JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 4, July–August 1997

Numerically Efficient Robustness Analysis of Trajectory Tracking for Nonlinear Systems

Jorge E. Tierno*
Honeywell Technology Center, Minneapolis, Minnesota 55418
Richard M. Murray† and John C. Doyle‡
California Institute of Technology, Pasadena, California 91125
and
Irene M. Gregory§
NASA Langley Research Center, Hampton, Virginia 23681

A numerical algorithm for computing necessary conditions for performance specifications is developed for nonlinear uncertain systems that follow a prescribed trajectory. This algorithm provides a computationally efficient means of evaluating the performance of a nonlinear system in the presence of noise, real parametric uncertainty, and unmodeled dynamics. The algorithm is similar in nature and behavior to the power algorithm for the structured singular value (μ) lower bound and does not rely on a descent method. The algorithm is tested on a flight control example.

I. Introduction

THE engineering motivation behind the theory of robust control is based on two unavoidable facts. First, all analysis and synthesis methods are based on the use of models. In most cases, it is impractical to try the designs in prototypes at the early stages. Furthermore, critical control systems must have a reasonable expectation of good performance before they are implemented because their failure can lead to significant losses. These factors force us to analyze systems extensively using models before proceeding to the implementation stage. Second, models are, by nature, incomplete and inaccurate. They are incomplete because they are deduced from a finite observation of the system. They are inaccurate because the observation is noisy and, even more important, because we are trying to project a physical system into a particular class of mathematical models we believe to be representative, e.g., the class of linear, finite-dimensional models.

A. Review of Previous Work

Theoretical and computational tools for analysis and synthesis of robust controllers for linear systems are well developed in a variety of instances. Controllers generated with these tools can provide guaranteed performance in the presence of structured uncertainty, and the worst-case disturbances for a given controller can be determined.

There are several possible choices of performance measures and uncertainty sets for which we can determine robustness. For linear time invariant (LTI) systems with structured uncertainty, analysis of robust performance can be reduced to computing a single scalar function of the system known as the structured singular value and denoted μ (Ref. 1). Although this function cannot be computed exactly, we are able to find computationally efficient algorithms to compute upper and lower bounds, such as the power algorithm for the μ lower bound,[2,3] without doing an explicit parameter search involving repeated simulation. This works because the system is linear and the performance and uncertainty descriptions are chosen so as to give computationally attractive solutions, even for large problems.

Robustness analysis of nonlinear systems, on the other hand, has stayed mainly at the theoretical level. Given the diversity of behavior in nonlinear systems, there are many branches in nonlinear robustness theory. Several methods have been proposed for studying the stability robustness of nonlinear systems and interconnections (see Refs. 4 and 5 and references therein). Also, extensive research has been carried out in the area of controller design that achieve robust stability and tracking (see Refs. 6–9 and references therein). These results are based on sufficient conditions only and do not provide a measure of their conservativeness. The work in this paper complements these results by developing computable necessary conditions for some tracking problems.

A large body of work has been devoted to extending the μ framework of linear robust performance analysis techniques to nonlinear systems. As part of this work, conditions for robust stability and robust performance similar to the structured singular value have been established. Robustness for nonlinear systems was proved to be equivalent to the existence of solutions to Hamilton–Jacobi equations[10] or nonlinear matrix inequalities.[11] However, computational methods to establish the existence of these solutions have not been developed to a level comparable to their linear counterpart, i.e., solutions of Riccati equations and linear matrix inequalities.

The state of the art in industry still consists of obtaining lower bounds to the performance indices through extensive simulation or local optimization techniques. In particular, probabilistic Monte Carlo methods are widely used to analyze the influence of external disturbances such as wind gusts[12] and in many cases are part of the certification guidelines.[13] This kind of Monte Carlo analysis consists of nonlinear simulation of the system model, subject to disturbance signals chosen at random. From the results obtained, we can infer typical behavior of the system and perform covariance analysis. Hammersley and Handscomb[14] is a good reference for various Monte Carlo methods and their applications.

B. Contribution of This Paper

The intent of the work described in this paper is to extend the robustness analysis techniques of linear systems, in particular, the associated computational methods, to nonlinear systems. Given the diversity of nonlinear behavior, this clearly cannot be done in complete generality and still maintain the efficiency and usability of the methods. We develop analysis methods for a specific nonlinear robust performance problem. We show that an efficient numerical algorithm can be developed to compute a lower bound on the corresponding performance index. The problem and the corresponding algorithm share the characteristics of current industrial practice mentioned earlier.

The problem we study is that of tracking a trajectory in the presence of noise and uncertainty. Many nonlinear analysis problems of

Received Jan. 16, 1996; revision received Dec. 19, 1996; accepted for publication March 7, 1997.

*Senior Research Scientist, Department of Control and Navigation, 3660 Technology Drive. Member AIAA.

†Assistant Professor, Control and Dynamical Systems.

‡Professor, Control and Dynamical Systems.

§Research Engineer, Department of Flight Dynamics and Control.

engineering interest can be reduced to such a problem. A common example is an airplane performing an automatic change of altitude and heading. The pilot enters the new heading and altitude, and the flight computer determines nominal commands to perform it. A second control loop regulates the airplane around the nominal trajectory. The flying qualities specifications determine an appropriate path to be completed in a finite predetermined time and the control system is designed accordingly. As the real system is not exactly the one used for the design, and as it is also subject to noise, the system will not exactly follow the nominal commanded trajectory. The question of interest becomes: Will the real trajectory, under the worst conditions possible, remain close enough to the nominal one in an appropriate norm? We will call this question the robust trajectory tracking problem. (We give it a more formal definition in the next section.)

We develop a power algorithm to compute a lower bound on the performance index associated with the robust trajectory tracking problem, i.e., the distance from the actual to the nominal trajectory. This algorithm is similar to the one developed for the structured singular value[1,3] and has similar behavior. As was the case for linear systems, the algorithm is not guaranteed to converge in general, and so its analysis is done empirically. We test this algorithm by applying it to simulations of real systems. We carry out performance tests on a simplified model of an F-16 jet fighter. The results of these tests are reported in Sec. V. These results indicate that without significant additional computation, and avoiding computationally expensive parameter searches, a lower bound on the given performance index can be computed that gives information on the worst-case behavior of the system, complementary to the information on typical behavior given by the standard Monte Carlo procedures.

II. Problem Formulation

We are concerned with the study of perturbations of systems around a prespecified nominal trajectory over a finite time horizon with initial time t_i and final time t_f. The performance specifications and all characterizations of the noise signals, undermodeled components, and uncertain parameters affecting the system will be given over this time interval.

