RECEDING HORIZON CONTROL OF AN F-16 AIRCRAFT: A COMPARATIVE STUDY

T. Keviczky and G.J. Balas

Department of Aerospace Engineering and Mechanics, University of Minnesota,
107 Akerman Hall, 110 Union Street S.E., Minneapolis, Minnesota 55455
Phone: +(612) 625-8000 Fax: +(612) 626-1558
E-mail: keviczky@aem.umn.edu / balas@aem.umn.edu

Keywords: Receding Horizon Control, Flight Control, Optimization

Abstract

A comparison between Receding Horizon Control (RHC) approaches is presented for the longitudinal axis control of an F-16 aircraft. The results suggest that the flexibility provided by an adaptive RHC scheme based on flight condition dependent linear prediction models is a necessary requirement for achieving good performance as opposed to a single LTI model based method. The adaptive scheme offers an attractive alternative to a full nonlinear model based RHC approach by trading off an acceptable degradation in performance to modest computational complexity and real-time implementability.

1 Introduction

Receding horizon control (RHC) methodologies, also known as model based predictive control methods, have been in the limelight of significant research efforts, motivated by several successful industrial applications [6, 1, 12]. The process industry provided a perfect fit for these algorithms that respected critical process-constraints to achieve safer and more efficient operation of industrial plants. These applications were not only "well-suited" for RHC methods but due to their relatively slow dynamics (large time constants), the significant computational effort of repetitive optimization, which is inherently involved in receding horizon approaches, could be accommodated by the relatively infrequent updates of the control signal.

In the past few decades it became apparent that predictive control methods possess qualities that could be utilized in more complex, nonlinear applications, possibly with much faster dynamics [14]. As more and more of these cutting edge systems (e.g. active suspension [7], gas turbine engine [8], civil aircraft [15], etc.) emerge as applications, for which RHC methods could provide a candidate solution, it is left to the system engineer to choose the particular approach from the many flavors of RHC design or possibly a combination of them, which best fits the problem at hand. A main consideration of RHC schemes is real-time implementation, i.e. whether sufficient computational resources are available to accommodate repetitive solution of the optimization problem within each sampling time interval.

This paper intends to highlight these issues in the application of three receding horizon control schemes to the longitudinal axis control of a nonlinear F-16 aircraft. The selection of the receding horizon algorithms was motivated by the following aspects of predictive controller design: process modelling, optimization method and complexity, and real-time implementability. To maintain a common ground for comparison of these methods, specific details of the optimization problem are kept the same for each approach. The results presented in Section 5 describe trade-offs that could be helpful in selecting a particular method for high-end applications.

2 F-16 modelling

The nonlinear model of the F-16 aircraft used in simulations and the problem formulation was obtained from [16] and is available at the web-site [17] as a low fidelity model. The dynamics of the continuous time aircraft model is represented as

$$\dot{x}_{NL} = f\left(x_{NL}, \delta\right) \tag{1}$$

The mathematical model uses simplified high-fidelity data from NASA Langley wind-tunnel tests conducted on a scale model of an F-16 aircraft [13]. For our investigations, only the longitudinal motion of the aircraft is considered and the states $x_{NL} \in \mathbb{R}^5$ and controls $\delta \in \mathbb{R}^2$ in the model are defined as

$$x_{NL} = [h, \theta, V_t, \alpha, q]^T, \quad \delta = [\delta_{th}, \delta_e]^T$$

where h stands for altitude [ft], θ for pitch angle [rad], V_t for total airspeed [ft/s], α for angle of attack [rad], q for pitch rate [rad/s], δ_{th} for thrust [lb] and δ_e for elevator deflection [deg]. Actuators for the control surface and engine are modelled as first-order systems, details of which can be found in [11].

2.1 Inner-loop control

The nonlinear F-16 model in (1) was augmented with an inner-loop controller based on pitch rate feedback. A benefit of the augmented system is stability of the closed-loop vehicle. Output predictions of an unstable system can cause numerical problems in optimization software [12]. This underscores the practical importance of having a stabilizing controller augment the unstable plant before RHC methods are applied (this of course is not a theoretical necessity). Another practical reason for employing an inner-loop in the receding horizon framework is that the RHC sampling rate can be reduced since the inner-loop is handling the high bandwidth disturbance and tracking requirements with its smaller sampling time implementation. This allows more computational time for the outer-loop RHC algorithm (even though the horizon lengths are expected to be longer). Furthermore, actual aircraft often come equipped with an inner-loop flight control system (most commonly stability or control augmentation systems – SAS/CAS). Even in case of an experimental aircraft, which serves as a controller testbed, flight control engineers are very reluctant to implement and test control algorithms without the existing, stabilizing inner-loop control system, which has been flight certified. Therefore, it is reasonable to assume that an inner-loop controller will augment the actual aircraft due to safety, certification or other implementation requirements.

In this paper, a pitch rate tracking linear $\mathcal{H}_\infty$-controller was chosen to provide a similar level of performance throughout a

sufficiently large flight envelope [4]. It is important to note, that the choice of the stabilizing inner-loop controller could be arbitrary. It is assumed to be developed and implemented independently of the outer-loop RHC schemes that are being investigated. This means that except for the full nonlinear RHC approach, a "grey-box" inner-loop philosophy is adopted, namely only a certain number of linearized models are assumed to be known of the inner-loop at certain flight conditions. This philosophy is sometimes motivated by the restrictions on the availability of nonlinear models that represent proprietary or otherwise sensitive material. In our opinion however, with the proposed adaptive RHC algorithm described in the next section, this approach serves as a viable alternative to the full nonlinear model-based technique as it is demonstrated in Sections 4 and 5 for the specific examples. Practical advantages of using linearized models as opposed to nonlinear ones are pointed out by other authors as well [3].

