

An Efficient Algorithm for Performance Analysis of Nonlinear Control Systems

Jorge E. Tierno* Richard M. Murray John C. Doyle
Division of Engineering and Applied Science
California Institute of Technology,
Pasadena, CA 91125, U.S.A.

Abstract

A numerical algorithm for computing necessary conditions for performance specifications is developed for nonlinear uncertain systems. The algorithm is similar in nature and behavior to the power algorithm for the μ lower bound, and doesn't rely on a descent method. The algorithm is applied to a practical example.

1 Introduction

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. A recent description of this approach is given by Packard and Doyle [7].

For linear time invariant (LTI) systems with complex, structured uncertainty, analysis of robust performance can be reduced to searching for the solution of a set of algebraic equations which give bounds on the achievable performance. One is thus able to find computationally efficient solutions, such as the power algorithm for the μ lower bound, 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.

Current research in linear systems theory is devoted to extending the existing theory to incorporate more realistic uncertainty structures (such as real parameters) and to solving linear time varying and time invaring problems. In this paper we take a step in a separate direction: we present an algorithm for computing worst case performance for affine and nonlinear control systems.

Performance analysis for nonlinear systems is difficult due to the wide variety of behavior and structures which can occur. Most existing tools are at a theoretical level, and software for analyzing robust performance is not widely available.

In this paper we develop a numerical tool for the analysis of a fairly general nonlinear performance problem. Emphasis was placed on creating a computationally sound tool, requiring only information usually available on the process being analyzed. Our analysis tool only requires a simulation program for the plant and, as we will show, it greatly improves the current state of the art in nonlinear analysis.

We will consider the problem of robust trajectory tracking: given a nominal trajectory for an uncertain, noisy nonlinear system, a feedback controller which stabilizes the trajectory and a description of the desired performance, find a lower bound on the worst case performance. General purpose nonlinear programming algorithms can be used to solve this problem. However, our experience with performance analysis for

linear systems suggests that specific algorithms can be designed that significantly outperform the off-the-shelf ones in the sense that they give better answers with less computational effort [6]. Based on this, we develop in this paper a power algorithm to solve a specific two point boundary value problem that is similar in computational nature to the power algorithm for the lower bound of μ , and so does not rely on classical descent techniques.

We will also present numerical results obtained using the algorithm described to analyze step tracking performance of a ducted fan experimental setup at Caltech [3].

2 The Robust Trajectory Tracking Problem

Many nonlinear analysis problems of engineering interest can be reduced to a problem of tracking a nominal trajectory. Be it a car changing lanes on an automated highway, an airplane taking a turn, or an idling engine going through a sudden change in load, the designer has in mind an appropriate path, to be completed in a finite predetermined time, and builds his control system accordingly. Since the real system is not exactly the one used for the design, and since it is also subject to noise, the system will not follow the intended trajectory. However, the design can still be considered successful if it remains close enough to it in an appropriate norm.

In this paper we will consider a restricted version of this problem. Our performance measure will be the two norm of the error signal (i.e. the difference between the nominal and the actual trajectory). If needed, the error signal can be weighted by a multiplicative time function. Noise signals will be bounded in the 2-norm. Unmodeled dynamics will be norm bounded operators. The only information available on these operators is their 2-induced norm. We will not restrict this operators to be causal. The system equations will be allowed to depend on a set of real parameters varying in closed intervals. The initial conditions for some or all of the state variables will also be allowed to vary in given closed intervals.

To simplify the notation we will work in the following with a system with one uncertain parameter, one unmodeled norm bounded component and one noisy input. However all the results presented generalize naturally.

Let u be the noise signal perturbing the system, and let y be the error signal, that is the difference between the nominal and the actual trajectories. Denote by v the output of the uncertain dynamical block, and δ the real uncertain parameter. The equations describing the system will then be

$$\begin{aligned}\dot{x} &= f(x, u, v, \delta, t) \\ y &= g(x, u, v, \delta, t) \\ z &= g(x, u, v, \delta, t)\end{aligned}$$

with the following constraints

$$\begin{aligned}|\delta| &\leq 1 \\ \|u\|_2 &= 1 \\ \|v\|_2 &= \|z\|_2.\end{aligned}\tag{1}$$

E-Mail: jorge@hot.caltech.edu

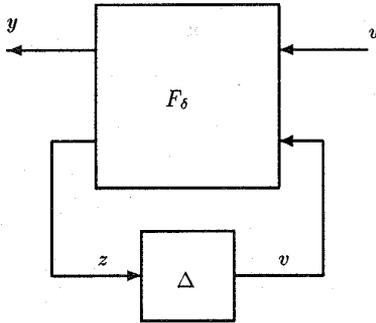


Figure 1: Schematic representation for the robust trajectory tracking problem.