We restrict our study to nonlinear systems whose dynamics can be represented by a smooth function F of the states $\boldsymbol{x}$, a set of signals $\boldsymbol{u}$ corresponding to external disturbances, a set of signals $\boldsymbol{v}$ corresponding to the effect of undermodeled components, a set of parameters δ, and a set of nominal commands $\boldsymbol{U}$ responsible for steering the nominal system along the nominal trajectory. The evolution of the system, thus, is the solution to

$$\dot{\boldsymbol{x}} = F(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t)$$

with initial conditions $\boldsymbol{x} = \boldsymbol{x}_0$. The trajectory is described by a set of coordinates $\boldsymbol{Y}$, given by a smooth function g of the same variables,

$$\boldsymbol{Y} = G(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t)$$

The nominal trajectory Y_n, thus, is given by

$$\boldsymbol{Y}_n = G(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{0}, \delta_n, \boldsymbol{U}, t)$$

where δ_n are the nominal values for the parameters.

The signals $\boldsymbol{v}$ are not independent of the other signals, but the exact dependence is not known (hence, the name undermodeled dynamics). We know, however, that $\boldsymbol{v}$ is the image under a structured, norm-bounded operator of a set of variables z, given as a smooth function of the other variables in the problem

$$z = H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t)$$

We denote Δ the mapping from $\boldsymbol{z}$ to $\boldsymbol{u}$, $\boldsymbol{\Delta}$ the class of operators with same block diagonal structure, and $\boldsymbol{B\Delta}$ the unit ball in $\boldsymbol{\Delta}$ with respect to the L_2 into L_2 induced norm. Because we analyze the system for a fixed nominal trajectory, we can incorporate the nominal commands into the functions F, G, and H and define

$$f_U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, t) = F(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t)$$

$$h_U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, t) = H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t)$$

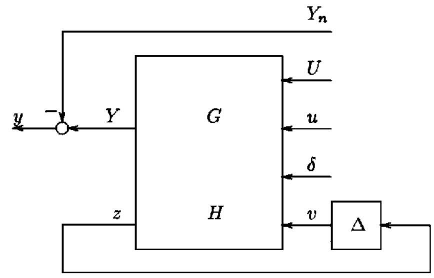

Fig. 1 Uncertain system interconnection.

As we are only interested in the error between the nominal trajectory and the actual one, we also define

$$g_U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, t) = G(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{U}, t) - \boldsymbol{Y}_n$$

Whenever no confusion is possible, we drop the subscript U. The structure of the operator Δ we consider is block diagonal and, thus, will induce a partition in the sets of signals $\boldsymbol{v}$ and z,

$$\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p] \qquad z = [z_1, z_2, \ldots, z_p]$$

Note that $\boldsymbol{v}$ and $\boldsymbol{z}$ are divided into the same number of blocks of signals, but correspondent blocks do not have to have the same number of signals. The size of the noise signals entering the problem also requires the introduction of a partition of the signal set $\boldsymbol{u}$

$$\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m]$$

Figure 1 shows a diagram of the system interconnection. There are several similarities with the linear μ framework[1] and some differences. The undermodeled dynamical components are feedback loops, as is the case in linear systems; however, the effect of the parameters on the system can be more general. The dynamics are not only nonlinear but are also nonautonomous: the system includes explicit functions of time.

We measure the sizes of all signals in the two-norm defined as usual over a finite time horizon:

$$\|\boldsymbol{a}\|_2 = \left[\frac{1}{t_f - t_i}\int_{t_i}^{t_f} \boldsymbol{a}^t \boldsymbol{a}\,\mathrm{d}t\right]^{\frac{1}{2}}$$

The performance index is given by the two-norm of the error signal

$$J = \|y\|_2$$

We consider the case where noise and uncertainty achieve their maximum values. (This can be done without loss of generality; see Remark.) The size of the noise signals will be specified for each block of signals in the partition given

$$\|\boldsymbol{u}_i\|_2 = N_i \qquad i = 1, 2, \ldots, m \tag{1}$$

The only information we have on the undermodeled dynamical components is that Δ is in $\boldsymbol{B\Delta}$. We use the induced two-norm as a measure of the size of Δ. This restriction is then equivalent to imposing the following relations between the signals $\boldsymbol{v}$ and $\boldsymbol{z}$:

$$\|\boldsymbol{v}_i\|_2 = \|z_i\|_2 \qquad i = 1, 2, \ldots, p \tag{2}$$

We allow the parameters δ to vary in closed intervals:

$$d_i \le \delta_i \le D_i \qquad i = 1, 2, \ldots, r$$

Finally, we also allow some or all of the initial conditions to vary in given closed intervals:

$$c_i \le x_i(t_i) \le C_i \qquad i = 1, 2, \ldots, n$$

Remark: By adding one parameter δ_{in}, the equality restrictions in Eqs. (1) and (2) can be converted into inequalities. Define, for example, a new output

$$\tilde{z}_i = \delta_{in} z_i$$

Then

$$\|\boldsymbol{v}_i\| \le z_i \Leftrightarrow \begin{cases} \|\boldsymbol{v}_i\| = \|\tilde{z}_i\| \\ |\delta_{in}| \le 1 \end{cases}$$

The preceding performance, noise, and uncertainty descriptions given in this section can be summarized as a constrained optimization problem. As we only use one norm, we will drop the subscript 2 to keep the notation unencumbered. We now state the following problem.

Problem (Robust Trajectory Tracking): Given an uncertain nonlinear system, driven by the equation

$$\dot{\boldsymbol{x}} = f_U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, t)$$

and the nominal command signal U, with initial conditions satisfying

$$c_i \le x_i(t_i) \le C_i \qquad i = 1, 2, \ldots, n$$

what is the maximum value of the norm of the error signal

$$\|\boldsymbol{y}\| = \|g_U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\delta}, t)\|$$

subject to the constraints

$$\|\boldsymbol{u}_i\| = N_i \qquad i = 1, 2, \ldots, m$$

$$\|\boldsymbol{v}_i\| = \|z_i\| \qquad i = 1, 2, \ldots, p$$

$$d_i \le \delta_i \le D_i \qquad i = 1, 2, \ldots, r$$

This problem may seem too specific to be of any consequence to real-life applications. When analyzing linear systems, we use different kinds of multiplicative weighting functions (weights) to impose a frequency content to noise signals, to determine the bandwidth of an uncertain operator, or to determine the region in the frequency domain over which we would like to impose the performance specification. Use of these weights makes the linear analysis techniques more flexible and is generally fundamental to establishing the validity of the method. With similar techniques we can reduce many interesting application problems to a particular instance of the Problem.

III. Necessary Conditions

We begin by establishing conditions characterizing local maxima. First-order conditions for extrema of dynamical systems have been developed for different optimization indices and signal constraints (for example, see Ref. 15 or 16). We derive the necessary conditions for a local maximum of the Problem from the Euler–Lagrange optimization setup. In this section, we first review briefly the standard Euler–Lagrange optimization framework for dynamical systems and show how the robust trajectory tracking problem reduces to an instance of the general Euler–Lagrange problem.

