by $w = \Delta(s)z$, then the overall system is stable for all $\Delta \in RH_\infty$ such that $||\Delta||_\infty < 1/\gamma$. The bounded real lemma [18] gives a bound on the H_∞ norm on a system. This lemma can be rewritten as

Lemma 1: Assume $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \in RH_\infty$. Define $R \triangleq \gamma^2 I - D^T D$ and

$$\begin{aligned} \text{ARI}(G(s), Y, \gamma) \quad &\triangleq \quad Y(A + BR^{-1}D^T C) + (A + BR^{-1} \\ &\quad \times D^T C)^T Y + YBR^{-1}B^T Y + C^T \\ &\quad \times (I + DR^{-1}D^T)C \\ &< \quad 0. \end{aligned}$$

Then, $||G(s)||_\infty < \gamma$ if and only if there exists an $Y = Y^T > 0$ such that

$$\tilde{J}_\infty = \text{tr}\{\int_0^\infty e^{\text{ARI}t}\, dt\} + \text{tr}\{\int_0^\infty e^{-Rt}\, dt\} < \infty.$$

Proof: $\tilde{J}_\infty = \text{tr}\{\int_0^\infty e^{\text{ARI}t}\, dt + \int_0^\infty e^{-Rt}\, dt\} = \sum_i \int_0^\infty e^{\lambda_{ARI,i}t}\, dt + \sum_j \int_0^\infty e^{-\lambda_{R,j}t}\, dt$ where $\lambda_{ARI,i}$ and $\lambda_{R,j}$ are the eigenvalues of ARI and R respectively. Thus, $\tilde{J}_\infty < \infty$ if and only if $\lambda_{ARI,i} < 0$ and $\lambda_{R,j} > 0$, which proves the result by the bounded real lemma. ■

Since $\tilde{J}_\infty$ has the property that it is bounded below (it is positive), and $\tilde{J}_\infty \to \infty$ as ARI and/or R become singular, it forms a useful barrier function for enforcing the H_∞ constraint. Since for different sets of plant parameters, Y may be different, we must make Y a variable as well. Thus we must enforce the positive definiteness of Y as well. Instead of considering Y directly, we consider instead the cholesky factorization, $Y = W^T W$ where W is upper triangular, and define the barrier function

$$\begin{aligned} J_\infty(K, W, \gamma) \quad &= \quad \text{tr}\{\textstyle\int_0^\infty e^{\text{ARI}(T_{zw}(s),Y,\gamma)t}\, dt\} \\ &\quad + \text{tr}\{\textstyle\int_0^\infty e^{-Rt}\, dt\} + \text{tr}\{\int_0^\infty e^{-Yt}\, dt\}. \end{aligned}$$

A simple calculation shows that for ARI< 0, $R > 0$ and $X > 0$,

$$\begin{aligned} J_\infty(K, W, \gamma) \quad &= \quad \text{tr}\{-\text{ARI}(T_{zw}(s), Y, \gamma)^{-1}\} + \text{tr}\{R^{-1}\} \\ &\quad + \text{tr}\{Y^{-1}\}. \end{aligned}$$

Assume that we wish to add an H_∞ constraint to the multiple objective controller design problem (17), i.e., we want to solve

$$\min_K J, \;\; s.t. \;\; ||T_{zw}||_\infty < \gamma_d$$

where $\gamma_d > 0$ is the desired value of the upper bound on $||T_{zw}||_\infty$. Instead of solving this problem directly, we instead solve the problem

$$\min_{K,W} J + w_\infty J_\infty(K, W, \gamma_d).$$

where $w_\infty > 0$ is a small positive constant. An initial controller K, and nonsingular upper triangular matrix W need to be found so that $J_\infty(K, W, \gamma_d)$ is finite. This can be done using the Hestenes's multiplier method by making γ a variable as well. Thus, referring to section IV-B, define $X \triangleq (K, W, \gamma)$, $\Gamma(X) \triangleq \text{ARI}(T_{zw}(s), Y, \gamma)$, the constraint $\Phi(X) = \gamma - \gamma_d$, and the cost $\bar{J}(X) \triangleq J + w_\infty J_\infty$. We only need one approximation to X, and this is $X_1 = P_1(X) = (K, W, \gamma_d)$, where W is left free. Finally, we modify step 3 in IV-B slightly to be:
3. Check if $\Gamma(X_1) < 0$ has a positive definite solution $\bar{Y}$. If yes, stop and obtain $\bar{W}$ from the Cholesky factorization of $\bar{Y}$. K and $\bar{W}$ are the initial matrices we are looking for. If not, go to step 4.

