Missile Autopilot Robustness to Uncertain Aerodynamics: Stability Hypersphere Radius Calculation

Kevin A. Wise*
McDonnell Douglas Missile Systems Company, St. Louis, Missouri 63166

Missile autopilot robustness to uncertain aerodynamic stability derivatives is determined by computing the stability hypersphere radius. Two procedures for computing this robustness measure are presented. The first method is a polynomial approach derived from Kharitonov's theorem. The second method, derived from the Lyapunov stability theory, uses a state space approach. The conservatism of these robustness tests is evaluated through application to a missile autopilot.

Introduction

MISSILE stability robustness depends upon the flight control system sensitivity to uncertain parameters and unmodeled dynamics. Robustness theory, used to determine stability robustness, can be categorized into tests derived from Kharitonov's theorem[1-9] (polynomial models), the Lyapunov stability theory[8-11] (state space models), the singular value theory[12-17] (frequency domain models), and the zero exclusion principle.[17-24] This paper presents two methods (polynomial and state space) used to compute the allowable variation bound on a vector of uncertain parameters. The square root of the magnitude of the parameter perturbation vector is called the stability hypersphere radius. The theory presented here was taken from Bhattacharyya.[8,9]

A flight control system is designed using a nominal system model. Our problem is to determine the control system sensitivity to perturbations in the nominal design parameters. This problem is solved here by calculating the radius of a stability hypersphere centered about the nominal parameter vector.

This technique is applied to a bank-to-turn missile longitudinal autopilot. The flight condition analyzed results in an open-loop unstable airframe. The uncertain parameters modeled in this analysis are the dimensional aerodynamic stability derivatives Z_α, Z_δ, M_α, and M_δ. These variables are derived from aerodynamic measurements of lift and pitching moment.

Our real parameter variation bounds presented in this paper were found to be conservative. This was attributed to the multilinear structure of the parameters entering the closed-loop characteristic polynomial. Exact bounds were computed using the DeGaston-Safonov[19] algorithm and are presented in Wise.[24] For problems that are multilinear in the polynomial coefficients, the DeGaston-Safonov[19] algorithm is preferred.

Autopilot and Missile Airframe Dynamics

The longitudinal flight control system for the missile airframe is shown in Fig. 1. Using block diagram manipulations, this autopilot is transformed into the matrix $K(s)$. The nominal rigid-body longitudinal dynamics, containing uncertain parameters, is represented by $G(s)$.

The autopilot $K(s)$ is described by the following state space quadruple (A_c, B_c, C_c, D_c):

$$A_c = \begin{bmatrix} 0 & 0 \\ K_q a_q & 0 \end{bmatrix}, \qquad B_c = \begin{bmatrix} K_a a_z & 0 \\ K_a K_q a_q & K_q a_q \end{bmatrix}$$

$$C_c = [K_q \quad 1], \qquad D_c = [K_q K_a \quad K_q]$$

where $K(s) = C_c(sI - A_c)^{-1}B_c + D_c$.

The states modeled in the open-loop, rigid-body airframe model are angle of attack α, pitch rate q, fin deflection δ, and fin rate $\dot{\delta}$. In state space form, the airframe dynamics are represented by the following state space triple (A, B, C):

$$A = \begin{bmatrix} Z_\alpha & 1 & Z_\delta & 0 \\ M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & -2\zeta\omega \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega^2 \end{bmatrix}$$

$$C = \begin{bmatrix} VZ_\alpha & 0 & VZ_\delta & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

where $G(s) = C(sI - A)^{-1}B$.

The above aerodynamics have been linearized about a trim angle of attack of 16 deg, Mach 0.8, and an altitude of 4000 ft. This creates an unstable open-loop airframe. The following parameters are the nominal values of the dimensional aerodynamic variables used in this analysis:

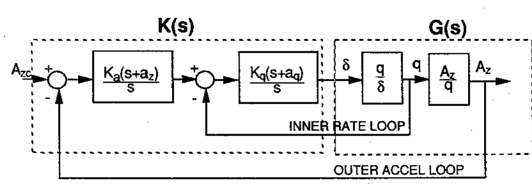
$$Z_\alpha = -1.3046 \quad 1/\mathrm{s}$$

$$Z_\delta = -0.2142 \quad 1/\mathrm{s}$$

$$M_\alpha = 47.7109 \quad 1/\mathrm{s}^2$$

$$M_\delta = -104.8346 \quad 1/\mathrm{s}^2$$

The remaining system parameters, which are assumed known, are $V = 886.78$ (ft/s), $\zeta = 0.6$, and $\omega = 113.0$ (rad/s).



UNCERTAIN PARAMETERS: Z_α, Z_δ, M_α, M_δ

FLIGHT CONDITION: M = 0.8, $\alpha = 16°$

OPEN LOOP UNSTABLE

Fig. 1 Acceleration command autopilot.

Received March 7, 1989; revision received Sept. 3, 1989.

*Staff Specialist, Advanced Guidance, Navigation, and Control. Senior Member AIAA.

The autopilot design $K(s)$ stabilizes the nominal plant model $G(s)$ using output feedback. The problem is to determine the perturbation bounds on the above imprecisely known, dimensional, aerodynamic stability derivatives such that the closed-loop system remains stable. The transfer function description of the open-loop system is

$$G(s) = \begin{bmatrix} \dfrac{\omega^2 V(Z_\delta s^2 + Z_\alpha M_\delta - Z_\delta M_\alpha)}{(s^2 - Z_\alpha s - M_\alpha)(s^2 + 2\zeta\omega s + \omega^2)} \\ \dfrac{\omega^2 (M_\delta s + M_\alpha Z_\delta - Z_\alpha M_\delta)}{(s^2 - Z_\alpha s - M_\alpha)(s^2 + 2\zeta\omega s + \omega^2)} \end{bmatrix} \tag{1}$$

Stability Hypersphere

The stability hypersphere radius can be calculated for polynomial and state space models of the closed-loop system. This section presents a polynomial method that assumes that the perturbation parameters enter the closed-loop characteristic polynomial linearly. As you may expect, this is a limiting assumption that introduces conservatism in the robustness prediction.

The flight control system is designed using nominal values of the model parameters. Arrange the open-loop transfer function polynomial coefficients into a nominal parameter vector $\boldsymbol{p}^o$. The true parameter vector is modeled as

$$\boldsymbol{p} = \boldsymbol{p}^o + \Delta\boldsymbol{p} \tag{2}$$

where $\Delta\boldsymbol{p}$ is a perturbation. Our problem is to determine the size of $\Delta\boldsymbol{p}$ such that the closed-loop system remains stable.

If our design problem contained two unknown parameters (p_1, p_2) then the nominal parameter vector would represent a point in a parameter plane, Fig. 2a. The stability hypersphere, in this case, would be a circle around this point. The parameter combinations interior to this circle would result in a stable closed-loop system. So that our robustness test is not conservative we would like at least one combination of the two parameters, lying on the circle perimeter, to create an unstable closed-loop system. Thus, the circle is tangent to the region of closed-loop instability. This is shown in Fig. 2a.

For the two-dimensional parameter space, the circle represents the stability hypersphere. The radius determines the magnitude of the allowable parameter perturbations. For an ℓ-dimensional parameter vector we have a hypersphere centered about the nominal design point in the parameter space. This stability hypersphere is tangent to the unstable hyperplane. The radius of the stability hypersphere is a measure of how large the parameter perturbations may be before instability results. This radius is an upper bound on $\|\Delta\boldsymbol{p}\|_2$ (the 2 norm of the parameter perturbation vector $\Delta\boldsymbol{p}$).

The stability hypersphere radius, shown in Fig. 2a, gives a conservative estimate on the allowable variation of parameter p_2. By using weights on the parameters, we can transform the circle of Fig. 2a into the ellipse of Fig. 2b, providing a better measure of the allowable perturbation in both p_1 and p_2. This transforms the stability hypersphere into a stability hyperellipsoid, centered about the nominal design point. The stability hyperellipsoid is also tangent to the unstable hyperplane.