The following theorem summarizes the Euler–Lagrange framework.

Theorem 1 (Ref. 16): For a dynamical system described by the equations

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) \qquad x(0) \text{ given} \qquad t_i \le t \le t_f$$

a performance index of the form

$$J = \int_{t_i}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t)\, \mathrm{d}t$$

and restrictions on the final state

$$G[\boldsymbol{x}(t_f)] = \boldsymbol{c}$$

where f and L have continuous partial derivatives with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$ and certain regularity conditions are met (see Ref. 16), if the signal $\boldsymbol{u}_0$ achieves an extremum of J, then there exists a vector of constants $\boldsymbol{\zeta}$ and a solution to the two-point boundary value problem

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t)$$

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial f}{\partial \boldsymbol{x}}\right)^t \boldsymbol{\lambda} - \left(\frac{\partial L}{\partial \boldsymbol{x}}\right)^t \tag{3}$$

$$0 = \left(\frac{\partial L}{\partial \boldsymbol{u}}\right) + \left(\frac{\partial f}{\partial \boldsymbol{u}}\right)^t \boldsymbol{\lambda} \tag{4}$$

with boundary conditions

$$x(0) \text{ given} \qquad \boldsymbol{\lambda}(t_f) = \left[\frac{\partial G}{\partial \boldsymbol{x}(t_f)}\right]^t \boldsymbol{\zeta}$$

Furthermore, if these conditions are met, we will have

$$\boldsymbol{\lambda}(t_i) = \frac{\partial J}{\partial \boldsymbol{x}(0)} \tag{5}$$

This theorem states necessary conditions for a set of signals being a first-order extremum of the performance index, in terms of the existence of a solution of an associated adjoint system equation (3). Furthermore, the solutions of the original and adjoint system verify the constraint equation (4). We will see that this constraint can be interpreted as an alignment condition between the inputs of the original nonlinear system and a set of outputs of the adjoint one. This structure will be exploited when we develop the power algorithm.

A. Reduction to the Euler–Lagrange Setup

To use the Euler–Lagrange necessary conditions in the robust trajectory tracking problem, we have to write the performance index, noise signals, uncertain parameters, and undermodeled dynamical components of the system as constraints compatible with the hypothesis of Theorem 1.

1. *Performance Index*

The performance index we described is very naturally written in the form required by Theorem 1. Letting

$$L = \tfrac{1}{2}\boldsymbol{y}^t\boldsymbol{y}$$

then optimizing $\|\boldsymbol{y}\|$ is equivalent to optimizing

$$J = \int_{t_i}^{t_f} L\, \mathrm{d}t =: \frac{1}{2}\|\boldsymbol{y}\|^2$$

2. *Noise Signals*

The only constraints allowed by Theorem 1 are in the final values of states. The norm restrictions for the noise signals and the uncertain operator then have to be imposed through final conditions of additional states created for that purpose. We will describe the case for one noise signal only; however, the generalization to several signals is obtained simply by repeating the single-signal case. To impose the two-norm condition on the noise signals, we add to the system a new state named x_u governed by the differential equation

$$\dot{x}_u = \tfrac{1}{2}\boldsymbol{u}^t\boldsymbol{u} \qquad x_u(t_i) = 0$$

Then $\|\boldsymbol{u}\| = N$ if and only if $x_u(t_f) = \frac{1}{2}N$.

3. *Undermodeled Dynamical Component*

The undermodeled dynamical block is characterized by a constraint on the norms of its input signals $\boldsymbol{v}$ and output signals $\boldsymbol{z}$:

$$\|\boldsymbol{v}\| = \|\boldsymbol{z}\|$$

To impose this equality, we add to the system a state x_Δ, governed by the differential equation

$$\dot{x}_\Delta = \tfrac{1}{2}(\boldsymbol{z}^t\boldsymbol{z} - \boldsymbol{v}^t\boldsymbol{v}) \qquad x_\Delta(t_f) = 0$$

Then $\|\boldsymbol{v}\| = \|\boldsymbol{z}\|$ if and only if $x_\Delta(t_f) = 0$.

B. Uncertain Parameters and Uncertain Initial Conditions

Optimality with respect to uncertain initial conditions or uncertain parameters will be established through the gradient condition given by Eq. (5). To treat parameters as initial conditions, we create a state that tracks the parameter δ. Let x_δ follow the equation

$$\dot{x}_\delta = 0 \qquad x_\delta(t_i) = \delta$$

At a local maximum, either this derivative of the performance index with respect to the value of the parameter or the state initial condition is zero, or it is negative and the parameter is at the lower end of the interval, or it is positive and the parameter is at the higher end of the interval.

C. Summary

Summarizing, the robust trajectory tracking problem is equivalent to optimizing the performance index

$$J = \int_{t_i}^{t_f} L \,\mathrm{d}t = \frac{1}{2}\|\boldsymbol{y}\|^2$$

for the system verifying the differential equation for dynamics

$$\dot{x} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, x_\delta)$$

noise constraint

$$\dot{x}_u = \tfrac{1}{2}\boldsymbol{u}^t\boldsymbol{u}$$

uncertainty constraint

$$\dot{x}_\Delta = \tfrac{1}{2}(z^t z - \boldsymbol{v}^t\boldsymbol{v})$$

and uncertain parameter

$$\dot{x}_\delta = 0$$

where the performance output

$$\boldsymbol{y} = g(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, x_\delta)$$

and the uncertainty output

$$z = h(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, x_\delta)$$

with given initial conditions

$$\boldsymbol{x}(t_i) = \boldsymbol{x}_0 \qquad x_u(t_i) = 0 \qquad x_\Delta(t_i) = 0 \qquad x_\delta(t_i) = \delta$$

and final conditions

$$x_u(t_f) = \tfrac{1}{2}, \qquad x_\Delta(t_f) = 0$$

This problem is in the form required by Theorem 1. A set of signals u and v and a parameter δ achieve the worst-case value of the performance index J only if there exists $\boldsymbol{\lambda}_x$, λ_u, λ_Δ, and $\boldsymbol{\lambda}_\delta$, verifying

$$\begin{aligned}
\dot{\boldsymbol{\lambda}} &= -\left(\frac{\partial f}{\partial \boldsymbol{x}}\right)^t \boldsymbol{\lambda} - \left(\frac{\partial h}{\partial \boldsymbol{x}}\right)^t z\lambda_\Delta - \left(\frac{\partial g}{\partial \boldsymbol{x}}\right)^t \boldsymbol{y} \\
\dot{\lambda}_\delta &= -\left(\frac{\partial f}{\partial \boldsymbol{\delta}}\right)^t \boldsymbol{\lambda} - \left(\frac{\partial h}{\partial \boldsymbol{\delta}}\right)^t z\lambda_\Delta - \left(\frac{\partial g}{\partial \boldsymbol{\delta}}\right)^t \boldsymbol{y} \\
\dot{\lambda}_u &= 0 \qquad \dot{\lambda}_\Delta = 0
\end{aligned} \tag{6}$$