To solve the Hestenes problem above, we need an initial K, W, and γ that satisfies ARI < 0, $R > 0$, $Y > 0$ and $\gamma > 0$. Using the method in IV-C, an initial stabilizing controller K can be found. Taking $\gamma = ||T_{zw}|| + \epsilon$, where ϵ is a small positive number, the ARI can then be readily solved for Y using a standard LMI technique.

F. Well-Posedness

By looking at the closed-loop system (20), it is easy to see that we must have $I - K\bar{D}$ nonsingular. While the cost functions provide a natural barrier to maintain the closed-loop stability requirement, no such thing is true for the well-posedness requirement in general. To enforce well-posedness, the following barrier function may be added to the cost function

$$J_{wp} = \frac{a}{\det(I - K\bar{D})} \tag{25}$$

with gradient

$$\nabla_{K_f} J_{wp} = \frac{a(I - K\bar{D})^{-1}\bar{D}^T}{\det(I - K\bar{D})}. \tag{26}$$

where $a > 0$ is a small positive constant.

V. Example: Helicopter Formation Flying

In the flight systems and control laboratory at the University of Toronto Institute for Aerospace Studies is a hardware-in-the-loop experiment consisting of three 3 degree of freedom helicopters (see fig. 1). We are going to have the first helicopter track a commanded trajectory, and then the second helicopter will follow the first and the third will follow the second. All states are available for feedback, and the variables we wish to track are the elevation ϵ and the travel λ. For the purposes of control system design, each of the helicopters are given by

Fig. 1. 3-DOF Helicopters

$$\begin{aligned} \dot{x}_i &= A_i x_i + B_i u_i + E_i d_i + E_{zi} w_i \\ y_i &= C_i x_i \qquad\qquad\qquad\qquad i = 1, 2, 3 \\ z_i &= C_{zi} x_i + D_{zi} u_i \end{aligned} \tag{27}$$

where the y_i are the outputs to be tracked and z_i are the regulated outputs for the purposes of H_∞ design. Thus, the regulated outputs are the motor voltages, and the helicopter states elevation, pitch and travel. Together with the definition of the regulated inputs, the H_∞ design captures both unmodeled plant and actuator dynamics. The tracking errors are given by

$$e_1 = y_1 - y_1^r,\ e_2 = y_2 - y_1 \text{ and } e_3 = y_3 - y_2. \tag{28}$$

The measurements available are the tracking errors, and the states that are not being tracked. The general form of the measurements are

$$\begin{aligned} y_1^m &= C_1^m x_1 + \Gamma_1 y_1^r, \\ y_2^m &= C_2^m x_2 + \Gamma_2 y_1, \\ y_3^m &= C_3^m x_3 + \Gamma_3 y_2. \end{aligned}$$

To guarantee asymptotic tracking for step inputs, we need integral control. This also simplifies J, since $J_{tss} = J_{dss} = 0$ and these do not need to be considered. Thus, the integrators are defined by

$$\dot{\eta}_i = e_i \ \ i = 1, 2, 3.$$

Augmenting each plant with the integrators and assembling the overall system, we obtain

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \dot{\bar{x}}_3 \end{bmatrix} &= \begin{bmatrix} \bar{A}_1 & 0 & 0 \\ \bar{A}_{21} & \bar{A}_2 & 0 \\ 0 & \bar{A}_{32} & \bar{A}_3 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} \\ &+ \begin{bmatrix} \bar{B}_1 & 0 & 0 \\ 0 & \bar{B}_2 & 0 \\ 0 & 0 & \bar{B}_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix} y_1^r \\ &+ \begin{bmatrix} \bar{E}_1 & 0 & 0 \\ 0 & \bar{E}_2 & 0 \\ 0 & 0 & \bar{E}_3 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \\ &+ \begin{bmatrix} \bar{E}_{z1} & 0 & 0 \\ 0 & \bar{E}_{z2} & 0 \\ 0 & 0 & \bar{E}_{z3} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \end{aligned} \tag{29}$$

where

$$\bar{A}^i = \begin{bmatrix} A_i & 0 \\ C_i & 0 \end{bmatrix},\ \bar{A}_{i+1,i} = \begin{bmatrix} 0 & 0 \\ -C_i & 0 \end{bmatrix},\ \ i = 1, 2, 3,$$

$$\bar{B}_i = \begin{bmatrix} B_i \\ 0 \end{bmatrix},\ \bar{E}_i = \begin{bmatrix} E_i \\ 0 \end{bmatrix},\ \bar{E}_{zi} = \begin{bmatrix} E_{zi} \\ 0 \end{bmatrix},\ \ i = 1, 2, 3,$$

and

$$\Gamma = \begin{bmatrix} 0 \\ -I \end{bmatrix}.$$

The integrator states are also available to the control law, so we must augment the measurement equations accordingly.

