Nonlinear Receding Horizon Control of F-16 Aircraft

Raktim Bhattacharya, Gary J. Balas * Alpay Kaya, Andy Packard †

Abstract

In this paper the application of receding horizon control (RHC) with the linear parameter varying (LPV) design methodology to a high fidelity, nonlinear F-16 aircraft model is demonstrated. The highlights of the paper are i)Use of RHC to improve upon the performance of a LPV regulator. ii)Discussion on details of implementation such as control space formulation, tuning of RHC parameters, computation time and numerical properties of the algorithms. iii)Simulated response of nonlinear RHC and LPV regulator.

1 Introduction

Receding horizon control, also known as model predictive control, has been popular in the process control industry for several years [1, 2]. It is based on the simple idea of repetitive solution of an optimal control problem and state updating after the first input of the optimal command sequence. The repetitive nature of the algorithm results in a state dependent feedback control law.

The focus of this paper is on the application of RHC techniques to nonlinear systems. Specifically, the use of affine, linear, parameter varying control [3] to derive a control Lyapunov function (CLF) which is used as a terminal penalty function in the RHC framework. Similar work has been done by Primbs *et al.*[4] and Ali *et al.*[5] on the Caltech Ducted Fan [6].In this work stability is guaranteed by choosing a suitable terminal cost. No constraints are necessary to guarantee stability. This approach is used to formulate our RHC problem for the F-16 model.

The receding horizon control problem investigated in this paper is a regulation based problem. We are interested in regulating state perturbations of F-16 to a target set, namely the origin. The performance of the controller is measured in terms of the value of a objective function, which depends on the state and control trajectories. A LPV controller is designed for the F-16 aircraft and a CLF is obtained from it. The CLF is used as a terminal cost in the formulation of the optimization problem for receding horizon control.

The purpose of this paper is to demonstrate an aerospace application of nonlinear RHC algorithm and also discuss the implementation issues in detail. The latter is missing in most publications in this field. This paper presents an overview of the nonlinear RHC theory, F-16 model details, LPV regulator design, optimization problem setup for RHC, formulation of control space, numerical issues in nonlinear simulation, hardware and software details.

*Department of Aerospace Engineering & Mechanics, Univ. of Minnesota, Minneapolis, MN-55414, {raktim,balas}@aem.umn.edu

†Department of Mechanical Engineering, UC Berkeley, CA-94720,{alpay,pack}@jagger.me.berkeley.edu

2 Stability and Performance of Nonlinear RHC

Nonlinear trajectory optimization problem can be posed as (pg 131 in [7])

$$J = \min_{u(.)} \Phi(x,t)\Big|_{t=t_0+T} + \int_{t_0}^{t_0+T} L(x,u)dt \qquad (1)$$

With system dynamics, initial condition and terminal boundary condition defined as $\dot{x} = f(x,u)$, $x(t_0) = x_0$ and $\Psi(x,t)\Big|_{t=t_0+T} = 0$. The optimal control trajectories for RHC are usually obtained by solving an Euler-Lagrange type of optimization. Its solution is locally optimal and is valid only for a single initial condition. The desired feedback nature of the receding horizon control law is achieved by resolving the optimization at every encountered state.

The unconstrained finite horizon optimization doesn't guarantee stability. The stability of finite horizon RHC algorithm is guaranteed by imposing constraints on the optimization.

2.1 Development of Theories for Stability

The first method to guarantee stability [8, 9] imposes a terminal boundary condition on state as $x(t_0 + T) = 0$ Since a nonlinear optimization problem with equality terminal constraint is computationally demanding, in [10] closed loop stability is ensured by imposing a terminal *inequality* constraint. The constraint requires $x(t_0 + T)$ to enter a suitable neighborhood of the origin. Once such a neighborhood is reached, the receding horizon controller is switched to a locally stabilizing linear controller. A different approach has been proposed in [11] where the receding horizon controller is obtained by solving a finite horizon problem with quadratic terminal state penalty $\Phi(x,t)\Big|_{t_0+T} = ax^TPx\Big|_{t_0+T}$ for some $a, P > 0$. In another method proposed by Primbs *et al.*[4], first a globally stabilizing control law is achieved by finding a global control Lyapunov function (CLF).