with final state conditions

$$\boldsymbol{\lambda}(t_f) = 0 \qquad \lambda_\delta(t_f) = 0$$

verifying the following alignment conditions:

$$\begin{aligned}
&\left(\frac{\partial f}{\partial \boldsymbol{u}}\right)^t \boldsymbol{\lambda} + \boldsymbol{u}\boldsymbol{\lambda}_u + \left(\frac{\partial h}{\partial \boldsymbol{u}}\right)^t z\boldsymbol{\lambda}_\Delta + \left(\frac{\partial g}{\partial \boldsymbol{u}}\right)^t \boldsymbol{y} = 0 \\
&\left(\frac{\partial f}{\partial \boldsymbol{v}}\right)^t \boldsymbol{\lambda} + \left[\left(\frac{\partial h}{\partial \boldsymbol{v}}\right)^t z - \boldsymbol{v}\right]\lambda_\Delta + \left(\frac{\partial g}{\partial \boldsymbol{v}}\right)^t \boldsymbol{y} = 0
\end{aligned} \tag{7}$$

and such that the initial state verifies

$$\lambda_\delta(t_i) = 0, \quad \text{or} \quad \begin{cases} \delta = -1 \\ \text{and} \\ \lambda_\delta(t_i) < 0 \end{cases} \quad \text{or} \quad \begin{cases} \delta = 1 \\ \text{and} \\ \lambda_\delta(t_i) > 0 \end{cases} \tag{8}$$

The nature of the performance index chosen and the fact that none of the functions f, g, or h depend explicitly on the additional states x_u and x_Δ give Eqs. (6) and (7) a particular structure. The states λ_u and λ_Δ will be constants, and the choice of the two-norm as a performance measure is reflected by the fact that the adjoint system dynamics are driven by the performance output of the system. In the following sections, we will make this structure clearer by introducing convenient notation, and we will show how a natural powerlike iteration can be derived from it.

D. Dynamical Systems Interpretation

For a given trajectory of the original nonlinear system, define the time varying matrices

$$A(t) = \begin{bmatrix} \left(\frac{\partial f}{\partial x}\right)^t & 0 \\ \left(\frac{\partial f}{\partial \delta}\right)^t & 0 \end{bmatrix} \qquad B(t) = \begin{bmatrix} \left(\frac{\partial g}{\partial x}\right)^t & \left(\frac{\partial h}{\partial x}\right)^t \\ \left(\frac{\partial g}{\partial \delta}\right)^t & \left(\frac{\partial h}{\partial \delta}\right)^t \end{bmatrix}$$

$$C(t) = \begin{bmatrix} \left(\frac{\partial f}{\partial u}\right)^t & 0 \\ \left(\frac{\partial f}{\partial v}\right)^t & 0 \end{bmatrix} \qquad D(t) = \begin{bmatrix} \left(\frac{\partial g}{\partial u}\right)^t & \left(\frac{\partial h}{\partial u}\right)^t \\ \left(\frac{\partial g}{\partial v}\right)^t & \left(\frac{\partial h}{\partial v}\right)^t \end{bmatrix}$$

and consider the dynamical system driven by the equations

$$-\dot{\boldsymbol{\Lambda}} = A\boldsymbol{\Lambda} + B\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\xi} \end{bmatrix} \qquad \begin{bmatrix} -\boldsymbol{\gamma} \\ \boldsymbol{\kappa} \end{bmatrix} := C\boldsymbol{\Lambda} + D\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\xi} \end{bmatrix} \tag{9}$$

with zero final conditions. Then, if the signals u and v and the value of the parameter δ achieve a local maximum, there exist constants λ_u and λ_Δ such that the following alignment conditions are verified between inputs and outputs of the direct and adjoint dynamical systems:

$$\boldsymbol{\nu} = \boldsymbol{y} \qquad \boldsymbol{\xi} = \lambda_\Delta z \tag{10}$$

and

$$\lambda_u \boldsymbol{u} = \boldsymbol{\gamma} \qquad \lambda_\Delta \boldsymbol{v} = \boldsymbol{\kappa} \tag{11}$$

Equation (8) states that at an optimum either the derivative of the performance index with respect to the value of the parameter is zero, or it is negative and the parameter is at the lower end of the interval, or it is positive and the parameter is at the higher end of the interval.

This interconnection structure is represented in Fig. 2. Note that although the two-point boundary value problem has both initial and final state conditions, in this case, they separate into two sets. The

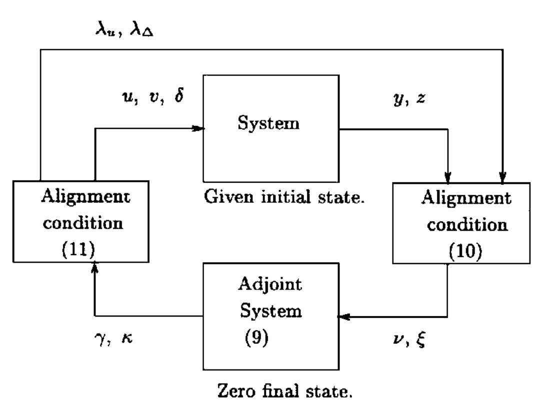

Fig. 2 Dynamical systems interpretation of the equilibrium conditions.

initial conditions are imposed on the states of the original nonlinear system, and the final conditions are imposed on the states of the adjoint system. It is this particular structure of the two-point boundary value problem that will allow us to develop a power algorithm to solve it.

IV. Power Algorithm

Power algorithms have been used successfully to compute, among other things, fixed points and local maximums of nonlinear functions and eigenvalues and singular values of matrices. The main advantage of power algorithms over other iterative methods is the simplicity of each iteration. Iterations usually consist of little more than a function evaluation. In the case of eigenvalue computations, iterations usually only require computing a matrix vector product. Another important characteristic of power methods is their tendency to explore a large region of the search space in their early stages. When applied to computing lower bounds for the linear performance index μ, power algorithms have proven to be fast and reliable. The robust trajectory tracking problem is very similar in nature to the structured singular value problem. [When applied to linear systems, the robust trajectory tracking problem actually reduces to a special case of μ (Ref. 17).] This motivates the development of a powerlike algorithm to compute a lower bound for the Problem. In this section, we first give a qualitative description of power algorithms when applied to matrix problems; then we briefly describe the application of the standard μ power algorithm to linear systems over a finite time horizon; finally, we show how these ideas can be extended in an elegant way to the general nonlinear problem.