Figure 1 gives a schematic representation of these equations. In order to determine how close the nominal trajectory is being tracked we need to compute

$$\max_{\|\delta\| \leq 1, \|u\|=1, \|v\|=\|z\|} \|y\|. \quad (2)$$

The preceding problem is a nonlinear constrained optimization problem. It is in general non-convex and some of the optimization variables live in an infinite dimensional space. An exact solution is thus out of the question: we have to settle for upper and lower bounds. Note that this is true even if the system is linear: the complexity of the problem does not come just from nonlinearities in the system but from the nonlinear nature of the optimization index and the constraints. In this paper we will develop an algorithm to compute a lower bound for equation (2), based on the search for locally worst case signals.

3 Necessary Conditions for Worst Case Signals

Any evaluation of the function $\|y\|_2$, for valid values of the parameters and signals is a lower bound on (2). So a simple way of getting lower bounds is through repeated simulation of the system for different values of the uncertain signals in the model. This is at present the state of the art of nonlinear analysis as applied in industry: good simulation models are developed and designs are tested through extensive simulation, usually selecting the uncertain signals at random. This approach is practical, since it requires information from the plant that is usually available, and often gives reasonable results. The algorithm we present on this paper will improve on this approach without sacrificing in simplicity or in the generality of the information required. Instead of simulating at random points, we would look for points that are good candidates for being local maximums. We will do this search through a "power-like" algorithm. In order to develop this algorithm we first have to establish the necessary conditions the signals and parameters have to meet in order to be worst case; we will do this by reducing the robust trajectory tracking problem to a well known dynamical systems optimization setup, that we present here for completeness.

Theorem 1 [2] For a dynamical system described by the equations:

$$\dot{x} = f(x, u, t) \quad x(0) \text{ given}, \quad t_0 \leq t \leq t_f \quad (3)$$

a performance index of the form

$$J = \int_{t_0}^{t_f} L(x, u, t) dt \quad (4)$$

and restrictions on the final state

$$G(x(t_f)) = c \quad (5)$$

if the signal u_0 achieves an extremum of J , then there exists a vector of constants μ and a solution to the two point boundary value problem:

$$\begin{aligned} \dot{x} &= f(x, u, t) \\ \dot{\lambda} &= - \left(\frac{\partial f}{\partial x} \right)^t \lambda - \left(\frac{\partial L}{\partial x} \right)^t \\ 0 &= \left(\frac{\partial L}{\partial u} \right) + \left(\frac{\partial f}{\partial u} \right)^t \lambda \end{aligned} \quad (6)$$

with boundary conditions:

$$\begin{aligned} x(0) &\text{ given} \\ \lambda(t_f) &= \left(\frac{\partial G}{\partial x(t_f)} \right)^t \mu \end{aligned} \quad (7)$$

Furthermore, if these conditions are met we will have

$$\lambda(t_i) = \frac{\partial J}{\partial x(t_i)} \quad (8)$$

We will show now how the robust trajectory tracking problem can be written in the form of Theorem 1. First we consider the performance condition. Define

$$L = \frac{1}{2} y^* y \quad (9)$$

Then optimizing $\|y\|$ is equivalent to optimizing

$$J = \int_{t_i}^{t_f} L dt =: \frac{1}{2} \|y\|^2 \quad (10)$$

The norm restrictions for the noise signals and the uncertain operator will be imposed through final conditions of additional states. Add to the system a new state named x_u and governed by the differential equation

$$\dot{x}_u = \frac{1}{2} u^* u \quad x_u(t_i) = 0 \quad (11)$$

Then $\|u\| = 1$ if and only if $x_u(t_f) = \frac{1}{2}$. Analogously for the uncertain block we add to the system a state x_Δ , governed by the differential equation

$$\dot{x}_\Delta = \frac{1}{2} (z^* z - v^* v) \quad x_\Delta(t_f) = 0 \quad (12)$$