In a design problem, bounds on individual parameters may be available. They are usually in the form:

$$\underline{p}_i < p_i < \bar{p}_i \tag{3}$$

Figure 2c displays these bounds about the nominal design point. Notice that the nominal design point need not be centered in the rectangle formed by the parameter perturbation bounds. In our two-dimensional parameter space, the bounds implied by Eq. (3) form a rectangle. In an ℓ-dimensional parameter space, the bounds implied by Eq. (3) form a perturbation polytope. The stability hyperellipsoid is

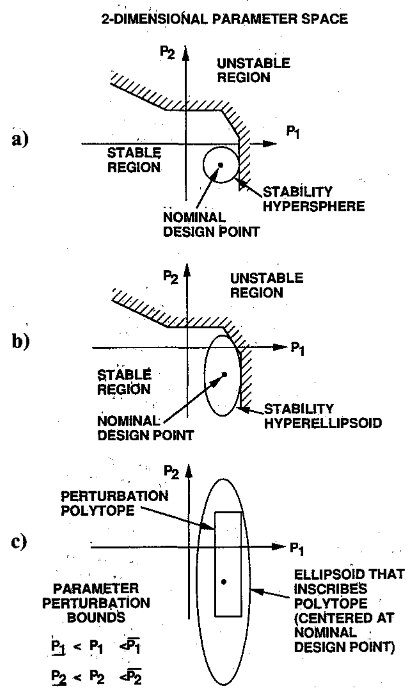


Fig. 2 Stability hyperspheres, hyperellipsoids, and perturbation polytopes.

centered about the nominal design point. If the stability hyperellipsoid inscribes the perturbation polytope, then we are guaranteed closed-loop stability for the allowable parameter perturbations.

Calculation of the Stability Hypersphere Radius

Consider the flight control system design problem pictured in Fig. 1. This problem has one control input (elevon fin deflection) and $m = 2$ sensor outputs (normal body acceleration and pitch rate). The airframe transfer function is

$$G(s) = \frac{1}{d(s)} \begin{bmatrix} n_1(s) \\ \vdots \\ n_m(s) \end{bmatrix} = N(s)\, d^{-1}(s) \tag{4}$$

where $d(s)$ is the open-loop characteristic equation, and $N(s)$ is a $m \times 1$ matrix of numerator polynomials. These can be expressed as follows:

$$\begin{aligned} d(s) &= d_p s^p + \cdots + d_o \\ N(s) &= \boldsymbol{n}_p s^p + \cdots + \boldsymbol{n}_o \end{aligned} \tag{5}$$

where the d_i are real scalars and the $\boldsymbol{n}_i$ are $m \times 1$ real vectors. It is assumed that $N(s)$ and $d(s)$ are coprime. The coefficients d_i and $\boldsymbol{n}_i$ are subject to perturbation. The nominal values are denoted by d_i^o and $\boldsymbol{n}_i^o$, with perturbation Δd_i and $\Delta\boldsymbol{n}_i$; $i = 0, ..., p$. The parameter vector $\boldsymbol{p}$ is formed by arranging the polynomial coefficients into a vector, viz.,

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{n}_p \\ d_p \\ \vdots \\ \boldsymbol{n}_o \\ d_o \end{bmatrix} \quad \boldsymbol{p}^o = \begin{bmatrix} \boldsymbol{n}_p^o \\ d_p^o \\ \vdots \\ \boldsymbol{n}_o^o \\ d_o^o \end{bmatrix} \quad \Delta\boldsymbol{p} = \begin{bmatrix} \Delta\boldsymbol{n}_p \\ \Delta d_p \\ \vdots \\ \Delta\boldsymbol{n}_o \\ \Delta d_o \end{bmatrix} \tag{6}$$

The size of the perturbation Δp is measured by its Euclidean length $\|\Delta\boldsymbol{p}\|_2$, given by

$$\|\Delta\boldsymbol{p}\|_2^2 = \|\Delta\boldsymbol{n}_o\|_2^2 + \cdots + \|\Delta\boldsymbol{n}_p\|_2^2 + (\Delta d_o)^2 + \cdots + (\Delta d_p)^2 \tag{7}$$

The autopilot transfer function is

$$K(s) = \frac{1}{d_c(s)}[\boldsymbol{n}_{co}(s) \cdots \boldsymbol{n}_{cm}(s)] = d_c^{-1}(s)N_c^T(s) \tag{8}$$

where

$$d_c(s) = d_{cq}s^q + \cdots + d_{co} \tag{9a}$$

$$N_c^T(s) = \boldsymbol{n}_{cq}^T s^q + \cdots + \boldsymbol{n}_{co}^T \tag{9b}$$

with the d_{ci} scalars and the $\boldsymbol{n}_{ci}^T$ $1 \times m$ row vectors of constant, known, real coefficients. It is assumed that $d_c(s)$ and $N_c^T(s)$ are coprime.

The closed-loop characteristic polynomial (CLCP) $\delta(s)$ is a polynomial of degree $n = p + q$ given by

$$\delta(s) = d_c(s)d(s) + N_c^T(s)N(s) \tag{10}$$

where

$$\delta(s) = \delta_n s^n + \delta_{n-1}s^{n-1} + \cdots + \delta_1 s + \delta_o \tag{11}$$

The CLCP coefficients are arranged into the closed-loop characteristic vector (CLCV) $\boldsymbol{\delta}$ with

$$\boldsymbol{\delta} = [\delta_o\ \delta_1 \cdots \delta_{n-1}\ \delta_n]^T \tag{12}$$

From Eq. (10), the controller maps the plant polynomial coefficients $(\boldsymbol{p})$ into the CLCP $\delta(s)$, which, equivalently, maps $\boldsymbol{p}$ into the CLCV $\boldsymbol{\delta}$. Thus, this mapping can be described as

$$X\boldsymbol{p} = \boldsymbol{\delta} \tag{13}$$

where $\boldsymbol{p}$ is defined in Eq. (6), $\boldsymbol{\delta}$ in Eq. (12), and the matrix X is composed of polynomial coefficients from the controller, given by

$$X = \begin{bmatrix} \begin{bmatrix} \boldsymbol{n}_{cq}^T & d_{cq} \\ \vdots & \vdots \\ \boldsymbol{n}_{co}^T & d_{co} \end{bmatrix} & \begin{bmatrix} \boldsymbol{n}_{cq}^T & d_{cq} \\ \vdots & \vdots \\ \boldsymbol{n}_{co}^T & d_{co} \end{bmatrix} & 0 \cdots 0 & \\ & & \cdots & \begin{bmatrix} \boldsymbol{n}_{cq}^T & d_{cq} \\ \vdots & \vdots \\ \boldsymbol{n}_{co}^T & d_{co} \end{bmatrix} \\ 0 \cdots 0 & & & \end{bmatrix} \tag{14}$$

where the $(q+1) \times (m+1)$ subblock

$$\begin{bmatrix} \boldsymbol{n}_{cq}^T & d_{cq} \\ \vdots & \vdots \\ \boldsymbol{n}_{co}^T & d_{co} \end{bmatrix}$$

is shifted over $m + 1$ columns and down by 1 row as it descends from left to right. X is of dimension $(q+p+1) \times (1+m)(1+p)$. For our flight control problem with two sensor outputs ($m = 2$), four state variables in the open-loop airframe model ($p = 4$), and proportional plus integral control elements in $K(s)$ ($q = 2$), the X is a 7×15 matrix. X is assumed to have full rank.

Our problem is to find the largest perturbation $\Delta\boldsymbol{p}$ such that the CLCP $\delta(s)$ remains Hurwitz (stable). The CLCP $\delta(s)$ (CLCV $\boldsymbol{\delta}$) fails to be Hurwitz if δ_n or δ_o vanish, or if any of the interior polynomial roots cross the $j\omega$ axis into the right half plane (RHP).

Define the following sets in the parameter space of the CLCV $\boldsymbol{\delta}$:

$$\Delta_o = \{\boldsymbol{\delta} \mid \boldsymbol{\delta} \in \mathscr{R}^{n+1}, \delta_o = 0\} \tag{15}$$

$$\Delta_n = \{\boldsymbol{\delta} \mid \boldsymbol{\delta} \in \mathscr{R}^{n+1}, \delta_n = 0\} \tag{16}$$

Δ_o is the set of CLCVs $\boldsymbol{\delta}$ of dimension $n + 1$, containing real polynomial coefficients, where the polynomial coefficient δ_o is zero. If $\delta_o = 0$, then the CLCP $\delta(s)$ has a zero at the origin, and is no longer asymptotically stable. The set Δ_n is defined in a similar way with $\delta_n = 0$. This creates a zero at infinity.