Our implementation of RHC for the F-16 model is based on the work by Jadbabaie, Yu and Hauser in [12]. This method is a hybrid of the methods proposed by Primbs *et al.*and De Nicolao *et al.*[13]. As in [4] it is assumed that a suitable CLF already exists. The CLF is used to replace the state inequality constraint with a terminal cost. The terminal cost is the cost incurred if the controller from the CLF is applied at the end of the horizon. We derived the CLF from a LPV regulator. In theory, as in [13], the CLF based controller is never implemented, it is used only to compute the terminal cost. However, due to the limitations in the optimization software we used, we apply the LPV regulator to pre-stabilize the system. The optimization software, NPSOL [14], has numerical problems with unstable systems. It uses shooting method to search for an optimal direction and a candidate solution can potentially destabilize an unstable system.

3 F-16 aircraft model

The nonlinear F-16 model used in receding horizon control is available in [15] which we shall represent as

$$\dot{x}_{NL} = f(x_{NL}, \delta) \qquad (2)$$

The mathematical model uses the wind-tunnel data from the NASA-Langley wind-tunnel tests on a scale model of an F-16 aircraft [16]. The aerodynamic data is valid up to Mach 0.99, angle of attack range of $-10^\circ \leq \alpha \leq 45^\circ$ and side slip angle range $-30^\circ \leq \beta \leq 30^\circ$. The wind-tunnel tests were conducted on sufficiently close points to capture the nonlinear behavior of the aerodynamic force and moment co-efficient. The states $x_{NL} \in \mathcal{R}^{12}$ and controls $\delta \in \mathcal{R}^4$ in the model are defined as $x_{NL} = [npos, epos, h, \phi, \theta, \psi, V_t, \alpha, \beta, p, q, r]$ and $\delta = [T, \delta_e, \delta_a, \delta_r]$. The notations have their usual meaning. Angular deflections in δ are in degrees and in x_{NL} they are in radians. Distances are in ft, velocity in ft/s and thrust in lb. Actuators for the control surfaces and engine are modelled as first order systems, details of which are discussed later in the paper.

4 LPV Modelling of F-16

Based on aerodynamic data in [16], the control variables in equation(2) enter affinely in δ_a, δ_r, and T, though not in δ_e. To derive an LPV model of the aircraft's nonlinear equation of motion, it is necessary that all the controls be in affine form. This is achieved by transforming (δ_e, T) into synthetic inputs (u_1^S, u_2^S) with $\xi : \mathcal{R}^2 \to \mathcal{R}^2$, so that the equation of motion, after defining $\delta_{a,r} := [\delta_a \;\; \delta_r]^T$ and $u^S := [u_1^S \;\; u_2^S]^T$, can be written as $\dot{x}_{NL} = h(x_{NL}) + g(x_{NL}) \begin{bmatrix} \delta_{a,r} \\ u^S \end{bmatrix}$.

4.1 Inverse Table

Using the synthetic input formulation means that the LPV regulator is designed using the (V_t,α) dependent matrix gain. Thus, the control commands are not in terms of elevator and thrust, but in terms of synthetic inputs. This necessitates the ability to calculate (δ_e, T) from a given (V_t, α, u^S). The synthetic input inverse table, denoted by ξ^{-1}, is used to recover the elevator and thrust values from the synthetic input values. A detail treatment of this is available in [17].

4.2 LPV Model of F-16

Assuming first-order dynamics for each actuator, the LPV model derived for the F-16 has 12 states (eight flight dynamics and four actuator) and four inputs, $x = \left[V_t \; \alpha \; \beta \; p \; q \; r \; \phi \; \theta \; \delta_a \; \delta_r \; \delta_1^S \; \delta_2^S\right]^T, u_{LPV} = \left[u_a \; u_r \; u_1^S \; u_2^S\right]^T$, both of which are centered about their trim values ($\overline{x}$,$\overline{u}_{LPV}$) and the LPV model is of the form

$$\dot{x} = A_{LPV}(\rho)x + B_{LPV}(\rho)u_{LPV}. \qquad (3)$$

Gain scheduling is done on parameter $\rho \in \mathcal{P} \subset \mathcal{R}^3$, defined as $\rho = \{V_t, \alpha, \beta\}$ It is assumed that the set $\mathcal{P}$ is compact and $\dot{\rho}$ is bounded (i.e., $\dot{\rho}_{min} \leq \dot{\rho} \leq \dot{\rho}_{max}$). The matrix functions $A_{LPV}(\rho), B_{LPV}(\rho)$ are appropriately defined. Since the scheduling parameters are also states of the system, the LPV model is actually a quasi-LPV model. A quasi-LPV system is a special class of LPV system where the system varies as functions of state variables and exogenous variables. The state variables and the exogenous variables are required to be continuous functions of time and measurable in real time. Note that the LPV full state control design algorithm cannot take advantage of the fact that the scheduled variables are also the system states. Therefore, the full state LPV regulator may be conservative.