Then $\|v\| = \|z\|$ if and only if $x_\Delta(t_f) = 0$. Finally, we will create a state that tracks the parameter δ . Let x_δ follow the equation

$$\dot{x}_\delta = 0 \quad x_\delta(t_i) = \delta \quad (13)$$

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

$$J = \int_{t_i}^{t_f} L dt = \frac{1}{2} \|y\|^2 \quad (14)$$

for the system verifying the differential equation

$$\begin{aligned}\dot{x} &= f(x, u, v, x_\delta) \\ \dot{x}_u &= \frac{1}{2}u^*u \\ \dot{x}_\Delta &= \frac{1}{2}(z^*z - v^*v) \\ \dot{x}_\delta &= 0,\end{aligned}\tag{15}$$

where

$$\begin{aligned}y &= g(x, u, v, x_\delta) \\ z &= h(x, u, v, x_\delta)\end{aligned}$$

with given initial conditions

$$x(t_o) = x_o, x_u(t_o) = 0, x_\Delta(t_o) = 0, x_\delta(t_o) = \delta\tag{16}$$

and final conditions

$$x_u(t_f) = \frac{1}{2}, x_\Delta(t_f) = 0.\tag{17}$$

This problem is in the form of equation (1). So a set of signals u , v , and a parameter δ achieve the worst case value of the performance index J only if there exists $\Lambda = (\lambda_x, \lambda_u, \lambda_\Delta, \lambda_\delta) \in \mathbb{R}^{n \times [t_o, t_f]}$, verifying

$$\dot{\lambda} = -\left(\frac{\partial f}{\partial x}\right)^t \lambda - \left(\frac{\partial h}{\partial x}\right)^t z \lambda_\Delta - \left(\frac{\partial g}{\partial x}\right)^t y\tag{18}$$

$$\dot{\lambda}_u = 0\tag{19}$$

$$\dot{\lambda}_\Delta = 0\tag{20}$$

$$\dot{\lambda}_\delta = -\left(\frac{\partial f}{\partial \delta}\right)^t \lambda - \left(\frac{\partial h}{\partial \delta}\right)^t z \lambda_\Delta - \left(\frac{\partial g}{\partial \delta}\right)^t y\tag{21}$$

with final state conditions

$$\begin{aligned}\lambda(t_f) &= 0 \\ \lambda_\delta(t_f) &= 0\end{aligned}\tag{22}$$

and verifying the following alignment conditions

$$\begin{aligned}\left(\frac{\partial f}{\partial u}\right)^t \lambda + u \lambda_u + \left(\frac{\partial h}{\partial u}\right)^t z \lambda_\Delta + \left(\frac{\partial g}{\partial u}\right)^t y &= 0 \\ \left(\frac{\partial f}{\partial v}\right)^t \lambda + \left(\left(\frac{\partial h}{\partial v}\right)^t z - v\right) \lambda_\Delta + \left(\frac{\partial g}{\partial v}\right)^t y &= 0\end{aligned}\tag{23}$$

and such that the initial state verifies:

$$\lambda_\delta(t_o) = 0, \text{ or } \begin{cases} \delta = -1 \\ \text{and} \\ \lambda_\delta(t_o) < 0 \end{cases} \text{ or } \begin{cases} \delta = 1 \\ \text{and} \\ \lambda_\delta(t_o) > 0. \end{cases}\tag{24}$$

Remarks: Equations (19) and (21) describe a linear time varying dynamical system whose inputs are the (scaled) outputs of the original system. We will refer to this system as the adjoint or co-system.

Equations (23) can be interpreted as an alignment condition between the outputs of the adjoint system and the inputs to the original dynamical system. Thus, these equations describe to dynamical systems interconnected in a feedback loop.

Equation (24) 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.

If we consider both the equations for the system, the co-system, and the alignment conditions together, we have a two point boundary value problem, i.e. a set of differential equations with boundary conditions at two distinct time instants.

Several methods for solving the general two point boundary value problem have been devised and thoroughly studied. (See for example [5], [1]). However, the standard methods are based on gradient descent. In what follows we present a method to solve this particular instance of the two point boundary value problem that avoids the problems of gradient descent methods. The algorithm is a generalization of the power algorithm for the lower bound of μ . In fact when applied to linear systems the proposed algorithm reduces to the standard power algorithm for μ as described by Young and Doyle [8].