For any real ω define the set Δ_ω as

$$\Delta_\omega = \{\delta \mid \delta \in \mathscr{R}^{n+1}, \delta(s) = (s^2 + \omega^2)\xi(s),\ \xi(s) \text{ arbitrary})\} \tag{17}$$

Δ_ω is the set of CLCVs $\boldsymbol{\delta}$ that have roots on the $j\omega$ axis. $\xi(s)$ is arbitrary because we are concerned only with the pair of roots that are crossing into the RHP. Our interest is in the parameters in the parameter space of $\boldsymbol{p}$ that map into the sets Δ_o, Δ_n, and Δ_ω. These will be the parameters that cause the CLCP $\delta(s)$ not to be Hurwitz. To determine what parameters are contained in these sets we now define the inverse images of these sets, with respect to the controller map X, in the parameter space of $\boldsymbol{p}$. Define

$$\Pi_o = X^{-1}(\Delta_o) = \{\boldsymbol{p} \mid \boldsymbol{p} \in \mathscr{R}^k, X\boldsymbol{p} \in \Delta_o\} \tag{18}$$

$$\Pi_n = X^{-1}(\Delta_n) = \{\boldsymbol{p} \mid \boldsymbol{p} \in \mathscr{R}^k, X\boldsymbol{p} \in \Delta_n\} \tag{19}$$

$$\Pi_\omega = X^{-1}(\Delta_\omega) = \{\boldsymbol{p} \mid \boldsymbol{p} \in \mathscr{R}^k, X\boldsymbol{p} \in \Delta_\omega\} \tag{20}$$

The set Π_o contains the parameters in the parameter space of $\boldsymbol{p}$ that cause δ_o to vanish. Similarly, Π_n contains the parameters in the space of $\boldsymbol{p}$ that cause δ_n to vanish. The set Π_ω contains the parameters in the space of $\boldsymbol{p}$ that cause the CLCP$\delta(s)$ to have a pair of roots on the $j\omega$ axis. We would like to determine how close our nominal parameter vector is to the parameters contained in these three sets.

Let r_o, r_n, and r_ω denote the Euclidean distances between the nominal parameter vector $\boldsymbol{p}^o$ and Π_o, Π_n, and Π_ω, respectively. The measures r_i denote how "close" (using a 2 norm) the nominal parameter vector is to the sets of parameters that cause the CLCP $\delta(s)$ not to be Hurwitz. Let $\boldsymbol{t}_o \in \Pi_o$ ($\boldsymbol{t}_o$ is a vector in the set Π_o), then

$$r_o = \|\boldsymbol{p}^o - \boldsymbol{t}_o\|_2 \tag{21}$$

is a measure of how close the nominal parameter vector $\boldsymbol{p}^o$ is to the set Π_o. The smallest r_o is given by $\boldsymbol{t}_o^*$ Π_o, that is

$$\|\boldsymbol{p}^o - \boldsymbol{t}_o^*\|_2 \le \|\boldsymbol{p}^o - \boldsymbol{t}_o\|_2 \qquad \text{for all } \boldsymbol{t}_o \in \Pi_o \tag{22}$$

Similarly, define t_n^* as

$$\|\boldsymbol{p}^o - \boldsymbol{t}_n^*\|_2 \le \|\boldsymbol{p}^o - \boldsymbol{t}_n\|_2 \qquad \text{for all } \boldsymbol{t}_n \in \Pi_n \tag{23}$$

and $\boldsymbol{t}_\omega^*$ where

$$\|\boldsymbol{p}^o - \boldsymbol{t}_\omega^*\|_2 \le \|\boldsymbol{p}^o - \boldsymbol{t}_\omega\|_2 \qquad \text{for all } \boldsymbol{t}_\omega \in \Pi_\omega \tag{24}$$

Let

$$r = \inf_\omega r_\omega, \qquad 0 \le \omega < \infty \tag{25}$$

With these definitions we can state the following theorem.[8]

Theorem 1

Let $K(s)$ be a fixed stabilizing controller. Then the radius of the largest stability hypersphere in the parameter space of $\boldsymbol{p}$, centered at $\boldsymbol{p}^o$, is given by

$$\rho(\boldsymbol{p}^o) = \min(r_o, r_n, r) \tag{26}$$

Proof of this theorem is given in Biernacki.[8] This theorem states that the perturbation $\Delta\boldsymbol{p}$ of $\boldsymbol{p}^o$ cannot destabilize the system unless $\boldsymbol{p}^o + \Delta\boldsymbol{p}$ intersects Π_o, Π_n, or Π_ω. Intersection with Π_o results in a root at the origin, with Π_n a root at

$s = \infty$, and with Π_ω a pair of roots on the $j\omega$ axis, as ω sweeps from 0 to near infinity, all of which are destabilizing.

Let

$$\delta \in \Delta_o, \qquad t \in \Pi_o, \qquad w_1 = [1, 0, \dots 0, 0]^T \tag{27}$$

Then

$$w_1^T \delta = \delta_o = 0 = w_1^T X p = w_1^T X t \tag{28}$$

Denote the first row of X as X_f. Then Eq. (28) states that

$$X_f t = 0 \tag{29}$$

This says that the parameter vectors t contained in Π_o are perpendicular to the first row of X. Thus, the shortest distance to Π_o must lie along this direction, and is given by

$$p^o - t_o^* = \gamma X_f^T \tag{30}$$

where γ is a constant. To compute γ, premultiply Eq. (30) by X_f. Thus,

$$X_f p^o - \cancelto{0}{X_f t_o^*} = \gamma X_f X_f^T \tag{31}$$

so that

$$\gamma = X_f p^o / X_f X_f^T \tag{32}$$

Substituting this into Eq. (21) results in

$$r_o^2 = (1/X_f X_f^T)[p^{oT} X_f X_f^T p^o] \tag{33}$$

This r_o is the minimum distance from p^o to the hyperplane Π_o.

Using similar manipulations, the distance r_n is given by

$$r_n^2 = (1/X_l X_l^T)[p^{oT} X_l X_l^T p^o] \tag{34}$$

where X_l is the last row of X. Equations (33) and (34) give the Euclidean distance measures of p^o to Π_o and Π_n. The remaining distance measure is r_ω as ω is varied from zero to a sufficiently large value, with the minimum r given by Eq. (25). For a CLCV δ in Π_ω, δ is given by

$$\delta = \Phi(\omega)\xi \tag{35}$$

where ξ is of dimension $(n-1) \times 1$ and is arbitrary. The $(n+1) \times (n-1)$ matrix $\Phi(\omega)$ is given by

$$\Phi(\omega) = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ \omega^2 & 0 & & \\ & \omega^2 & \cdots & 1 \\ & & & 0 \\ & & & \omega^2 \end{bmatrix} \tag{36}$$

$\Phi(\omega)$ takes the $(n-1)$-dimensional Hurwitz polynomial (written as a vector) and multiplies into it two zeros located at $\pm j\omega$. The δ generated by $\Phi(\omega)\xi$ is arbitrary except for the fact that it contains two zeros along the $j\omega$ axis. Thus, if $t_\omega \in \Pi_\omega$ then t_ω has two zeros on the $j\omega$ axis. Then,

$$X t_\omega = \Phi(\omega)\xi \tag{37}$$

Partition X and t_ω as follows:

$$X = [X_I \quad X_J] \tag{38a}$$

$$t_\omega = \begin{bmatrix} t_I \\ t_J \end{bmatrix} \tag{38b}$$

where X_I is nonsingular. The construction of X_I so that it is nonsingular requires interchanging columns of X, thus reordering parameters in p. From Eqs. (37) and (38)

$$X_I t_I = \Phi(\omega)\xi - X_J t_J \tag{39}$$

$$t_I = X_I^{-1}\Phi(\omega)\xi - X_I^{-1} X_J t_J \tag{40}$$

Then, every $t_\omega \in \Pi_\omega$ is given by

$$t_\omega = \underbrace{\begin{bmatrix} X_I^{-1}\Phi(\omega) & -X_J t_J \\ 0 & I \end{bmatrix}}_{P(\omega)} \underbrace{\begin{bmatrix} \xi \\ t_J \end{bmatrix}}_{\xi} \tag{41}$$

where $P(\omega)$ is a constant real matrix for each ω and ξ_t is an arbitrary real vector. If we let ξ_t sweep over all real vectors, then the mapping in Eq. (41) will yield all $t_\omega \in \Pi_\omega$. The distance measure r_ω is the distance from t_ω to p^o, that is

$$t_\omega - p^o = P(\omega)\xi - p^o \tag{42}$$

with

$$\|t_\omega - p^o\|_2^2 = p^{oT} p^o - 2\xi_t^T P^T(\omega) p^o + \xi_t^T P^T(\omega) P(\omega) \xi_t \tag{43}$$

To find the ξ_t that minimizes this distance (the closest to p^o) in Eq. (43), for a fixed ω, the gradient of Eq. (43) with respect to ξ_t is computed and equated to zero. Doing so yields

$$\xi_t^* = (P^T(\omega)P(\omega))^{-1} P^T(\omega) p^o \tag{44}$$

with

$$r_\omega^2 = p^{oT}\Big(I - P(\omega)\big((P^T(\omega)P(\omega)\big)^{-1} P^T(\omega)\Big) p^o \tag{45}$$

The minimum distance r is given by

$$r = \inf_\omega r_\omega \tag{46}$$

Since all the parameters in Eq. (45) are known, r_ω can be computed vs frequency, and the minimum value over ω is computed [Eq. (46)]. This technique is now demonstrated.