A. Power Algorithms for Linear Matrices

Power algorithms are normally used to compute eigenvalues and singular values of matrices. Suppose that the matrix M is diagonalizable and that its largest eigenvalue is unique. Starting from a random point v_0, the power algorithm produces a sequence as follows[18]:

$$z^{(k)} := M\boldsymbol{\nu}^{(k-1)} \qquad \lambda^{(k)} := \left\| z^{(k)} \right\|_2 \qquad \boldsymbol{\nu}^{(k)} := z^{(k)} / \lambda^{(k)}$$

Let x_i be the eigenvectors of the matrix M. If v_0 has a component along the maximum eigenvalue direction

$$\boldsymbol{\nu}^{(0)} = a_1\boldsymbol{x}_1 + \sum_{i=2}^{n} a_i\boldsymbol{x}_i$$

it follows that

$$M^k\boldsymbol{\nu}^{(0)} = a_1\lambda_1^k\left[\boldsymbol{x}_1 + \sum_{i=2}^{n} \frac{a_i}{a_1}\left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{x}_j\right]$$

If

$$\lambda_i / \lambda_1 < 1 \tag{12}$$

the component of $\boldsymbol{\nu}^{(0)}$ along the maximum eigenvector direction is amplified, whereas the others are contracted, and the procedure converges to the maximum eigenvalue. The rate of convergence is determined by the ratios in Eq. (12). Because singular values of M are eigenvectors of M^*M, the same procedure can be used to compute the maximum singular value. In this case we will have to perform alternating power steps, by M and M^*. The power algorithm is most useful when we only need to compute a few of the matrix eigenvalues and when the separation in magnitude between the eigenvalues is large. For a more complete discussion of the power algorithm as applied to eigenvalue computation, see Refs. 18 and 19.

Power algorithms are naturally suited to searching for directions of maximum gain, a fact that makes them appealing to the computation of performance measures in disturbance rejection settings. Computing the structured singular value, for example, implies computing the largest of the spectral radii of the matrices in a set. There are two search directions: one within the set of matrices to find the one that has the largest spectral radius and one for a given matrix to find its eigendirection of maximum gain. The power algorithm for μ does both searches simultaneously. From a qualitative viewpoint, the power steps, alternating multiplications by M and M^*, help look for directions of maximum gain of the particular matrix under study. The alignment conditions forced between power steps perform the search in the matrix set. A discrete time linear system, considered over a finite time horizon, can be represented by the matrix of the map between inputs and outputs. Many robust performance questions asked of this type of system can be solved by computing the structured singular value of the map matrix with respect to an adequate structure.[17] Note that, in this case, the power step with respect to M is taken by integrating the difference equations of the system forward in time and the power step by M^* is computed by simulating an adjoint dynamical system backward in time. The succession of simulations of linear systems amplifies the component the chosen input has along the direction of maximum gain, whereas the operations carried out between power steps look for the system in the class with maximum gain.

Conceptually, all of the operations necessary to carry out this procedure are still well defined when the system is nonlinear. The substitutions needed to convert one case into the other are natural. Multiplication by the system matrix is equivalent to integration of the equations of motion. Multiplication by the adjoint matrix is equivalent to integration of the transposed linearized system. Although numerically more difficult, the operations are not conceptually different from the linear case and, in principle, the heuristics explaining why this type of algorithm is efficient at finding directions of maximum gain still hold, even when the system is nonlinear. In the next section, we prove that these substitutions are the adequate ones, and from them we develop a power algorithm to compute a lower bound for the nonlinear robust trajectory tracking problem.

B. Power Algorithm for Nonlinear Systems

In the preceding section, we showed how the first-order necessary conditions for robust performance could be interpreted as an interconnection of dynamical systems. Four maps can be identified in the diagram of Fig. 2. The first map integrates the equations of motion of the original system, with the given initial conditions and for a set of input signals u, v. The constants λ_u and λ_Δ are not changed. (These are included in the definition of the map to simplify the final expression.) Thus,

$$\Phi : [\boldsymbol{u}(t), \boldsymbol{v}(t), \delta, \lambda_u, \lambda_\Delta] \mapsto [\boldsymbol{y}(t), z(t), \delta, \lambda_u, \lambda_\Delta]$$

The position of the map Φ in the system interconnection corresponding to the necessary conditions is shown in Fig. 3a.

The second map generates the input signals for the adjoint system by forcing the alignment conditions in Eq. (10):

$$\Pi : [\boldsymbol{y}(t), z(t), \delta, \lambda_u, \lambda_\Delta] \mapsto [\boldsymbol{y}(t), \lambda_\Delta z(t), \lambda_u, \lambda_\Delta]$$

Figure 3b indicates the position of this map in the general interconnection.

The third map in the sequence integrates the equations for the adjoint system along the current trajectory with the given inputs:

$$\Psi : (\boldsymbol{\nu}, \boldsymbol{\xi}, \lambda_u, \lambda_\Delta) \mapsto (\gamma, \boldsymbol{\kappa}, \lambda_u, \lambda_\Delta)$$

The final map we define computes new inputs for the original system by using the alignment condition in Eq. (11) and also computes new values for δ, λ_u, and λ_Δ. There is more than one way of carrying out this evaluation, and we propose one possibility as follows:

$$\Theta : (\gamma, \boldsymbol{\kappa}, \lambda_u, \lambda_\Delta) \mapsto [\boldsymbol{u}(t), \boldsymbol{v}(t), \delta, \lambda_u, \lambda_\Delta]$$

$$\lambda_u := \frac{\|\gamma\|}{U_e}\frac{\gamma \cdot \boldsymbol{u}}{\|\gamma\| U_e} \qquad \boldsymbol{u} := \frac{\gamma}{\lambda_u}$$

$$\lambda_\Delta := \frac{\|\boldsymbol{\kappa}\|}{\|z\|}\frac{\boldsymbol{\kappa} \cdot \boldsymbol{v}}{\|\boldsymbol{\kappa}\|\|\boldsymbol{v}\|} \qquad \boldsymbol{v} := \frac{\boldsymbol{\kappa}}{\lambda_\Delta}$$

and δ is computed as follows:

$$\chi := \delta + \lambda_\delta(t_i) \qquad \delta := \begin{cases} -1 & \chi < -1 \\ \chi & -1 \le \chi \le 1 \\ 1 & \chi > 1 \end{cases}$$

Figures 3c and 3d show these maps relative to the necessary conditions interconnection.

The mappings Φ and Ψ are, as described in the preceding section, the nonlinear equivalent of power steps or matrix vector multiplications. The operations carried out in the mappings Π and Θ impose the alignment conditions for optimality. The diagram in Fig. 2 commutes when the signals $\boldsymbol{u}$ and $\boldsymbol{v}$ and the constants δ, λ_u, and λ_Δ are the ones corresponding to an extremum point. In other words, by going once around the diagram following the maps Φ, Π, Ψ, and Θ, we should return to the starting point. This is formalized in the following.