A cost function for (3) penalizing not only the states and inputs, but the actuator rates, is considered here. Clearly, this will result in the state cost being coupled to the input cost.

5 RHC formulation

The optimization problem is for RHC is defined as $J = \min_{u(.)} \int_{t_0}^{t_0+T} L(x,u)dt + V[x(t_0+T)]$, where $L(x,u)$ is a positive definite function of $x(.)$, $u(.)$ and $V[x(t_0+T)]$ is the terminal penalty. The value function $V(x)$ is defined as $V(x) = x^T X^{-1}(\rho)x$, as shown in pg. 80 of [4] and $X^{-1}(\rho)$ is obtained from the LPV regulator synthesis.

6 Formulation of Control Space for Optimization

A common practice for solving optimal control problems is to convert them into parameter optimization problems. There are several methods available for conversion [18]. Numerical integration, collocation, direct transcription and differential inclusion are examples of these conversion methods. Most of these techniques are similar in nature and differ by what is guessed (state or control), how the integration is carried out (implicit or explicit), and the order of integration. This process of conversion is called "suboptimal" control because search is restricted to a particular subspace of the finite dimensional control space. For our RHC simulation, we have used B-splines [19] to formulate our control space [20]. For a r^{th} order B-spline, the control vector is defined as $u(t) = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t)$, where $\phi_k(t)$ are normalized scalar B-spline basis functions defined on a uniform sequence of break points or knot sequence $t_N = \{k\triangle t\}_{k=-r+1}^{N+r-1}$. N is thenumber of time intervals, $\alpha_k \in \mathcal{R}^4$ are the parameters of optimization and $\triangle t$ is the fixed time step in the discretised horizon.

As an example, for a horizon length of 0.3 s and discrete time step of 0.05 s, we have $N = 6$ intervals. If we use 3^{rd} order B-splines, i.e. piecewise quadratic polynomials, then the number of parameters that determine the control trajectory would be $N+r-1 = 6+3-1 = 8$ for each control, making a total of $4*8 = 32$ parameters to optimize. The knot sequence for this case would be $t_N = \{0.05*k\}_{k=-2}^{8}$ with $N+2r-1 = 11$ break points. To visually understand how B-splines are constructed, consider a knot sequence $k = [0, 0.05, 0.1, 0.15, 0.2, 0.25]$ and $\alpha = [0.4001, 0.1988, 0.6252]$. The coefficient vector for this example is chosen at random. The resulting graph and basis functions are shown in Fig.1. The resulting curve, shown in solid line, is the summation of the three non-solid curves. The shape of the curve is determined by the knot sequence, the order of B-spline and the coefficient multiplying each basis function. A 3^{rd} order B-spline exhibits positional, first derivative and second derivative continuity.

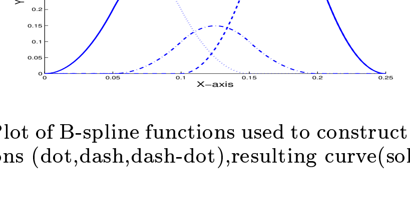

Figure 1: Plot of B-spline functions used to construct a trajectory. Basis functions (dot,dash,dash-dot),resulting curve(solid)

7 Simulation Setup

7.1 F-16 Model

The system interconnection for the F-16 aircraft is shown in Fig.2. The control objective is to regulate the system from a perturbed state back to a trim condition using receding horizon control. The optimization to compute optimal open-loop