$$\begin{bmatrix} \bar{y}_1^m \\ \bar{y}_2^m \\ \bar{y}_3^m \end{bmatrix} = \begin{bmatrix} \bar{C}_1^m & 0 & 0 \\ \bar{C}_{21}^m & \bar{C}_2^m & 0 \\ 0 & \bar{C}_{32}^m & \bar{C}_3^m \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} + \begin{bmatrix} \bar{\Gamma}_1 \\ 0 \\ 0 \end{bmatrix} y_1^r,$$

where

$$\bar{C}_i^m = \begin{bmatrix} C_1^m & 0 \\ 0 & I \end{bmatrix},\ \ i = 1, 2, 3,$$

$$\bar{C}_{i+1,i}^m = \begin{bmatrix} \Gamma_{i+1} C_i & 0 \\ 0 & 0 \end{bmatrix},\ \ i = 1, 2.$$

We wish $y_1 \to y_1^r$ and $e_i \to 0,\ i = 2, 3, 4,$ as $t \to \infty$. Therefore, we define the output to be

$$y = \begin{bmatrix} y_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \bar{C}_1 & 0 & 0 \\ -\bar{C}_1 & \bar{C}_2 & 0 \\ 0 & -\bar{C}_2 & \bar{C}_3 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix}, \tag{30}$$

where

$$\bar{C}_i = \begin{bmatrix} C_i & 0 \end{bmatrix},\ \ i = 1, 2, 3.$$

We choose the desired closed-loop system to be

$$\begin{bmatrix} \epsilon_1 \\ \lambda_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} g_\epsilon(s) & 0 \\ 0 & g_\lambda(s) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1^r \\ \lambda_1^r \end{bmatrix} \tag{31}$$

The general form of the non-zeros terms above is $g_.(s) = \frac{d_.^2}{s^2 + 2\zeta_. d_. s + d_.^2}$, with the numerical values $d_\epsilon = 2$, $\zeta_\epsilon = 0.9$, $d_\lambda = 1.5$ and $\zeta_\lambda = 0.9$. For robustness, two more helicopter models are created. Thus, we have three different helicopter models. This is not to be confused with the three physical helicopters, lead, follower 1 and follower 2 (the lead, follower 1 and follower 2 all have the same model). The first helicopter model is the nominal model, and will be used in the definition of all of the performance cost functions. The second model has the same form as the first, except that it has perturbed mass and geometrical properties. It was found that when the design was performed enforcing stability only for these two helicopter models, the resulting controller was unstable on the real system, even though it was stable on a fully nonlinear simulation. This was suspected to be due to the unmodeled actuator dynamics of the motors. Since the actuator dynamics are unknown, we approximate them with first-order low-pass filters. Adding these to the nonlinear simulation showed that these had the effect of destabilizing the system. Thus, for the third helicopter model, we append the first-order actuator dynamics

$$V_f = \frac{1}{0.3s + 1} \bar{V}_f \text{ and } V_b = \frac{1}{0.3s + 1} \bar{V}_b$$

to the nominal helicopter dynamics.

LQR was used to find a stabilizing controller for the nominal model. It was found that this controller simultaneously stabilized all three helicopter models, so it may be used as an initial controller for the optimization procedure. It was found that the magnitude of the controller gains had to be significantly restricted if the resulting controller was to stabilize the real system. The result of this restriction is that we were unable to match the desired step response very closely.

Optimizing (without the H_∞ constraint), while enforcing stability of the three helicopter models yielded good step responses of the nominal linearized helicopter models, as seen in fig. 2. However, when the resulting controller was implemented on the real system, the response was very poor as seen in figs. 3 and 4. This was found to be due to friction in the pivots. To overcome this problem, the closed-loop eigenvalues must be forced to be fast enough, so that the friction can be overcome in a short enough time. Thus, the optimization was performed again (without the H_∞ constraint), with the closed-loop eigenvalues of the nominal lead helicopter being forced to lie to the left of -0.3, and the closed-loop eigenvalues of the nominal follower helicopters being forced to lie to the left of -0.2. This is at the expense of the performance of the linearized models, however the performance of the real system is much better (since the system performance was not significantly changed with the added H_∞ constraint as described in the next paragraph, figs. 5, 6 and 7 are representative of this case as well).