From Eqs. (4) and (5), the nominal system $G(s) = N(s)d^{-1}(s)$ is given by

$$N(s) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} s^4 + \begin{bmatrix} 0 \\ 0 \end{bmatrix} s^3 + \begin{bmatrix} \omega^2 V Z_\delta \\ 0 \end{bmatrix} s^2 + \begin{bmatrix} 0 \\ \omega^2 M_\delta \end{bmatrix} s^1 + \begin{bmatrix} \omega^2 V(Z_\alpha M_\delta - Z_\delta M_\alpha) \\ \omega^2 (Z_\delta M_\alpha - Z_\alpha M_\delta) \end{bmatrix} s^0 \tag{47a}$$

$$= n_4 s^4 + n_3 s^3 + n_2 s^2 + n_1 s + n_0 \tag{47b}$$

$$d(s) = s^4 + (2\zeta\omega - Z_\alpha)s^3 + (\omega^2 - M_\alpha - 2\zeta\omega Z_\alpha)s^2 - (2\zeta\omega M_\alpha + \omega^2 Z_\alpha)s - \omega^2 M_\alpha \tag{48a}$$

$$= d_4 s^4 + d_3 s^3 + d_2 s^2 + d_1 s + d_0 \tag{48b}$$

The controller $K(s) = d_c^{-1}(s) N_c^T(s)$ is given by

$$N_c^T(s) = [K_a K_q \quad 0]s^2 + [K_a K_q (a_z + a_q) \quad K_q]s^1 + [K_a K_q a_z a_q \quad K_q a_q]s^0 \tag{49a}$$

$$= n_{c2}^T s^2 + n_{c1}^T s + n_{c0}^T \tag{49b}$$

$$d_c(s) = s^2 + 0 \cdot s^1 + 0 \cdot s^0 \tag{50a}$$

$$= d_{c2} s^2 + d_{c1} s + d_{c0} \tag{50b}$$

The parameter vector $\boldsymbol{p}$ is

$$\boldsymbol{p} = [\boldsymbol{n}_0^T, d_0, \boldsymbol{n}_1^T, d_1, \ldots, \boldsymbol{n}_4^T, d_4]^T \tag{51}$$

which is of dimension 15×1. The CLCP $\delta(s)$ is

$$\delta(s) = \delta_6 s^6 + \cdots + \delta_1 s + \delta_0$$

The CLCV $\boldsymbol{\delta}$ containing the above polynomial coefficients is of dimension 7×1. Removing the zero elements from Eq. (51), the following mapping is defined:

$$\underbrace{\left[\begin{array}{ccccccc|cc} & & & & & & d_{c2} & & \\ & & & & & d_{c2} & d_{c1} & & \\ & & & & d_{c2} & d_{c1} & d_{c0} & & n_{c12} \\ & & & d_{c2} & d_{c1} & d_{c0} & & n_{c22} & n_{c11} \\ n_{c12} & n_{c22} & d_{c2} & d_{c1} & d_{c0} & & & n_{c21} & n_{c10} \\ n_{c11} & n_{c21} & d_{c1} & d_{c0} & & & & n_{c20} & \\ n_{c10} & n_{c20} & d_{c0} & & & & & & \end{array}\right]}_{X_I \qquad\qquad\qquad X_J} \underbrace{\begin{bmatrix} n_{01} \\ n_{02} \\ d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ n_{12} \\ n_{21} \end{bmatrix}}_{p} = \boldsymbol{\delta} \tag{52}$$

Substituting numerical values into $\boldsymbol{p}^o$ [Eq. (52)] results in the nominal parameter vector listed in Table 1. The stability hypersphere radius $\rho(\boldsymbol{p}^o)$ is computed by evaluating Eq. (33) to compute r_o, Eq. (34) to compute r_n, and Eq. (45) to compute r_ω. Figure 3 displays the r_ω computation vs frequency. The stability hypersphere radius is computed using Eq. (26), which is

$$\rho(\boldsymbol{p}^o) = \min(1.6644 \times 10^9, 1.00, 18.38)$$

$$\rho(\boldsymbol{p}^o) = 1.00$$

These results show that

$$\|\Delta \boldsymbol{p}\|_2 < 1.0$$

for stability. The stability hypersphere radius is an upper bound on the 2 norm of the parameter perturbation vector $\Delta \boldsymbol{p}$. This bounds the square root of the sum of the squares of the elements of $\Delta \boldsymbol{p}$, where each Δp_i has different units. Our problem is to relate the stability hypersphere radius to individual parameter variation bounds. More specifically, to the aerodynamic stability derivatives. We see from Eq. (47) that several of the parameters in $\boldsymbol{p}$ are multilinear combinations of

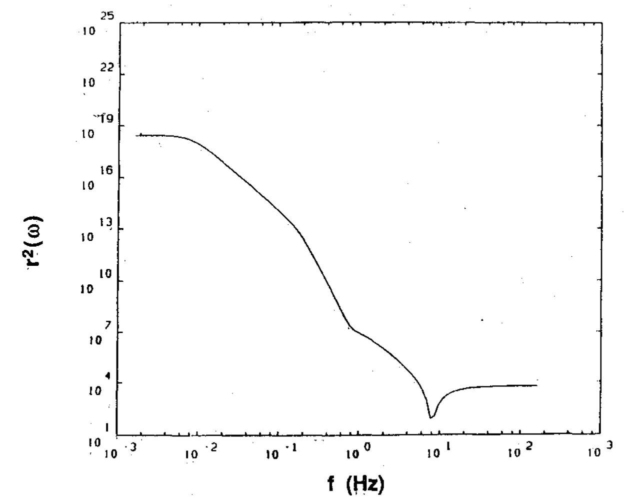


Fig. 3 Stability hypersphere radius linear mapping.

Table 1 Parameter variation bounds

	Nominal value	Allowable variation
n_{21}	-2.4259×10^6	$1.3741 \times 10^{-5}\%$
n_{12}	-1.3386×10^6	$2.4902 \times 10^{-5}\%$
d_4	1.0	33.33%
d_3	1.5950×10^2	$2.0899 \times 10^{-1}\%$
d_2	1.2928×10^4	$2.5784 \times 10^{-3}\%$
d_1	9.1110×10^3	$3.6583 \times 10^{-3}\%$
d_0	-6.0922×10^5	$5.4715 \times 10^{-5}\%$
n_{02}	-1.8769×10^6	$1.7760 \times 10^{-5}\%$
n_{01}	1.6644×10^9	$2.0276 \times 10^{-8}\%$

the aerodynamic stability derivatives. Thus, due to these nonlinearities, we can only relate the stability hypersphere radius back to the elements of $\boldsymbol{p}$, not to the aerodynamic stability derivatives themselves. We can divide this upper bound on $\Delta \boldsymbol{p}$ evenly among the $\ell = 9$ parameters contained in $\boldsymbol{p}$ as follows. Let

$$\|\Delta \boldsymbol{p}\|_2^2 = \sum_{i=1}^{\ell} \Delta p_i^2 = \ell(\overline{\Delta p})^2 = \rho^2(\boldsymbol{p}^o)$$