Lemma 2: The signals u and v and the parameter δ verify the optimality conditions if and only if there exists λ_u, λ_Δ such that

$$(\boldsymbol{u}, \boldsymbol{v}, \delta, \lambda_u, \lambda_\Delta)$$

is a fixed point of the composition $\Theta \circ \Psi \circ \Pi \circ \Phi$.

We can now use an iterative algorithm to search for the fixed points of this composition. The standard form of such an algorithm is given by the pseudocode in Table 1.

If the algorithm converges, it converges to a fixed point of the composition $\Theta \circ \Psi \circ \Pi \circ \Phi$ and, thus, according to Lemma 2, to a set of signals that meet the necessary conditions for a critical point.

Remarks: To prove convergence, we would have to prove that the composition $\Theta \circ \Psi \circ \Pi \circ \Phi$ is a contraction around local maximums. We can only prove this under very limiting conditions. The standard power algorithm for the lower bound of μ has not been proven to be stable for a general uncertainty structure and, in fact, is known to be unstable in some cases.[20] Hence, the evaluation of the algorithm will have to be done empirically. This is a major drawback of this approach, as we cannot guarantee the stability of the algorithm, nor can we provide a test of convergence. However, although the same limitations are present, in the linear case, the power algorithm has proven to be very successful[20]; our experience in the nonlinear case with several examples has been positive.[21]

The algorithm developed has many of the characteristics of the power algorithm for linear systems. In particular, if the system is linear, the adjoint system is LTI, and in this case, the algorithm reduces to the standard power iteration alternating multiplication by M and M^*.

The iteration presented is not the only one possible. In particular, it is not necessary to update the adjoint system every iteration. The adjoint can be computed once every n iterations, for example. The effect of this modification depends on the characteristics of the nonlinear dynamics, the size of the disturbances, and the particular trajectory under consideration. In principle, it could be harmful or beneficial. As the behavior of the nonlinear system is a lot more diverse than in the linear case, it is important to tailor the analysis tools to the particular problem at hand. An advantage of the power algorithm is that, due to its simplicity, it easily accepts modifications that adapt it to the application being considered. Given the wide diversity in the behavior of nonlinear systems, it is fair to say that no one optimization algorithm will work in all cases. This makes comparison between algorithms hard. It also makes it difficult to prove that any given algorithm is suitable for a general purpose. The performance of the algorithms must be evaluated on a case-by-case basis. Nonlinear systems, however, can still be classified in families of problems. All aircraft models, for example, are similar, and variations in the possible behaviors are more quantitative than qualitative in nature. Evaluation of the algorithm can be done on test cases for classes of problems. These examples have to be chosen so as to contain all of the important characteristics of their class.

Table 1 Power algorithm pseudocode

```
x^(1) := random;
repeat
    x^(i+1) := Θ ∘ Ψ ∘ Π ∘ Φ(x^(i));
until (|x^(i+1) − x^(i)| < ε|x^(i)|)
```

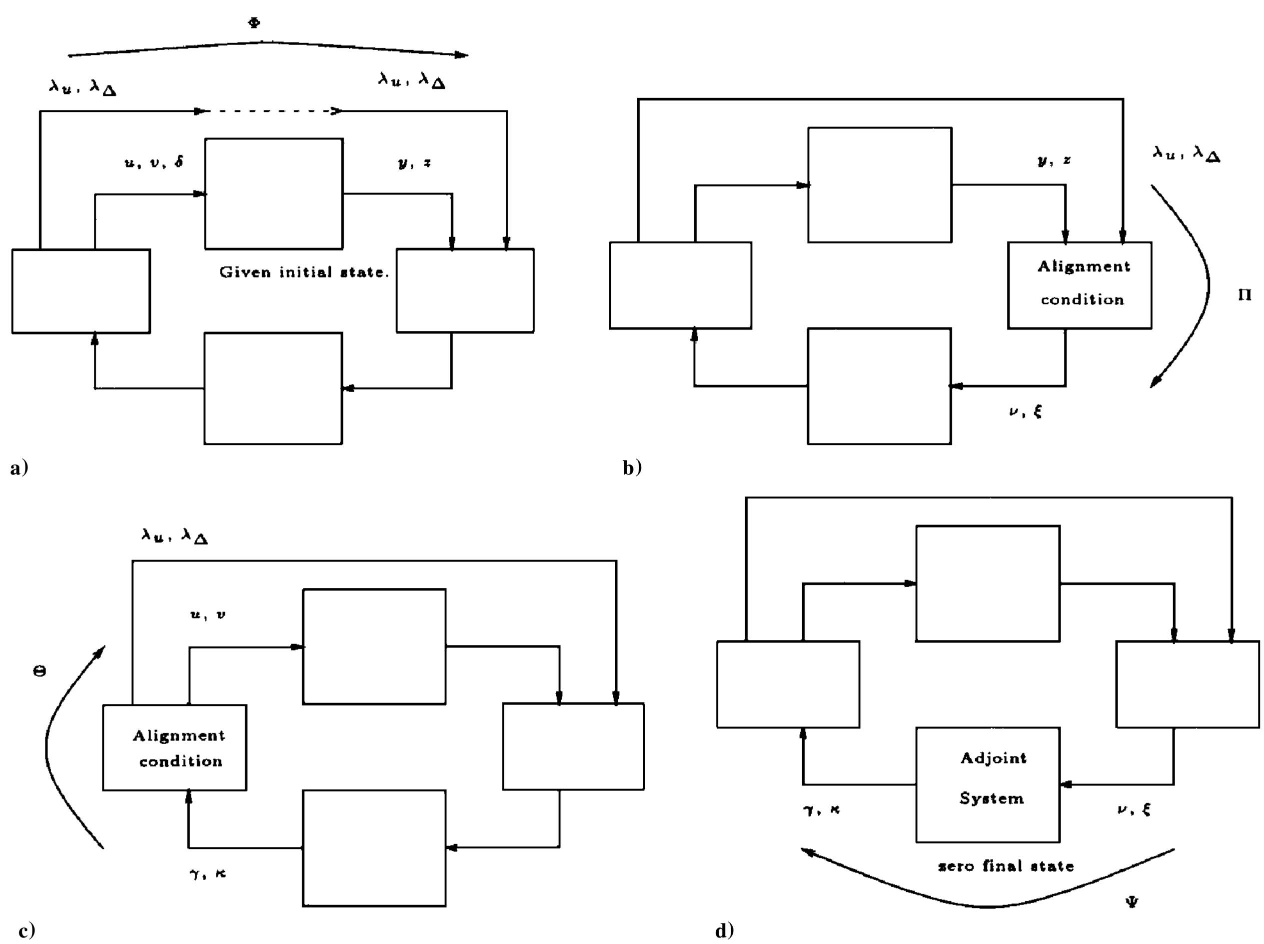

Fig. 3 Maps Φ, Π, Θ, and Ψ.