Finally, we added the H_∞ constraint as well as the eigenvalue restriction. With the initial stabilizing controller, for each helicopter, the value of the H_∞ norm is $||T_{zwi}||_\infty = 3$ for $i = 1, 2, 3$. We add the constraint that for each helicopter, $||T_{zwi}||_\infty < 2.5$ for $i = 1, 2, 3$. Fig. 5 shows the response of the linearized system. Figs. 6 and 7 show the performance of the real system.

VI. Concluding Remarks

A parameter optimization method has been presented for multiple-objective controller design for controllers of any order and structure. Rather than dealing with the objectives directly, a series of cost functions are defined to deal with the objectives indirectly. By optimizing a weighted sum of these cost functions and tuning the weights, a satisfactory controller may be found. The cost functions are defined in such a way that they and all the gradients have analytical expressions. Because the analytical expressions for the cost functions are only valid for stabilizing controllers, the use of LQ cost functions as barrier functions to enforce stability has been introduced. By use of a multiplier method it has been shown how to obtain initial stabilizing controllers that can stabilize p specified plants simultaneously. A multiplier method has also been introduced for imposing H_∞ constraints. The use of the method has been demonstrated on a helicopter formation flying example.

References

[1] N. Elia and M.A. Dahleh, *Controller Design with Multiple Objectives,* IEEE Trans. on Automatic Control, Vol. 42, No. 5, pp. 596-613, May, 1997.

[2] M.V. Salapaka and M. Khammash, *Multi-objective MIMO Optimal Control Design without Zero Interpolation,* Proc. of the American Control Conference, Philadelphia, Pennsylvania, June, 1998.

[3] C.W. Scherer, *From Mixed to Multi-Objective Control,* Proc. of the 38th Conference on Decision and Control, Phoenix, Arizona, USA, December 1999.

[4] C.W. Scherer, *Lower Bounds in Multi-Objective H_2/H_∞ Problems,* Proc. of the 38th Conference on Decision and Control, Phoenix, Arizona, USA, December, 1999.

[5] M. Trentini and J.K. Pieper, *A Case Study of Multi-Objective Control: Synthesis and Analysis,* Proc. of the 1998 IEEE International Conference on Control Applications, Trieste, Italy, September 1-4, 1998.

[6] H.H.T. Liu and J.K. Mills, *Multiple Simultaneous Specifications Control Problem and Its Application,* Proc. of the American Control Conference, Philadelphia, Pennsylvania, June, 1998.

[7] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance,* Prentice-Hall, 1991.

[8] H.D. Joos, *A methodology for multi-objective design assessment and flight control synthesis tuning,* Aerospace Science and Technology, no. 3, pp. 161–176, 1999.

[9] C. Scherer, P. Gahinet and M. Chilali, *Multiobjective Output-Feedback Control via LMI Optimization,* IEEE Trans. on Automatic Control, Vol. 42, No. 7, pp. 896–911, July, 1997.

[10] S. Dussy and L. El Ghaoui, *Measurement-Scheduled Control for the RTAC Problem,* Proc. of the American Control Conference, Albuquerque, New Mexico, June, 1997.

[11] P. Fleming, *Designing control systems with multiple objectives,* IEE Colloquium on Advances in Control Technology, 1999.

[12] A.H.J. de Ruiter, *Integrated Multiple-Objective Control Design Methodologies - with applications to flight control* Ph.D. Thesis, University of Toronto, 2005.

[13] E.J. Davison, *The Robust Decentralized Control of A General Servomechanism Problem,* IEEE Trans. on Automatic Control, vol. 21, no. 1, pp. 14–24, February, 1976.

[14] E.J. Davison and I.J. Ferguson, *The Design of Controllers for the Multivariable Robust Servomechanism Problem Using Parameter Optimization Methods,* IEEE Trans. on Automatic Control, vol. 26, no. 1, pp. 93–110, February, 1981.

[15] M.R. Hestenes, *Multiplier and Gradient Methods,* Journal of Optimization Theory and Applications, Vol. 4, No. 5, pp. 303–320, 1969.

[16] D.P. Bertsekas, *Combined Primal-Dual and Penalty Methods for Constrained Minimization,* SIAM Journal on Control, Vol. 13, No. 3, pp. 521–544, May 1975.

[17] C. Wenk, C. Knapp, *Parameter optimization in linear systems with arbitrarily constrained controller structure,* IEEE Trans. on Automatic Control, vol. 25, no. 3, pp. 496–500, June, 1980.

[18] K. Zhou with J.C. Doyle, *Essentials of Robust Control,* Prentice Hall, Upper Saddle River, N.J., 1998.