Then,

$$\overline{\Delta p} = \frac{\rho(\boldsymbol{p}^o)}{\sqrt{\ell}} = \frac{1}{3} = 0.3333$$

$\overline{\Delta p}$ represents the variation uniformly distributed over the nine parameters. If we divide this by the magnitude of each nominal parameter and multiply by 100% we obtain the upper bound on the percentage variation allowable in each parameter. Thus,

$$\%\text{ bound on } \Delta p_i = \frac{\rho(\boldsymbol{p}^o)}{\sqrt{\ell}} * \frac{100\%}{|p_i^o|} \tag{53}$$

Table 1 displays these bounds. The only reasonable uncertainty bound predicted is the 33.33% bound on d_4, which has a nominal value of 1.0. Because of the large magnitudes of the other polynomial coefficients, the allowable variation bounds are very small. This produces unusable results. We see from Eq. (48) that d_4 is not a function of the aerodynamic parameters. Thus, its magnitude is certain. Also, we are treating the actuator parameters ω and ζ and velocity V as known parameters. The mapping defined by Eq. (50) operates on the polynomial coefficients rather than, as we would like, the uncertain aerodynamic stability derivatives. It also does not discriminate between certain and uncertain parameters. This ultimately produces conservative predictions of robustness. This is a common problem when using polynomial robustness tests derived from Kharitonov's theorem. One would guess that by introducing scaling in the mapping [Eq. (52)], which nondimensionalizes each of the polynomial coefficients, more reasonable results would be obtained. Also, we could define an affine parameter mapping that would relate the uncertain parameters to the polynomial coefficients, thus removing parameters and/or constants that do not vary. In the next section both of these concepts are used to produce less conservative robustness predictions.

Stability Hypersphere Using an Affine Parameter Mapping

In the previous section, the stability hypersphere radius was computed by mapping the polynomial coefficients into the CLCV $\boldsymbol{\delta}$ space, i.e., $X\boldsymbol{p} = \boldsymbol{\delta}$. In this section, we introduce an affine parameter mapping that linearly describes the polynomial coefficients in terms of a smaller set of parameters. Let $\boldsymbol{a}$ be a vector of primary parameters that are uncertain. The polynomial coefficients in $\boldsymbol{p}$ depend affinely on $\boldsymbol{a}$ through the relation

$$\boldsymbol{p} = A\boldsymbol{a} + \boldsymbol{b} \tag{54}$$

with A a real matrix and $\boldsymbol{b}$ a real vector. Denote the nominal value of the uncertain parameter vector $\boldsymbol{a}$ as $\boldsymbol{a}^o$. The CLCV $\boldsymbol{\delta}$ using Eq. (54) is now

$$XA\boldsymbol{a} + X\boldsymbol{b} = \boldsymbol{\delta} \tag{55}$$

Equation (55) defines an affine transformation mapping the parameter space of $\boldsymbol{a}$ into the space of CLCV's $\boldsymbol{\delta}$. The sets Δ_o, Δ_n, and Δ_ω for Eqs. (15–17) are defined in a similar manner as before, with Π_o, Π_n, and Π_ω as their inverse images. Unlike before, some of these sets may be empty. Thus, the definition of r_o, r_n, and r_ω are modified to include:

$$r_o = \infty \quad \text{if} \quad \Pi_o = \emptyset \tag{56}$$

$$r_n = \infty \quad \text{if} \quad \Pi_n = \emptyset \tag{57}$$

$$r_\omega = \infty \quad \text{if} \quad \Pi_\omega = \emptyset \tag{58}$$

The stability hypersphere Theorem 1 is modified as follows[8]:

Theorem 2

Let $K(s)$ be a stabilizing controller. Then the radius of the largest stability hypersphere in the parameter space of $\boldsymbol{a}$, centered as $\boldsymbol{a}^o$, is given by

$$\rho(\boldsymbol{a}^o) = \min(r_o, r_n, r) \tag{59}$$

As before, we want to compute the distances r_o, r_n, and r_ω. Following Eqs. (28) and (29), $\boldsymbol{t} \in \Pi_o$ if and only if

$$X_f A\boldsymbol{t} + X_f\boldsymbol{b} = 0 \tag{60}$$

Equation (60) fails if $X_fA = 0$ and $X_f\boldsymbol{b} \neq 0$, which results in Π_o being empty. Thus, $r_o = \infty$. If $X_fA \neq 0$ then r_o is the positive square root of

$$r_o^2 = (1/X_f^TA^TAX_f)\left[\boldsymbol{a}^{oT}A^TX_fA\boldsymbol{a}^o + 2\boldsymbol{a}^{oT}A^TX_f^TX_f\boldsymbol{b} + \boldsymbol{b}^TX_f^TX_f\boldsymbol{b}\right] \tag{61}$$

Similarly, if $X_lA \neq 0$ then r_n is the positive square root of

$$r_n^2 = (1/X_l^TA^TAX_l)[\boldsymbol{a}^{oT}A^TX_lA\boldsymbol{a}^o + 2\boldsymbol{a}^{oT}A^TX_l^TX_l\boldsymbol{b} + \boldsymbol{b}^TX_l^TX_l\boldsymbol{b}] \tag{62}$$

If $X_lA = 0$ and if $X_l\boldsymbol{b} \neq 0$ then $r_n = \infty$. It is not possible for both $X_fA = 0$ and $X_f\boldsymbol{b} = 0$ (or using X_l) simultaneously if $K(s)$ stabilizes the nominal point $\boldsymbol{a}^o$.

Starting from Eq. (41) we have

$$\boldsymbol{t}_\omega = P(\omega)\xi_t = A\boldsymbol{t}_a(\omega) + \boldsymbol{b} \tag{63}$$

Partition the A matrix such that

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} t_a(\omega) = \begin{bmatrix} P_1(\omega) \\ P_2(\omega) \end{bmatrix} \xi_t - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{64}$$

where A_1 is invertible. Thus,

$$A_1\boldsymbol{t}_a(\omega) = P_1(\omega)\xi_t - \boldsymbol{b}_1 \tag{65}$$

$$A_2\boldsymbol{t}_a(\omega) = P_2(\omega)\xi_t - \boldsymbol{b}_2 \tag{66}$$

Solve for $t_a(\omega)$ using Eq. (65):

$$t_a(\omega) = A_1^{-1}P_1(\omega)\xi_t - A_1^{-1}\boldsymbol{b}_1$$

Substitute into Eq. (66)

$$A_2A_1^{-1}P_1(\omega)\xi_t - A_2A_1^{-1}b_1 = P_2(\omega)\xi_t - b_2$$

$$\underbrace{\left[A_2A_1^{-1}P_1(\omega) - P_2(\omega)\right]}_{B(\omega)}\xi_t = \underbrace{A_2A_1^{-1}b_1 - b_2}_{c}$$

$$B(\omega)\xi_t = \boldsymbol{c} \tag{67}$$

Let the solution of Eq. (67) be of the form

$$\xi_t = D(\omega)\bar{\xi}_t + \boldsymbol{e}(\omega) \tag{68}$$

Then

$$t_a(\omega) = A_1^{-1}P_1(\omega)D(\omega)\bar{\xi}_t + A_1^{-1}P_1(\omega)\boldsymbol{e}(\omega) - A_1^{-1}b_1 \tag{69a}$$

$$= \tilde{P}(\omega)\bar{\xi}_t + \boldsymbol{\tau}(\omega) \tag{69b}$$

where

$$\tilde{P}(\omega) = A_1^{-1}P_1(\omega)D(\omega) \tag{70}$$

$$\boldsymbol{\tau}(\omega) = A_1^{-1}P_1(\omega)\boldsymbol{e}(\omega) - A_1^{-1}\boldsymbol{b}_1 \tag{71}$$

For additional convenience we will drop the ω dependence in Eqs. (70) and (71). The distance from a vector in Π_ω to $\boldsymbol{a}^o$ is

$$\|\boldsymbol{t}_a(\omega) - \boldsymbol{a}^o\|_2^2 = (\boldsymbol{t}_a(\omega) - \boldsymbol{a}^o)^T(\boldsymbol{t}_a(\omega) - \boldsymbol{a}^o)$$
$$= (\tilde{P}\bar{\xi}_t + \tau - \boldsymbol{a}^o)^T(\tilde{P}\bar{\xi}_t + \tau - \boldsymbol{a}^o)$$
$$= \bar{\xi}_t^T\tilde{P}^T\tilde{P}\bar{\xi}_t + 2\bar{\xi}_t\tilde{P}^T\tau - 2\bar{\xi}_t\tilde{P}^T\boldsymbol{a}^o - 2\tau^T\boldsymbol{a}^o + \tau^T\tau + \boldsymbol{a}^{oT}\boldsymbol{a}^o \tag{72}$$