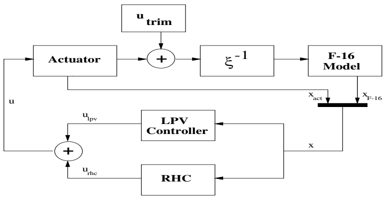

Figure 2: System Interconnection

control trajectory is carried out every 0.05 sec, i.e the receding horizon controller runs at 20Hz. The pre-break points LPV controller runs every 0.01 sec (100 Hz). A fourth order Runge-Kutta integrator is used to update the system states. The step size for the integrator is 0.01s, same as the LPV controller sampling rate. The F-16 model used for state updates and control trajectory synthesis is a full nonlinear model given in (2). The aircraft model is written in `ANSI C` which includes routines needed to interpolate aerodynamic data from lookup tables. The airplane is trimmed at an altitude of 10,000 ft and angle of attack $\alpha = 7°$. Trimming is done by minimizing a quadratic cost function of the moments acting on the aircraft. Although the full nonlinear model has 12 aircraft states, we have only considered the state trajectories that are common with the LPV plant, i.e. $(V_t, \alpha, \beta, p, q, r, \phi, \theta)$, the controls are the same as original, i.e $(T, \delta_e, \delta_a, \delta_r)$. Note that the control signals generated by the LPV and RHC controllers are the synthetic inputs. The actuator and the trim values are for control vector $(\delta_a, \delta_r, u_1, u_2)$. They are converted to $(T, \delta_e, \delta_a, \delta_r)$ by the block labelled as ξ^{-1}. The actuators are modelled as first order linear systems with $|\delta_a| \leq 20°, |\dot{\delta_a}| \leq 80°/s, \tau_s = 0.0495s$ lag; $|\delta_r| \leq 30°, |\dot{\delta_r}| \leq 120°/s, \tau_s = 0.0495s$ lag; $|u_1| \leq 1.1$ rad, $|\dot{u_1}| \leq 2.4$ rad/s, $\tau_s = 0.0495s$ lag; $|u_2| \leq 0.8$ rad, $|\dot{u_2}| \leq 1.0$ rad/s, $\tau_s = 0.5s$ lag).

The magnitude and rate bounds of the actuators are implemented in the nonlinear simulation.

7.2 Numerical Details

Numerical analysis is a key part in receding horizon control simulations. Formulation of control space, integration algorithm used to update states and most importantly the optimization software, all require numerical computation. Using higher order algorithms provide better results but it is also computationally expensive. Therefore one has to compromise between accuracy of solution and computational time. The online nature of the receding horizon control law has stringent real-time demands on the software used. We are still quite far from applying such control laws to aerospace problems in real life, based on the formulation in this paper. Nevertheless, an effort has been made to reduce the execution time of the nonlinear simulation software as much as possible. The following subsections briefly describe the tuning of the NPSOL software and finer numerical details that helped in reducing the computational time.

7.2.1 NPSOL

NPSOL[14] is the heart of our simulation software. NPSOL is an optimization package written in `FORTRAN`. This was converted to `ANSI C` using `f2c`. The `ANSI C` version of the library is 30-50% slower than the `FORTRAN` code. We used the `ANSI C` version because of the convenience in linking and compiling with other `ANSI C` modules of our simulation package.

The algorithm and software parameter selection for NPSOL is briefly described here.

- NPSOL uses sequential quadratic programming (SQP) for nonlinear optimization. The SQP algorithm requires the gradient of the cost function with respect to the optimization parameters. The user has the option to supply these gradient as user defined functions, or let NPSOL determine them numerically. We chose the latter for convenience at the expense of computational time.

- The basic structure of NPSOL involves major and minor iterations. The minor iterations solves a quadratic subproblem that determines the search direction. Once the search direction has been computed, the major iteration proceeds to the next iterate which produces a sufficient decrease in the augmented Lagrangian merit function. The sequence of iterates from the major iterations converge to a first-order Kuhn-Tucker point of the nonlinear optimization problem. Naturally, the limits on the maximum iteration number for both major and minor iterations affect the solution of the problem. Major iteration limit was set to 10 and 5 for the minor iterations. These limits may not be suitable for all RHC simulations with various horizon lengths.