4 A Power Algorithm

For a trajectory that meets the necessary conditions for a critical point, the Euler Lagrange conditions can be naturally separated into (i) a dynamical system with initial conditions only; (ii) a dynamical system with final conditions only; (iii) two sets of aligning conditions between the inputs and outputs of the two systems; and (iv) conditions relating the initial conditions of both systems. It is also important to note that the adjoint system depends on the trajectory of the original one.

So, if the perturbations signals achieve the necessary conditions, the following composition of mappings yields the identity map:

- Simulate the system along the current inputs.
- Compute the co-system along the current trajectory, and simulate it backwards in time.
- From the alignment conditions in (23) compute updated values for u , v , λ_u , and λ_Δ .
- Update the value of x_δ with the following rule:

$$\begin{aligned}\chi &= x_\delta + \lambda_\delta(t_o) \\ x_\delta &= \begin{cases} -1 & \chi < -1 \\ \chi & -1 \leq \chi \leq 1 \\ 1 & \chi > 1. \end{cases}\end{aligned}$$

Denote this composition by

$$(u^1, v^1, \delta^1, \lambda_u^1, \lambda_\Delta^1) = \Phi(u^o, v^o, \delta^o, \lambda_u^o, \lambda_\Delta^o).\tag{25}$$

Remarks: From the first equation in (23), using the old values of λ_Δ and of the state trajectory we can compute $u \lambda_u$. Since λ_u is a scalar, and we know the norm of u we can separate this product into its components.

From the second equation in (23) we can compute $\left(\left(\frac{\partial h}{\partial v}\right)^t z - v\right) \lambda_\Delta$. Since we now the norm of v , and the value of $\left(\frac{\partial h}{\partial v}\right)^t z$ we can determine v and λ_Δ by intersecting the line passing through the origin and with direction $\left(\left(\frac{\partial h}{\partial v}\right)^t z - v\right)$ with the circle centered at $\left(\frac{\partial h}{\partial v}\right)^t z$ and with radius $\|v\|$.

When solving the differential equations with a numerical integrator, we will obtain values for all the signals at a finite number of time instants. The number of operations necessary to the the signal updating described grows linearly with the number of time instants.

The following iterative algorithm searches for fixed points of Φ , by evaluating it repeatedly.

1. Simulate the system with $u = 0$, $v = 0$, $x_\delta = 0$. Use the time steps generated by the simulation routine as a time axis.

2. Select random values for u along the time axis. (the signal is to be interpolated in between time-steps). Normalize u to fit the norm requirement. Set $v^0 = 0$, and $x_\delta^0 = 0$. Set $\lambda_\Delta^0 = 1$.

3. Repeat

$$(u^{i+1}, v^{i+1}, x_\delta^{i+1}, \lambda_u^{i+1}, \lambda_\Delta^{i+1}) := \Phi(u^{i+1}, v^i, x_\delta^i, \lambda_u^i, \lambda_\Delta^i)$$

4. until

$$(u^{i+1}, v^{i+1}, x_\delta^{i+1}, \lambda_u^{i+1}, \lambda_\Delta^{i+1}) = (u^{i+1}, v^i, x_\delta^i, \lambda_u^i, \lambda_\Delta^i)$$

Remarks: If the algorithm converges, it converges to a fixed point of Φ and thus to a set of signals that meet the necessary conditions for a critical point.

In order to prove convergence we would have to prove that Φ is a contraction around fixed points. That has not been proved even for the simpler case when the system is linear.

When solving the differential equations with a numerical integrator, we will obtain values for all the signals at a finite number of time instants. The number of operations necessary to the the signal updating described grows linearly with the number of time instants.

5 Application to a practical example

We use the preceding algorithm to test the noise rejection capabilities of a full state feedback LQR design for an experimental ducted fan platform (Figure 2).

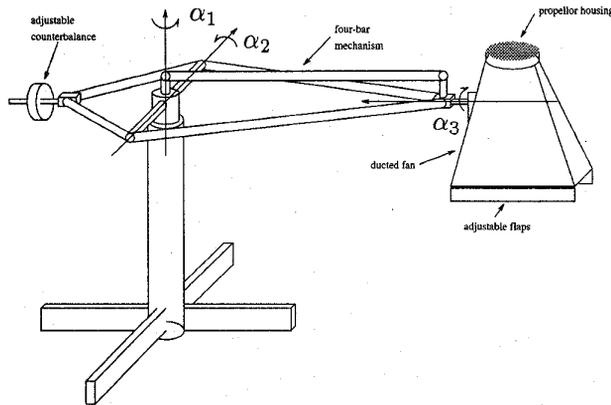


Figure 2: Overview of the experimental setup.