V. Numerical Example: F-16 Maneuver

In this section, we present the results obtained from trying the power algorithm presented in Sec. IV. The system studied is nonlinear; parts of the model are obtained from first principles (equations of motion of rigid bodies), and parts of the model are obtained through measurements and implemented with look-up tables. The model includes command and rate saturations. This system, and the performance problem we set up for it, has many of the characteristics of typical aircraft (or other types of vehicles).

The results obtained prove the viability of the algorithm. Further evaluation of its behavior and its applicability to real-life problems can only be obtained through extensive use in industrial applications.

We want to determine whether the algorithm is suitable for aerospace applications. As a first step, the algorithm's ability to handle a model that includes a number of nonlinear equations and tabular data with a relatively high number of parameters, all characteristic of a typical aircraft, must be ascertained.

The aircraft used in this example application is an F-16. The aerodynamic model is a reduced version of the full model obtained in wind-tunnel tests at NASA Langley Research Center in 1979 (Ref. 22). It consists of tabular data with typical interpolation routines and nonlinear equations of motion. The engine model is that of an after-burning turbofan. The airplane model used in this application is defined for speed range of up to Mach 0.6 and angle-of-attack interval between −10 and 45 deg. The model includes four traditional controls (elevator, aileron, rudder, and throttle) and 13 states (velocity vector, attitude angles, angular velocities, navigational position, altitude, and engine power). Furthermore, the aerodynamic coefficients are built up in a traditional way, and the equations of motion are full nonlinear flat-Earth equations.

The trajectory considered is a constant-climb, constant-g coordinated turn. The aircraft initiates the maneuver at 10,000 ft flying at 500 ft/s. The F-16 is then trimmed to climb at 50 ft/s while maintaining a 4.5-g coordinated turn. We study the maneuver over a 30-s interval. Figure 4 illustrates the nominal trajectory.

During the maneuver, the aircraft is subjected to atmospheric turbulence in vertical, horizontal, and lateral directions modeled by von Karman spectra and implemented by Dryden filters.[13] In addition, seven parameters in the model are allowed to vary individually on a closed interval. These parameters include variation in c.g. position as well as uncertainty in the aerodynamic forces and moments along each axis. For the example presented here, the numerical values for the variations are shown in Table 2. The bounds given for the center of mass are relative to the mean aerodynamic chord. The bounds given for the aerodynamic forces and moments are as a fraction of the value given by the look-up tables.

The algorithm is asked to find the combination of parameters and wind gusts that produces the largest norm of the performance variable vector, i.e., turning radius and altitude error. The worst-case combination produced by the algorithm gives the value of each of the parameters at the endpoint of the allowable interval of variation. Table 3 summarizes the values found for the parameters.

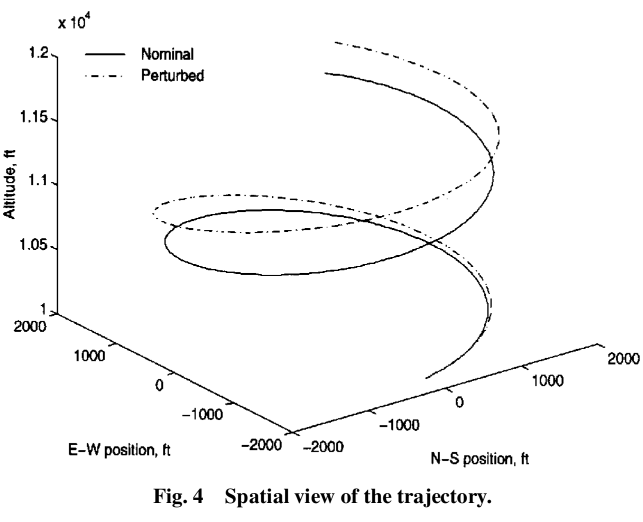

Fig. 4 Spatial view of the trajectory.

Table 2 Allowed variations for the uncertain parameters

Parameter	Name	Lower bound	Upper bound
Center of mass	c.g.	$0.195\bar{c}$	$0.205\bar{c}$
oX force	Cx	0.975	1.025
oY force	Cy	0.985	1.015
oZ force	Cz	0.97	1.03
Rolling moment	Cl	0.95	1.05
Pitching moment	Cm	0.95	1.05
Yawing moment	Cn	0.95	1.05

Table 3 Final values for the uncertain parameters

Parameter	Value
Cg	$0.195\bar{c}$
Cx	1.025
Cy	0.985
Cz	0.97
Cl	0.95
Cm	0.95
Cn	1.05

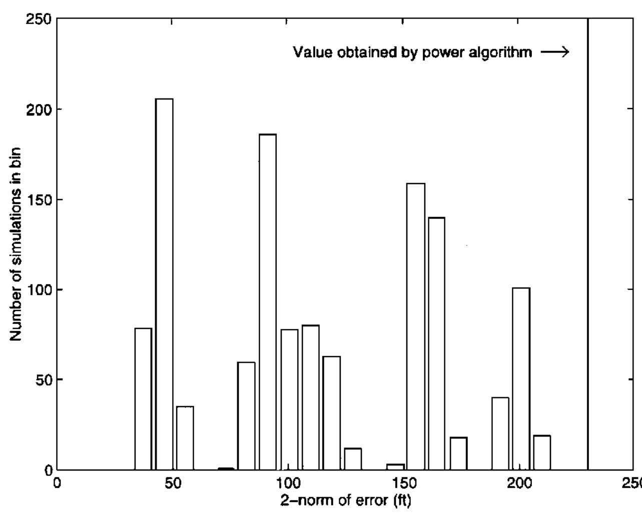

Fig. 5 Comparison of stochastic and worst-case analysis.

The resulting two norm of the performance variables is 230 ft. The behavior of the airplane under the worst-case parameter variation selected by the algorithm is illustrated in Fig. 4. The solid line represents nominal trajectory, and the dashed line represents the perturbed trajectory.

For comparison with more traditional ways of evaluating nonlinear system behavior, Monte Carlo simulations were run. For each parameter, the endpoints of the interval of variation were selected as allowable values. A system simulation with random turbulence subjected to the same restrictions is run for each possible combination of parameter values, 128 in this case. For each of these parameter combinations, 10 simulations are performed for a total of 1280 simulations. The power algorithm took $4\frac{1}{2}$ h for the signals to converge within 0.1% (51 iterations); however, a value within 0.1% of the final performance index had already been achieved after 13 iterations. The Monte Carlo simulations were computed in slightly more than 4 h on the same platform. Figure 5 presents a frequency distribution of the error two norms obtained from the Monte Carlo simulations. The vertical line indicates the worst-case value. The two methods have similar computational burden and provide information that is complementary. From the results of the Monte Carlo simulations, we can infer typical behavior of the system, as well as a probability distribution of the performance index. From the worst-case analysis, we obtain a better knowledge of the outer edge of this probability distribution. Obtaining this worst-case kind of information from random simulation would require a much larger number of experiments.