- Optimization on bounded parameters are usually faster than for unbounded ones. The actuator model imposes rate and magnitude limits on the control action. Incorporating these constraints into the parameter optimization formulation improve the numerical conditioning of the problem. The bounds on the parameters of optimization, $\alpha_k \in \mathcal{R}^4$, control the magnitude limits of the B-spline curves generated. We defined the bounds of the control rates to be the bounds of these parameters. B-spline curves generated with α_k forms the time trajectory of u_{rhc}, which influences the trajectory of $\dot{x}_{act}$. Since we are indirectly shaping the control rate trajectory with our manipulated variables, $u_{rhc}(\alpha_k)$, bounds on the optimization parameters, α_k, are selected based on the limits associated with control rates $\dot{x}_{act}$. Note that the stabilizing controller u_{lpv} acts in addition to u_{rhc}. Bounding u_{rhc} with limits of $\dot{x}_{act}$ doesn't guarantee boundedness of $\dot{x}_{act}$.

- NPSOL needs an interval to be defined over which gradients can be estimated via finite difference. If a difference interval is not specified by the user, NPSOL computes it automatically with a procedure that requires six calls to the cost function. Since this is computationally expensive, we defined this function discretisation interval as 0.001 for our simulations. The solution time with interval specified is 185.06 sec and without it is 334.9 sec.

- Warm start was implement from the second solution of optimal control. The time with warm start is 185.06 sec and time without is 175.41 sec.

- Optimality tolerance was set to 0.01. The optimality tolerance determines the desired accuracy of the optimal solution. For example, with a tolerance of 10^{-6} the solution will have six correct figures.

7.2.2 Simulation Details

This section presents the numerical information required for implementation of receding horizon control law.

- The optimization to generate the control trajectories was performed every 0.05 sec. Since this optimization determines the optimal command from the current state, this is equivalent to saying that the feedback information was provided every 0.05 sec.

- Third order B-splines were used in the control space formulation. This assumes the control trajectory behaves as quadratic polynomials in the time interval used to discretise the horizon length. A second order B-spline would yield a piecewise linear control trajectory and a first order B-spline would form a piecewise constant control trajectory. Since we have four control variables to manipulate, reducing the order of B-spline r by one reduces the total number of parameters $4(N+r-1)$ by 4, for a given number of intervals N in the horizon length. Reducing the number of parameters will reduce the computational time but the set of allowable control trajectories is also reduced. The effect of B-spline order on cost-to-go is investigated in this paper.

- A fourth order Runge Kutta algorithm is used to integrate the system with fixed step size of 0.01s.

- The nonlinear F-16 model, aerodynamic data interpolation algorithms, basic linear algebra computation, is implemented in `ANSI C`.

8 Results

This section presents the results of the receding horizon control optimization simulation, with the state-feedback LPV controller used both for pre-stabilizing and as the CLF in the RHC formulation. The performance of RHC and LPV controllers are compared in terms of the cost incurred to reach the origin from a given initial condition. A 5.0 sec RHC simulation took 149s and 9348s cputime for horizon lengths of 0.1s and 1.0s, with third order B-splines as the basis for control space formulation and 0.05s horizon length discretisation. The F-16 model was trimmed at an altitude of 10K ft and 7° α. The trim states and controls at this flight condition are $\left[\bar{V_t}, \bar{\theta}, \bar{\alpha}, \bar{u_1}, \bar{u_2}\right] = [387.58, 0.12, 0.12, -0.13, 0.13]$, with velocity in ft/s and rest in radians. The remaining states and controls are zero at trim. The initial condition of the system was set by adding a 15° and 10° perturbation in angle of attack α and side slip angle β respectively. The control objective is to return the aircraft to its trim condition while respecting the limits on actuator magnitude and rates.

Results for RHC simulation with horizon lengths 0.1s and 1.0s are shown in figures 4 and 5. The control trajectories shown are the actuator outputs and not the controller outputs, and hence they satisfy magnitude and rate bounds. The cost for LPV control is 108.24, with cost computed as

$$J = \int_0^{5.0} [x \; u_{LPV}] \, P \begin{bmatrix} x \\ u_{LPV} \end{bmatrix} dt$$

where

$$\begin{aligned}
P &= \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}_{16\times 16} \\
P_{11} &= diag\{6.25 * 10^{-4}, 205, 3282, 156, 277, 100, \\
&\quad 3282, 820, 217, 101, 72, 3\}_{12\times 12} \\
P_{12} &= \begin{bmatrix} 0_{8\times 4} \\ diag\{-14.5, -9.6, -8.42, -0.5\} \end{bmatrix}_{12\times 4} \\
P_{21} &= [0_{4\times 8} \;\; diag\{-14.5, -9.6, -8.42, -0.5\}]_{4\times 12} \\
P_{22} &= diag\{217, 101, 72, 3\}_{4\times 4}
\end{aligned}$$