The experimental vehicle consists of a relatively simple ducted fan aircraft that can provide two dimensional vectored and reverse thrust. The aircraft is bolted to a rotating arm, which limits its motion to three degrees of freedom: one rotational and two translational, approximately on the surface of a sphere defined by the arm. With this geometry, the ducted fan is completely controllable with just the vectored thrust. The ducted fan itself is a wooden duct powered by a variable speed electric motor driving a propeller. A detachable flap assembly, mounted at the end of the duct gives the fan vectored and reverse thrust. The setup and its nonlinear model are described in more detail in [3].

We consider a controller whose objective is to make the vehicle track a step in vertical position, i.e. we want to move it from hover at zero altitude to hover at another prespecified altitude. The performance measure for the disturbance rejection problem will be the 2-norm of the distance from the nominal to the actual trajectory in the y position of the vehicle from beginning to end of the maneuver. A time domain

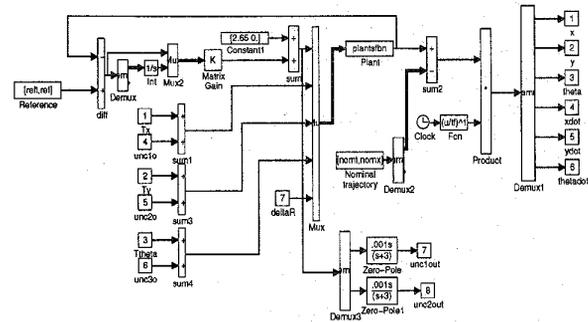


Figure 3: Simulink diagram for the ducted fan experiments.

weight multiplies the error signal to capture the fact that we consider the error at the final time to be more important than at the beginning. We will have three disturbances entering as torques along the three axes of rotation. The 2-norm of each of this noise signals is set to be .006 Nm over the 9 second time horizon (i.e., the rms value of the torque is .006 Nm). The uncertain block tries to capture the unmodeled dynamics of the electrical motor that is driving the fan. The input to the uncertain block is the command signal given to the motor filter through a high pass filter. The uncertain parameter represents the application point of the reaction force on the vehicle, since this point changes with, among other factors the position of the flaps. This distance is considered to be $.25m + \delta$ with $|\delta| < .1m$. A Simulink™ schematic diagram of the simulation model is given in Figure 3. More information on the experimental setup used, including copies of the simulation files can be obtained through the World Wide Web at <http://avalon.caltech.edu/~dfan>. A copy of the software used and its documentation is available from <ftp://avalon.caltech.edu/pub/cds>.

In Figure 4 we show the evolution of the error between the nominal and the actual trajectory starting from a random noise signal. Figure 5 shows the last two iterations of the same error signal.

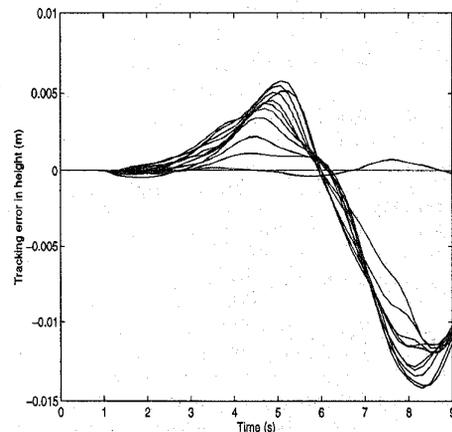


Figure 4: Evolution of the worst case trajectory.

In Figures 6 and 7 we show the disturbance signal and the input and output to the uncertain block respectively.

There are other more traditional approaches to solving this constrained optimization problem. We could for example use standard shooting methods to solve the two point boundary

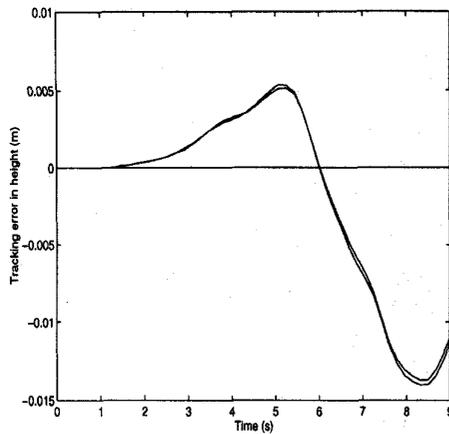


Figure 5: Trajectory for the last two iterations of the algorithm.