It is also important to develop computable upper bounds. For a restricted set of systems, an upper bound on the robust trajectory tracking problem performance index can also be computed,[23] in the form of a linear matrix inequality (LMI). However, better numerical tools are needed to solve the resulting LMIs given their size and structure.

VI. Conclusions

A numerically efficient algorithm for computing necessary conditions for performance specifications is shown to work on aerospace type problems and provide complementary information to that of Monte Carlo analysis methods. This algorithm addresses the problem of robust trajectory tracking, i.e., determining how far from the nominal the actual trajectory is, under the worst-case disturbances and uncertainty. To achieve the same tradeoff between generality, applicability, and computational efficiency that the structured singular value framework achieves for linear systems, we used the two norm as the measure for the noise signals, the undermodeled components gain, and the performance objectives.

We developed an algorithm to compute a lower bound on the maximum distance between nominal and actual trajectories. The algorithm can accommodate both external disturbances and unmodeled dynamical components in the system as sources of error. This algorithm is similar in nature to the power algorithm for μ and shares many of its numerical properties. Although the algorithm is not proven to converge in general, a numerical study of its behavior shows that it is well behaved when applied to some practical examples and that it outperforms alternative methods on those same problems. The computation load of the proposed method is similar to that of traditional Monte Carlo approaches. The information on the system provided by this method is complementary to the information provided by Monte Carlo analysis.

More experience is needed with the behavior of the lower bound power algorithm. It has to be tried in a larger variety of nonlinear systems and with a larger array of uncertainty descriptions. It is important to understand the bounds on its performance to develop improvements for it; this is the way the power algorithm for linear systems was perfected. It is also important to gain experience with the setup for the robust trajectory tracking problem: how to choose uncertainty descriptions, time- and frequency-domain weights, noise bounds, and performance criteria.

Acknowledgments

This work was partially supported by NASA and the Air Force Office of Scientific Research under Grant F49620-95-1-0419. The authors would like to thank Eugene A. Morelli for providing the F-16 simulation code.

References

[1]Packard, A., and Doyle, J. C., "The Complex Structured Singular Value," *Automatica*, Vol. 29, No. 1, 1993, pp. 71–109.

[2]Packard, A., Fan, M. K. H., and Doyle, J. C., "A Power Method for the Structured Singular Value," *Proceedings of the 27th Conference on Decision and Control* (Austin, TX), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1988, pp. 2132–2137.

[3]Young, P. M., and Doyle, J. C., "Computation of μ with Real and Complex Uncertainties," *Proceedings of the 29th Conference on Decision and Control* (Honolulu, HI), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1990, pp. 1230–1235.

[4]Jiang, Z. P., Teel, A. R., and Praly, L., "Small Gain Theorem for ISS Systems and Applications," *Mathematics of Control, Signals, and Systems*, Vol. 7, No. 2, 1994, pp. 95–120.

[5]Jiang, Z. P., and Praly, L. P., "Preliminary Results About Robust Lagrange Stability in Adaptive Nonlinear Regulation," *Systems and Control Letters*, Vol. 23, No. 1, 1994, pp. 67–78.

[6]Kristie, M., Kanellakopoulos, I., and Kokotovic, P. V., *Nonlinear and Adaptive Control Design*, Wiley, New York, 1995.

[7]Krstic, M., Sung, J., and Kokotovic, P. V., "Robust Control of Nonlinear Systems with Input Unmodelled Dynamics," *IEEE Transactions on Automatic Control*, Vol. 41, No. 6, 1996, pp. 913–920.

[8]Freeman, R. A., and Kokotovic, P. V., "Global Robustness of Nonlinear Systems to State Measurement Disturbances," *Proceedings of the 32nd IEEE Conference on Decision and Control* (San Antonio, TX), Inst. of Electrical and Electronics Engineers, New York, 1993, pp. 1507–1512.

[9]Teel, A., and Praly, L., "On Output-Feedback Stabilization for Systems with ISS Inverse Dynamics and Uncertainties," *Proceedings of the 32nd IEEE Conference on Decision and Control* (San Antonio, TX), Inst. of Electrical and Electronics Engineers, New York, 1993, pp. 1942–1947.

[10]van der Schaft, A. J., "L_2-gain Analysis of Nonlinear Systems and Nonlinear State Feedback H_∞ Control," *IEEE Transactions on Automatic Control*, Vol. 37, No. 6, 1992, pp. 770–784.

[11]Lu, W. M., and Doyle, J. C., "Robustness Analysis and Synthesis of Nonlinear Uncertain Systems," California Inst. of Technology, CIT-CDS-94-010, Pasadena, CA, 1994.

[12]"Application of Multivariable Control Design Guidelines to Aircraft Control Laws," Honeywell Technology Center and Lockheed Martin, Wright Lab. Rept. WL-TR-96-3099, Wright–Patterson AFB, OH, May 1996.

[13]"Advisory Circular: Automatic Landing Systems," Federal Aviation Administration, Rept. AC 20-57, Washington, DC, Jan. 1971.

[14]Hammersley, J. M., and Handscomb, D. C., *Monte Carlo Methods*, Methuen, London, 1975.

[15]Luenberger, D. G., *Optimization by Vector Space Methods*, Wiley, New York, 1968, Chap. 9.

[16]Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, and Control*, Halsted, New York, 1975, Chap. 2.

[17]Tierno, J. E., and Doyle, J. C., "Finite Time Horizon Robust Performance Analysis," *Proceedings of the 33rd Conference on Decision and Control* (Orlando, FL), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1994, pp. 3080–3085.

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

[19]Parlett, B. N., and Poole, W. G., "A Geometric Theory for the QR, LU, and Power Iterations," *SIAM Journal of Numerical Analysis*, Vol. 1, Feb. 1973, pp. 389–412.

[20]Newlin, M. P., and Glavaski, S., "Advances in the Computation of the μ Lower Bound," *Proceedings of the American Control Conference* (Seattle, WA), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1995, pp. 442–446.

[21]Tierno, J. E., "A Computational Approach to Nonlinear System Analysis," Ph.D. Thesis, Dept. of Electrical Engineering, California Inst. of Technology, Pasadena, CA, Nov. 1996.

[22]Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulation*, Wiley, New York, 1992.

[23]Tierno, J. E., and Murray, R. M., "Robust Performance Analysis for a Class of Uncertain Nonlinear Systems," *Proceedings of the 34th Conference on Decision and Control* (New Orleans, LA), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1995, pp. 1684–1689.