Nonlinear simulations reveal that the RHC cost, is significantly smaller than the LPV cost. This is reflected in the fast convergence of the perturbations to origin. The variation of cost with horizon length is as follows

Horizon length(s)	0.1	0.2	0.4	1.0
Cost	89.92	82.54	77.98	77.23
CPU Time(s)	149	412	1989	9348

The horizon length plays an important role in the performance of the receding horizon controller. It can be inferred from figures 4 and 5 that the state and control trajectories become less oscillatory as horizon length is increased. The effect of horizon length on the trajectories can be explained by drawing analogies between PID controllers and receding horizon controllers.

The RHC optimization with horizon length of 0.05s led to control trajectories that caused limit cycle like behaviour in the system. This phenomenon can be explained based on the speed of response of the system. The first order actuators in the F-16 aircraft model have time constant of 0.049s. Therefore a horizon length of 0.05 was too short for the RHC algorithm to observe the effects of a given control input. The sequence of consistent large control inputs led the system into instability.

For longer horizon lengths, the system is able to fully respond to a given control input. The RHC algorithm is able to observe both transient and post-transient effects of the system. This rich information about the system's response, yields control trajectories that are less aggressive. The resulting system trajectories are smoother and more optimal. The variation of the cost-to-go with horizon length is shown in Fig.3.

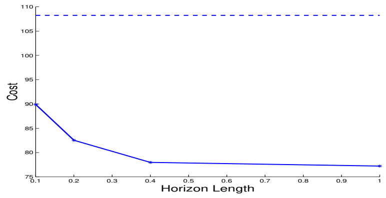

Figure 3: Cost vs horizon length (B-spline order = 3) dashed=LPV cost, solid=RHC cost

The over all cost-to-go is lower for longer horizon lengths because the RHC algorithm has richer knowledge of the system behavior to a given input, so better is the decision on optimality for every control trajectory generated. This also translates to the fact that optimization over longer time horizon reduces the contribution of the terminal cost on the total cost. Since the terminal cost serves as an approximation which is greater than the value of the truncated integral, reducing its contribution yields a total cost that is closer to the optimal.

It is also observed that increasing the horizon length beyond a certain time doesn't improve the performance of the controller in terms of the cost-to-go. This is because for a long enough time horizon the system is steered close to the origin within that time and the value of the truncated integral is almost zero. The horizon length has reached ∞ relative to the system response time.

From the simulation, we conclude that combination of LPV and receding horizon controller provide a better performance than the LPV controller alone. The difference is significant even for small horizon lengths and low B-spline orders. The combination of the two controllers yields control trajectories that is able to aggressively maneuver the state perturbations to the origin and at the same time satisfies limitations on the control rate and magnitude. RHC simulations were done on a 400 Mhz Pentium II machine running Debian Linux.

9 Summary and Conclusions

The combination of on-line receding horizon control and off-line linear, parameter-varying regulator design, led to improved performance on a F-16 flight control example. The LPV regulator is used to pre-stabilize the unstable, F-16 aircraft and as a control Lyapunov function end point penalty in the RHC optimization. The numerics of the nonlinear RHC optimization is a significant issue. The convergence of the RHC optimizations and the overall computational time are some issues that need to be examined in order to achieve a reliable on-line control algorithm.

References

[1] S.J. Qin and T. Badgwell. An overview of industrial model predictive control technology. *AIChE Symposium Series*, 93:232–56, 1996.

[2] A. Bemporad and M. Morari. Robust Model Predictive Control: A Survey. Technical report, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Physikstrasse 3, CH-8092 Zürich, Switzerland, www.control.ethz.ch, 1999.

[3] J. Shamma and J. Cloutier. Gain-scheduled missile autopilot design using linear parameter varying transformation. *Journal of Guidance, Control and Dynamics*, 16:256–263, 1993.

[4] J.A. Primbs. *Nonlinear Optimal Control: A Receding Horizon Approach*. PhD thesis, California Institute of Technology, 1999.

[5] A. Jadbabaie, J. Yu, J. Primbs, and Y. Huang. Comparison of nonlinear control designs for a ducted fan model. *IFAC World Congress, IFAC-2c-112*, Beijing, China, July 1999.