We want to find the smallest distance between $\boldsymbol{t}_a(\omega)$ and $\boldsymbol{a}^o$. This is done by sweeping $\bar{\xi}_t$ over all real vectors. Equation (72) expresses this distance in terms of $\bar{\xi}_t$. To find the smallest distance, we compute the gradient of Eq. (72) with respect to $\bar{\xi}_t$ and equate to zero:

$$\frac{\partial(\cdot)}{\partial\bar{\xi}_t} = 2\tilde{P}^T\tilde{P}\bar{\xi}_t + 2\tilde{P}^T\tau - 2\tilde{P}^T\bar{a}^o = 0 \tag{73}$$

Thus,

$$\bar{\xi}_t^* = (\tilde{P}^T\tilde{P})^{-1}\tilde{P}^T(\boldsymbol{a}^o - \tau) \tag{74}$$

Substituting Eq. (74) into Eq. (72) yields the minimum r_ω^2 as a function of frequency:

$$r_\omega^2 = (\boldsymbol{a}^o - \tau)^T\tilde{Q}(\boldsymbol{a}^o - \tau) \tag{75}$$

$$\tilde{Q} = I - \tilde{P}(\omega)(\tilde{P}^T(\omega)\tilde{P}(\omega))^{-1}\tilde{P}^T(\omega) \tag{76}$$

As before, r in Eq. (59) is given by

$$r = \inf_\omega r_\omega$$

where r_ω is given by the positive square root of Eq. (76).

There are two requirements that must be met in order to implement the algorithm of the previous section. They are 1) Partition $X = [X_I \quad X_J]$ such that X_I is invertible, and 2) Partition $A = [A_1 \quad A_2]^T$ such that A_1 is invertible subject to

$$X\boldsymbol{p}^o = XA\boldsymbol{a}^o + X\boldsymbol{b} = \boldsymbol{\delta} \tag{77}$$

The parameter mapping obtained is as follows:

$$\boldsymbol{p} = A\boldsymbol{a} + \boldsymbol{b} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}\boldsymbol{a} + \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} \tag{78}$$

$$\begin{bmatrix} n_{01} \\ d_0 \\ n_{12} \\ d_3 \\ n_{21} \\ n_{02} \\ d_4 \\ d_2 \\ d_1 \end{bmatrix} = \left[\begin{array}{ccccc} \omega^2 V & 0 & 0 & 0 & 0 \\ 0 & -\omega^2 & 0 & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & \omega^2 V \\ \hline -\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -2\zeta\omega & 0 \\ 0 & -2\zeta\omega & 0 & -\omega^2 & 0 \end{array}\right]$$

$$\times \begin{bmatrix} Z_\alpha M_\delta - Z_\delta M_\alpha \\ M_\alpha \\ M_\delta \\ Z_\alpha \\ Z_\delta \end{bmatrix} + \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 2\zeta\omega \\ 0 \\ \hline 0 \\ 1 \\ \omega^2 \\ 0 \end{array}\right]$$

The rows of the X matrix, Eq. (52), are reordered to match the polynomial coefficient ordering in Eq. (78). This structure produces both X_I and A_1 invertible. Our goal is to compute the stability hypersphere radius in the parameter space of $\boldsymbol{a}$, given by Eq. (78). By introducing scaling, we can nondimensionalize the parameters and can compute the stability hyperellipsoid radius that will give better bounds on the individual parameter variations. Consider

$$\boldsymbol{a} = Q\tilde{\boldsymbol{a}} \tag{79}$$

Now $\tilde{\boldsymbol{a}}$ is a vector containing ones and the Q matrix contains the inverse of the nominal parameter magnitudes along its diagonal. Clearly, the mapping introduced by Eq. (79) maps the set of all hyperspheres in the $\tilde{\boldsymbol{a}}$ space into the hyperellipsoids in the $\boldsymbol{a}$ space, and the mapping is one to one. Thus, the largest hyperellipsoid in the $\boldsymbol{a}$ space can be found by computing the largest stability hypersphere in the $\tilde{\boldsymbol{a}}$ space. Thus,

$$\boldsymbol{p} = \tilde{A}\tilde{\boldsymbol{a}} + \boldsymbol{b} \tag{80}$$

where

$$\tilde{A} = AQ \tag{81}$$

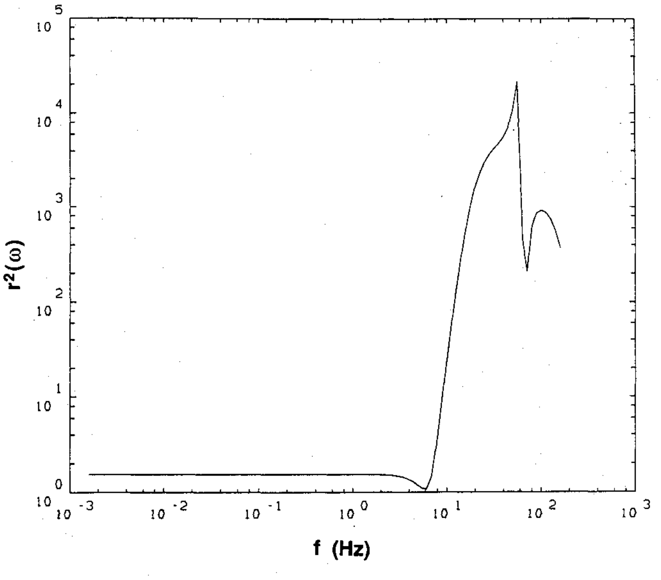


Fig. 4 Stability hypersphere radius affine parameter mapping.

To compute the stability hypersphere radius we use the algorithm outlined in the previous section but with Eq. (81) instead of A. We then map the stability hypersphere in the $\tilde{\boldsymbol{a}}$ space to the stability hyperellipsoid in the $\boldsymbol{a}$ space. Performing these calculations results in:

$$\tilde{\rho}(\boldsymbol{p}^o) = \min(1.0, \infty, 1.044)$$

This is the stability hypersphere radius for the normalized parameters, with r_ω vs frequency shown in Fig. 4. The stability hypersphere radius for the normalized parameters is 1.0. Evenly distributing this among the five parameters results in

$$\%\text{ bound on } \Delta p_i = \frac{\tilde{\rho}(\boldsymbol{p}^o)}{\sqrt{l}} * 100\% = 44.7\%$$

Scaling this bound back into the $\boldsymbol{a}$ parameter space results in the following parameter tolerances:

$$Z_\alpha M_\delta - Z_\delta M_\alpha = 147.0 \pm 65.71$$
$$M_\alpha = 47.71 \pm 21.33$$
$$M_\delta = -104.8 \pm 46.55$$
$$Z_\alpha = -1.305 \pm 0.5833$$
$$Z_\delta = -0.2142 \pm 0.0957$$

The parameter variation bounds using the affine parameter mapping and the stability hyperellipsoid approach are greatly improved over the previous section results. A small amount of conservatism will be present in the above parameter variation bounds due to the first parameter being a multilinear combination of the remaining four parameters.

Stability Hypersphere Radius Calculation Using Lyapunov Approach

The previous two sections used polynomial based theorems to compute the upper bound (using a 2 norm) on the allowable parameter variation $\Delta\boldsymbol{p}$. This section computes this same upper bound derived from a state space model of the closed-loop system. Consider the state space realizations of $K(s)$ and $G(s)$. The closed-loop system, in state space form, is

$$\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \underbrace{\begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{bmatrix}}_{M} \begin{bmatrix} x \\ x_c \end{bmatrix}$$

$$= \left[\underbrace{\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} D_c & C_c \\ B_c & A_c \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}}_{C} \right] \begin{bmatrix} x \\ x_c \end{bmatrix} \tag{82}$$

The closed-loop system will be stable if and only if $M = A + BKC$ is stable. Let

$$\boldsymbol{p} = [p_1 \cdots p\ell]^T$$

denote uncertain parameters in the plant triple (A, B, C), which enter into the closed-loop matrix M in a linear fashion.