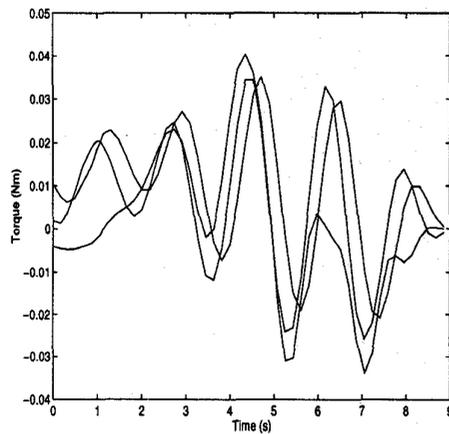


Figure 6: Final noise signals.

value problem. However since the adjoint system is usually unstable, integration of the combined differential equations forwards in time is numerically ill posed. Although the optimization problem is relatively low size, it is not suited for solution through numerical algorithms. Another way of setting the problem as a standard optimization problem, is to discretize in time the input signals and optimize the performance index subject to the norm constraints in the input. This gives rise however to a very large optimization problem, that push the performance boundaries of the shelf optimization packages. For the example set in this paper NPSOL [4] took three times as long to find a worse answer to a simplified version of the problem.

6 Conclusions

In this paper we showed how a power algorithm can be used to compute a necessary condition for robust performance of nonlinear systems by solving a particular instance of the two point boundary value problem. We also presented an example illustrative of the behavior of that algorithm.

Due to the nature of the results presented here, the success of the algorithm can only be determined by gathering experience from many different problems and making computational comparisons with other approaches. It is unlikely that we will be able to derive guarantees stronger than local

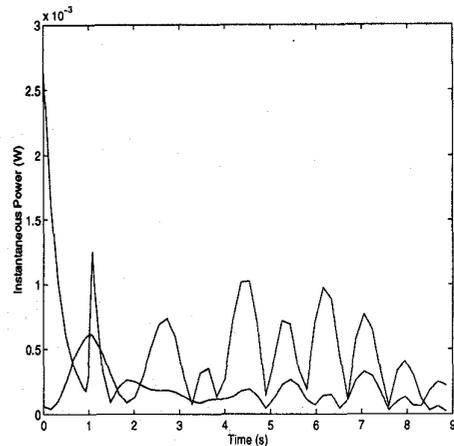


Figure 7: Power of final inputs and outputs to the uncertain block.

convergence. However, we see a great need for computational tools that behave reasonably in examples derived from applications. The algorithm presented here shows that such tools are possible, and that they can provide useful answers in reasonable periods of time.

A deeper investigation of the numerical properties of our algorithm is needed and future research will concentrate on this. A more thorough comparison with other optimization methods will be carried out. It is important to identify in what problems the algorithm performs well and in what problems it doesn't. As it was done with the power algorithm for the lower bound of μ this may allow us to perfect the algorithm to widen the set of problems it can solve without compromising on its average behavior.

Acknowledgements

This work was partially supported by NASA, NSF and AFOSR.

References

- [1] U.M Ascher, R.M. Mattheij, and R.D. Russel. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice Hall, 1988.
- [2] A. E. Bryson and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Halsted Press, 1975.
- [3] H. Choi, P. Sturdza, and R. M. Murray. Design and construction of a small ducted fan engine for nonlinear control experiments. In *Proceedings of the American Control Conference*, pages 2618-2622, 1994.
- [4] P. E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. *User's Guide for NPSOL*. Dept. of Operations Research, Stanford University, January 1986.
- [5] H.B. Keller. *Numerical methods for two point boundary-value problems*. Blaisdell, 1968.
- [6] M. Newlin and S. Glavaski. Advances in the computation of the μ lower bound. In *Proceedings of the American Control Conference*, 1995. to appear.
- [7] A. Packard and J. C. Doyle. The complex structured singular value. *Automatica*, 29(1):71-109, 1993.
- [8] P. M. Young and J. C. Doyle. Computation of μ with real and complex uncertainties. In *Proceedings of the 29th Conference on Decision and Control*, pages 1230-1235. IEEE, 1990.