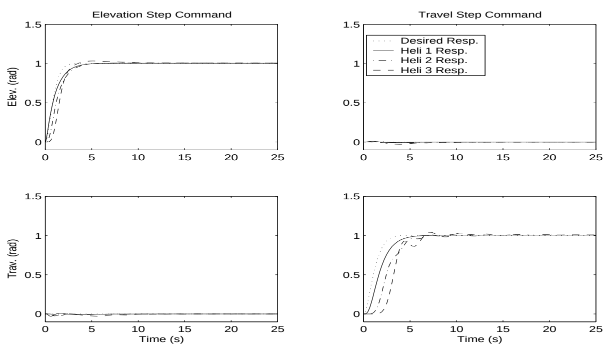

Fig. 2. Step Response of Linearized Helicopters, no eigenvalue restriction

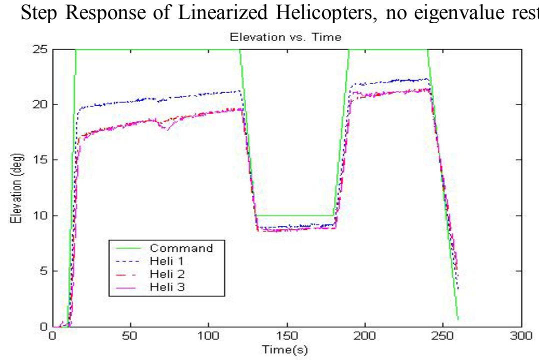

Fig. 3. Helicopter Elevation Response, no eigenvalue restriction

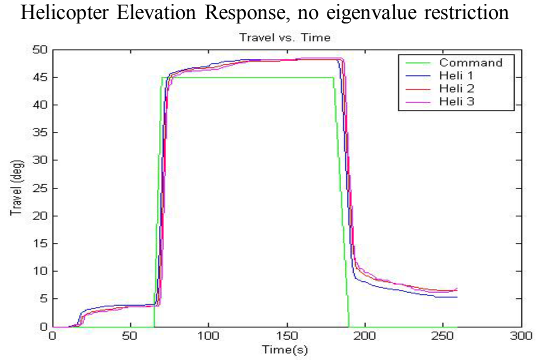

Fig. 4. Helicopter Travel Response, no eigenvalue restriction

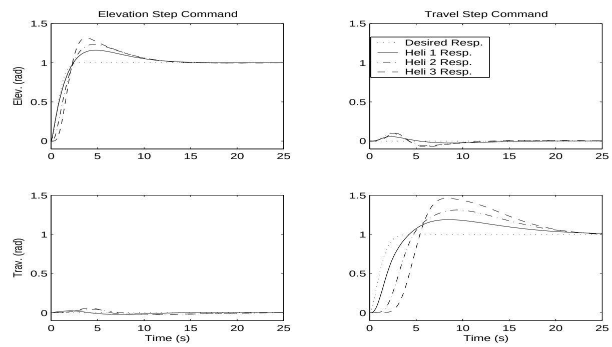

Fig. 5. Step Response of Linearized Helicopters, eigenvalues restricted, H_∞ constraint

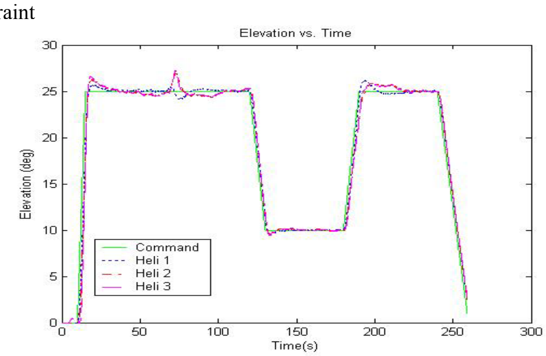

Fig. 6. Helicopter Elevation Response, eigenvalues restricted, H_∞ constraint

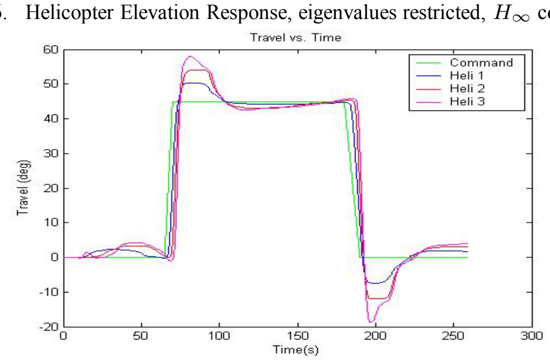

Fig. 7. Helicopter Travel Response, eigenvalues restricted, H_∞ constraint