Two models may be used in describing how the parameter uncertainties enter the closed-loop system matrix. The first model, as in the polynomial case, models the parameter vector as

$$\boldsymbol{p} = \boldsymbol{p}^o + \Delta\boldsymbol{p} \tag{83}$$

with $\boldsymbol{p}^o$ the nominal parameter vector and $\Delta\boldsymbol{p}$ the uncertainties. With this model $\Delta\boldsymbol{p}$ has the same units as $\boldsymbol{p}$. The second model uses a multiplicative uncertainty model given by

$$\boldsymbol{p} = \boldsymbol{p}^o(1 + \Delta\boldsymbol{p}) \tag{84}$$

where the variation Δp is dimensionless and each element is bounded by unity. Since the stability hypersphere radius bounds the 2 norm of Δp, the units on the individual elements of Δp influence the degree of conservatism of the bound [Eq. (53)]. Results for both these models are presented.

The controller parameters in (A_c, B_c, C_c, D_c) are known and do not vary. (It is posible to treat the controller parameters as the uncertain variables but this is not considered here.) The closed-loop system can be modeled as

$$A + BKC = A^o + B^oKC^o + \sum_{i=1}^{\ell} E_i \Delta p_i$$

where (A^o, B^o, C^o) represent the nominal plant description (using p^o), which is stabilized by the controller $K(s)$. The matrices E_i are known real matrices that are the structural definitions for the uncertain parameters p_i. Let $M_o = A^o + B^oKC^o$. M_o is stable. The closed-loop uncertain system is

$$\dot{x} = Mx \tag{85}$$

$$M = M_o + \sum_{i=1}^{\ell} E_i \Delta p_i \tag{86}$$

Since the closed-loop linear time invariant system is stable, we have, for a positive definite Q,

$$M_o^TP + PM_o + Q = 0 \tag{87}$$

where P exists uniquely and is positive definite symmetric. Equation (87) is the Lyapunov equation for linear time invariant systems. We want to analyze the uncertain closed-loop system. To do so we will form the Lyapunov function

$$V(x) = x^TPx \tag{88}$$

where P is the solution matrix to Eq. (87). Using this $V(x)$ the closed-loop uncertain system will be stable if and only if

$$V(x) > 0 \quad \text{for all } x \tag{89a}$$

$$\dot{V}(x) \le 0 \quad \text{for all } x \tag{89b}$$

The first condition is satisfied for $V(x)$ given by Eq. (88) since P is the positive definite solution of Eq. (87). The second condition is used to form an upper bound on the Δp_i. To do this we must determine how large the Δp_i can be such that $\dot{V}(x) \le 0$. The derivative of $V(x)$ is given by

$$\dot{V}(x) = \dot{x}^TPx + x^TP\dot{x} \tag{90}$$

Substitute for x using Eqs. (85) and (86)

$$\dot{V}(x) = x^T\left(M_o + \sum_{i=1}^{\ell} E_i \Delta p_i\right)^T Px + x^TP\left(M_o + \sum_{i=1}^{\ell} E_i \Delta p_i\right)x$$

From Eq. (87)

$$\dot{V}(x) = x^T\underbrace{(M_o^TP + PM_o)}_{-Q}x + x^TP\left[\sum_{i=1}^{\ell} (E_iP^T + PE_i)\Delta p_i\right]x$$

$$\dot{V}(x) = -x^TQx + x^T\left[\sum_{i=1}^{\ell} (E_i^TP + PE_i)\Delta p_i\right]x \tag{91}$$

We must show that

$$-x^TQx + x^T\left[\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right]x \le 0$$

or

$$x^T\left[\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right]x \le x^TQx \tag{92}$$

We know that

$$\underline{\sigma}[Q] \le \frac{x^TQx}{x^Tx} \le \bar{\sigma}[Q] \qquad \text{(Rayleigh principle)}$$

$$\underline{\sigma}[Q]x^Tx \le x^TQx \tag{93}$$

Substitute Eq. (93) into Eq. (92)

$$x^T\left[\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right]x \le \underline{\sigma}[Q]x^Tx \tag{94}$$

Also

$$\left|x^T\left[\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right]x\right|$$

$$\leqslant \|x^T\|_2\left\|\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right\|_2\|x\|_2$$

$$\|x^T\|_2\left\|\sum_{i=1}^{\ell}(E_i^TP + PE_i)\Delta p_i\right\|_2\|x\|_2$$

$$\le \|x\|_2^2\left(\sum_{i=1}^{\ell}|\Delta p_i|\,\|E_i^TP + PE_i\|_2\right) \tag{95}$$

The steps in Eq. (95) are used to separate out the $|\Delta p_i|$. (Our goal is to isolate the perturbation.) However, in separating these norms, conservatism is introduced. The degree of conservatism depends upon how the perturbation effects the closed-loop system, that is on the structure of E_i. Substituting Eq. (95) in Eq. (94) results in

$$\|x\|_2^2\left(\sum_{i=1}^{\ell}|\Delta p_i|\,\|E_i^TP + PE_i\|_2\right) \le \underline{\sigma}[Q]x^Tx \tag{96}$$

Thus,

$$\sum_{i=1}^{\ell}|\Delta p_i|\,\| + E_i^TP + PE_i\|_2 \le \underline{\sigma}[Q] \tag{97}$$

Let

$$\eta_i = \|E_i^TP + PE_i\|_2 = \bar{\sigma}[\,\cdot\,]$$

$$\sum_{i=1}^{\ell}|\Delta p_i|\,\|E_i^TP + PE_i\|_2 = \underbrace{[|\Delta p_1| \cdots |\Delta p\ell|]}_{\gamma}\begin{bmatrix}\eta_1\\ \vdots\\ \eta_\ell\end{bmatrix} \le \sigma[Q] \tag{98}$$

In vector form, we can write Eq. (98) as

$$\gamma\eta \le \underline{\sigma}[Q]$$

where γ is a row vector containing the magnitudes of the Δp_i. Thus,

$$\|\gamma\eta\|_2^2 \le \|\gamma\|_2^2\|\eta\|_2^2 \le \underline{\sigma}^2[Q] \tag{99}$$

Unfortunately, Eq. (99) adds more conservatism. Expanding these norms results in

$$\|\gamma\|_2^2 = \sum_{i=1}^{\ell}|\Delta p_i|^2$$

$$\|\eta\|_2^2 = \sum_{i=1}^{\ell}\eta_i^2$$

Table 2 Stability hypersphere radius using Lyapunov optimization

Parameter variation model	Stability hypersphere radius, $\rho(p^o, Q)$	Allowable variation
$p = p^o + \Delta p$	1.6208×10^{-4}	0.007%
$p = p^o(1 + \Delta p)$	4.5107×10^{-4}	0.02%

Then

$$\sum_{i=1}^{\ell} |\Delta p_i|^2 \leq \frac{\underline{\sigma}^2[Q]}{\sum_{i=1}^{\ell} \eta_i^2} \tag{100}$$

Equation (100) gives an expression for the stability hypersphere radius squared. Denote the stability hypersphere radius as $\rho(p^o, Q)$. Equation (100) gives $\rho^2(p^o, Q)$ as a ratio of the minimum singular value of the Q matrix squared to the sum of the maximum singular value of the matrices $E_i^T P + PE_i$. Thus, ρ depends upon the choice of the matrix Q, and the nominal system parameters p^o, which together form the Lyapunov solution matrix P in Eq. (87).

It should be clear that the "size" of the matrix Q in Eq. (87) defines the solution matrix P [Eq. (87)], which determines $\rho(p^o, Q)$. Given a controller $K(s)$ we desire to know the largest stability hypersphere radius about our nominal parameter vector p^o. Thus, an optimization problem is formed. The function to be maximized is $\rho(p^o, Q)$. Our problem is now

$$\max_Q \rho^2(p^o, Q) = \frac{\underline{\sigma}^2[Q]}{\sum_{i=1}^{\ell} \|E_i^T P + PE_i\|_2^2} \tag{101}$$

Unfortunately, little is known about the geometry of $\rho(p^o, Q)$. If $\rho(p^o, Q)$ were convex, then there would be a unique solution to Eq. (101) (our results show that it is not). In order to solve Eq. (101), a conjugate gradient optimization algorithm is employed. The optimization routines, in fortran, were taken from Ref. 25. In order to implement the conjugate gradient optimization algorithm the gradient of Eq. (101) is required. Both analytical implementation and a numerical approximation of the gradient were used. The Chapter 5 Appendix in Ref. 9 contains the analytical derivation of the gradient.