[6] M. Milam and R.M. Murray. A testbed for nonlinear systems: The caltech ducted fan. *1999 IEEE Conference on Control Applications*, 1999.

[7] Arthur E. Bryson, Jr. and Yu-Chi Ho. *Applied Optimal Control*. Hemisphere Publication Corporation, Washington D.C., 1975.

[8] S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems. *J. Optim. Theory Appl.*, 57:265–93, 1988.

[9] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Trans. Auto. Cont.*, 35:814–24, 1990.

[10] D.Q. Mayne and H.Michalska. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Auto. Cont.*, 38:1623–33, 1990.

[11] T. Parisini and R. Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31:1443–51, 1995.

[12] A. Jadbabaie, J. Yu, and J. Hauser. Stabilizing receding horizon control of nonlinear systems: A control lyapunov function approach. *American Control Conference*, 3:1535–9, 1999.

[13] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Trans. Auto. Cont.*, 43(7):1030–6, 1998.

[14] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. NPSOL - nonlinear programming software. Stanford Business Software, Inc., 2680 Bayshore Parkway, Suite 304, Mountain View, CA 94043.

[15] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. Wiley-Interscience Publication, John Wiley & Sons, Inc., 1992.

[16] L.T. Nguyen, M.E Ogburn, W.P. Gilbert, K.S. Kibler, P.W. Brown, and P.L. Deal. Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability. Technical Report 1538, NASA Langley Research Center, Hampton, Virginia, 1979.

[17] Jong Yeob Shin. *Worst-Case Analysis and Linear Parameter-Varying Gain-Scheduled Control of Aerospace Systems*. PhD thesis, Department of Aerospace Engg & Mechanics, University of Minnesota, 2000.

[18] D. Hull. Conversion of optimal control problems into parameter optimization problems. *AIAA Guidance Navigation and Control Conference*, AIAA-96-3812, 1996.

[19] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[20] A.L. Schwartz. *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*. PhD thesis, Electronics Research Laboratory, UC Berkeley, 1996.

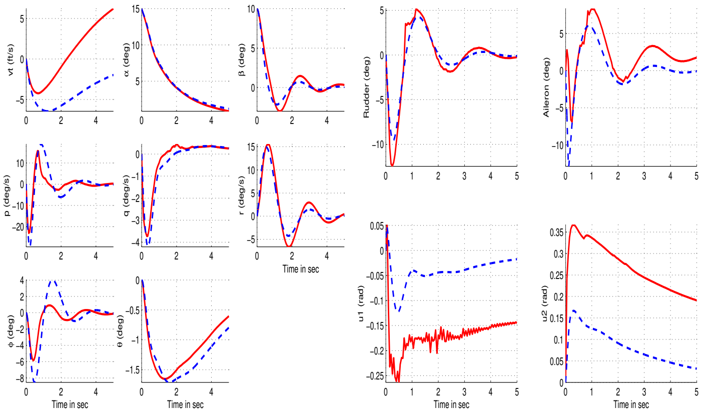

Figure 4: Perturbation trajectories horizon length=0.1s; RHC(solid), LPV(dashed).

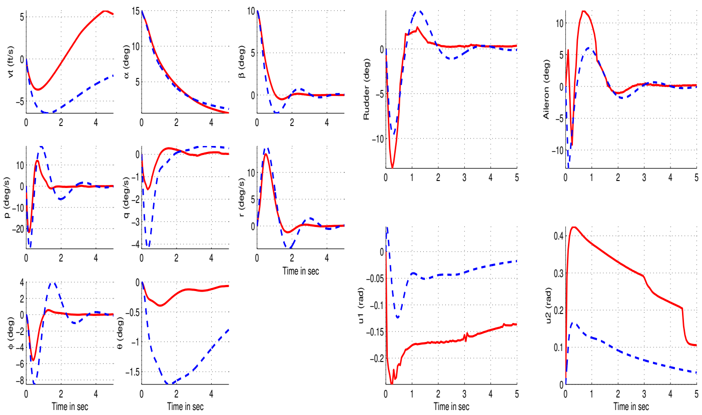

Figure 5: Perturbation trajectories, horizon length=1.0s; RHC(solid), LPV(dashed).