In the optimization problem of Eq. (101), the Q matrix is factored as $Q = L^T L$, and the optimization is performed over the matrix L. This guarantees a positive definite matrix Q, as required by Eq. (87).

Using Eq. (86) the closed-loop M matrix is

$$M = \begin{bmatrix} Z_\alpha & 1 & Z_\delta & 0 & 0 & 0 \\ M_\alpha & 0 & M_\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\omega^2 K_a K_q V Z_\alpha & -\omega^2 K_q & -\omega^2(1 + K_a K_q V Z_\delta) & -2\zeta\omega & \omega^2 K_q & \omega^2 \\ -K_a a_z V Z_\alpha & 0 & -K_a a_z V Z_\delta & 0 & 0 & 0 \\ -K_a K_q a_q V Z_\alpha & -K_q a_q & -K_a K_q a_q V Z_\delta & 0 & K_q a_q & 0 \end{bmatrix} \tag{102}$$

The uncertain parameters are

$$p = [Z_\alpha \; Z_\delta \; M_\alpha \; M_\delta]^T \tag{103}$$

The closed-loop matrix is modeled as

$$M = M_o + E_1 \Delta Z_\alpha + E_2 \Delta Z_\delta + E_3 \Delta M_\alpha + E_4 \Delta M_\delta \tag{104}$$

Using this approach there are only four parameters. The previous parameter that was a multilinear combination of aerodynamic stability derivatives only appears in the CLCP. Thus, this approach offers fewer parameters. The uncertain parameters in Eq. (103) are modeled using Eqs. (83) and (84).

The matrix M_o is given by Eq. (102) using the nominal parameter vector p^o. The uncertainties enter the closed-loop system through the matrices E_i. When using state space models to analyze parameter uncertainties, the rank of the matrix E_i is used to describe the perturbation.

In Bhattacharyya,[9] the Q matrix Lyapunov optimization problem was applied to a helicopter design problem. We used this same problem to test our Lyapunov optimization software. When starting at the same initial condition, our software determined a smaller minimum. Bhattacharyya[9] adjusted both the L matrix and controller parameters to create a design that satisfied a target $\|\Delta p\|_2$. Our results, using his initial controller parameters, determined a $Q = L^T L$ that satisfied his target $\|\Delta p\|_2$ without adjusting the controller parameters. The initial condition selected for the optimization significantly impacts the results. This indicates that this function, Eq. (100), is nonconvex.

Table 2 displays the results of the stability hypersphere radius calculation using Lyapunov optimization for both parameter variation models [Eqs. (83) and (84)]. These very small bounds indicate extreme conservatism in the computation of the stability hypersphere radius. This conservatism results from the steps in Eqs. (95) and (97) that were used to separate out the uncertainty parameter magnitudes.

Conclusions

There are several robustness theories available that can analyze control system sensitivity to real parameter variations. However, many of these theories are very conservative. The stability hypersphere theory, using a polynomial method derived from Kharitonov's theorem and a state space method derived from the Lyapunov stability theory, were applied here determining a missile pitch acceleration command autopilot sensitivity to uncertain aerodynamic stability derivatives. The results using this theory indicate that a 44.7% variation in the aerodynamic stability derivatives can occur with the closed-loop system remaining stable. Of the methods evaluated in this paper, the stability hypersphere radius calculation using the affine parameter mapping generated the least conservative prediction.

To test the conservatism using these approaches, this same problem was solved using a Monte Carlo eigenanalysis. The eigenanalysis predicted a 61% allowable parameter variation. This same 61% bound was predicted by the DeGaston-Safonov multiloop stability margin algorithm. Thus, the stability hypersphere approach is conservative in its prediction of stability robustness.

References

[1]Kharitinov, V. L., "Asymptotic Stability of an Equilibrium Position of a Family of Systems of Linear Differential Equations," *Differential. Uravnen.*, Vol. 14, No. 11, 1978, pp. 2086–2088.

[2]Barmish, B. R., "Invariance of the Strict Hurwitz Property for Polynomials with Perturbed Coefficients," *IEEE Transactions on Automatic Control*, Vol. AC-29, No. 10, 1984, pp. 935–936.

[3]Bialas, S., and Garloff, J., "Stability of Polynomials under Coefficient Perturbation," *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 3, 1985, pp. 310–312.

[4]Yeung, K. S., and Wang, S. S., "A Simple Proof of Kharitinov's

Theorem," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 9, 1987, pp. 822–823.

[5]Wei, K. H., and Yedavalli, R. K., "Invariance of Strict Hurwitz Property for Uncertain Polynomials with Dependent Coefficients," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 10, 1987, pp. 907–909.

[6]Anderson, B. D. O., Jury, E. I., and Mansou, M., "On Robust Hurwitz Polynomials," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 10, 1987, pp. 909–913.

[7]Argoun, M. B., "Frequency Domain Conditions for the Stability of Perturbed Polynomials," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 10, 1987, pp. 913–916.

[8]Biernacki, R. M., Hwang, H., and Bhattacharyya, S. P., "Robust Stability with Structured Real Parameter Perturbations," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 6, 1987, pp. 495–506.

[9]Bhattacharyya, S. P., *Robust Stabilization Against Structured Perturbations*, Springer Verlag, New York, 1987.

[10]Martin, J. M., "State Space Measures for Robust Stability," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 6, 1987, pp. 509–512.

[11]Hewer, G., and Kenney, C., "The Sensitivity of the Stable Lyapunov Equation," *Journal of Control and Optimization*, Vol. 26, No. 2, 1988, pp. 321–343.

[12]Wise, K. A., "Singular Value Robustness Tests for Missile Autopilot Uncertainties," *Journal of Guidance, Control, and Dynamics* (to be published); also, AIAA Paper 89-3552, Aug. 1989.

[13]Doyle, J. C. "Structured Uncertainty in Control System Design," *Proceedings of the IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1985, pp. 260–265.

[14]Doyle, J. C., Wall, J. E., and Stein, G., "Performance and Robustness Analysis for Structured Uncertainty," *Proceedings of the 22nd IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1982, pp. 629–636.

[15]Morton, B. G., and McAfoos, R. M., "A Mu-Test for Robustness Analysis of a Real-Parameter Variation Problem," *Proceedings of the American Control Conference*, American Automatic Control Council, May 1985.

[16]Morton, B. G., "New Applications of Mu to Real Parameter Variation Problems," *Proceedings of the 25th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1985, pp. 233–238.

[17]Jones, R. D., "Structured Singular Value Analysis for Real Parameter Variations," AIAA Paper 87-2589, 1987.

[18]Zadeh, L. A., and Desoer, C. A., *Linear System Theory*, McGraw-Hill, New York, 1963, Chap. 11, Theorem 21, p. 476.

[19]DeGaston, R. R., and Safonov, M., "Exact Calculation of the Multiloop Stability Margin," *IEEE Transactions on Automatic Control*, Vol. 33, No. 2, 1988, pp. 156–171.

[20]Pena, R., and Sideris, A., "A General Program to Compute the Multivariable Stability Margin for Systems with Parametric Uncertainty," *Proceedings of the American Control Conference*, 1988, pp. 317–322.

[21]Sideris, A., and Pena, R., "Robustness Margin Calculation with Dynamic and Real Parametric Uncertainty," *Proceedings of the American Control Conference*, 1988, pp. 1201–1206.

[22]Sideris, A., and Pena, R., "Fast Computation of the Multivariable Stability Margin for Real Interrelated Uncertain Parameters," *Proceedings of the American Control Conference*, 1988, pp. 1483–1488.

[23]Fu, M., Olbrot, A. W., and Polis, M. P., "Introduction to the Parametric Approach to Robust Stability," *Control Systems Magazine*, Vol. 9, No. 5, 1989, pp. 7–11.

[24]Wise, K. A., "A Comparison of Six Robustness Tests Evaluating Missile Autopilot Robustness to Uncertain Aerodynamics," *Proceedings of the American Control Conference*, American Automatic Control Council, May 1990, pp. 755–763.

[25]Press, W., Flanney, B., Teykolsky, S., and Vetting, W., *Numerical Recipes*, Cambridge Univ. Press, Cambridge, England, UK, 1986.