The nonlinear model in (1) is augmented with the inner-loop $\mathcal{H}_\infty$-controller to be used as a prediction model for the nonlinear RHC. This inner-loop model also represents the "actual" aircraft for the implementation of linear RHC schemes. The nonlinear model was linearized and discretized at several trim flight conditions as discussed in Section 4. The set of linearized models is used for interpolation in the adaptive RHC scheme of Section 3.2. Each linearized inner-loop model has the commanded thrust and pitch rate as inputs, and altitude, velocity, vertical acceleration (nz), actual thrust, thrust rate and elevator deflection and rate as outputs to be able to enforce actuator constraints, maneuvering limits and tracking performance. The output signals are assigned to these three objective groups denoted by u, z and y, respectively. The commanded input signals are denoted by r.

$$y = [h, V_t]^T, \quad z = nz,$$
$$u = \left[\delta_{th}, \tfrac{d\delta_{th}}{dt}, \delta_e, \tfrac{d\delta_e}{dt}\right]^T, \quad r = [\delta_{th\,cmd}, q_{cmd}]^T$$

Using the above notation, the linearized discrete-time inner-loop models have the form

$$\begin{aligned} x(k+1) &= Ax(k) + Br(k), \\ [y(k), z(k), u(k)]^T &= Cx(k) + Dr(k) \end{aligned} \tag{2}$$

3 RHC problem formulation

A brief summary of the three receding horizon control approaches compared in this paper follows.

3.1 Linear RHC

The optimization problem setup follows the formulation of [12]. The term 'linear' here refers to the prediction model used in the cost formulation. In most linear predictive controllers, the performance is specified by the following quadratic cost function to be minimized, which will also be adopted here:

$$\begin{aligned} J(k) = &\sum_{i=1}^{H_p} \left\| \hat{y}\,(k+i \mid k) - y_{ref}\,(k+i \mid k) \right\|_Q^2 + \\ &+ \sum_{i=0(\delta H_c)}^{H_c-1} \left\| \Delta r\,(k+i \mid k) \right\|_R^2 + \rho\varepsilon \end{aligned} \tag{3}$$

where $\hat{y}\,(k+i \mid k)$ is the i-step ahead prediction of the outputs based on data up to time k. H_p denotes the output prediction horizon. These predictions of the outputs are functions of future control increments $\Delta r\,(k+i \mid k)$ for $i = 0, \delta H_c, 2\delta H_c, \ldots, H_c - 1$. The integer number of samples H_c is called the control horizon, the control signal is allowed to change only at integer multiples of δH_c samples and is set to be constant for all $i \geq H_c$. This means that the future control signal is a stairstep function with steps occuring at δH_c intervals. The reference signal y_{ref} represents the desired outputs, Q and R are suitably chosen weighting matrices. The slack variable ε and its weight ρ is used for softening constraints.

In order to obtain the predictions for the signals of interest, a model of the process is needed. By using a linear model, the resulting optimization problem of minimizing $J(k)$ will be a quadratic programming (QP) problem, for which fast and numerically reliable algorithms are available. The linearized inner-loop model, developed in the previous section, is augmented with extra states to fit the formulation in this RHC scheme. Two integrators are added to convert the control changes Δr into actual controls r, one associated with thrust command and the other with pitch rate command. A simple disturbance model is incorporated to the state space description of the inner-loop model in equation (4), which assumes constant disturbances are acting on outputs. The disturbance model also serves to mitigate the effect of model mismatch. The augmented linear inner-loop model has the following form

$$\begin{aligned} \overbrace{\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \\ r(k) \end{bmatrix}}^{\hat{\xi}(k+1)} &= \overbrace{\begin{bmatrix} A & 0 & B \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}}^{\mathcal{A}} \overbrace{\begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix}}^{\hat{\xi}(k)} + \overbrace{\begin{bmatrix} B \\ 0 \\ I \end{bmatrix}}^{\mathcal{B}} \Delta r(k) \\ \underbrace{\begin{bmatrix} \hat{y}(k) \\ \hat{z}(k) \\ \hat{u}(k) \end{bmatrix}}_{\hat{w}(k)} &= \underbrace{\left[\begin{array}{c|c|c} C & \begin{matrix} I \\ 0 \end{matrix} & D \end{array}\right]}_{\mathcal{C}} \underbrace{\begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix}}_{\hat{\xi}(k)} + \underbrace{D}_{\mathcal{D}} \Delta r(k) \end{aligned} \tag{4}$$

By using successive substitution, it is straightforward to derive that the prediction model of inner-loop outputs (signals of interest) over the prediction horizon is given by equation (5).

Denote parts of the state matrices $\mathcal{C}$ and $\mathcal{D}$ in equation (5) that correspond to the predicted $\hat{y}(k)$ outputs in $\hat{w}(k)$, with a y subscript. Consider only those predicted outputs that appear in the performance index

$$\begin{aligned} \hat{y}(k) &= \mathcal{C}_y \hat{\xi}(k) + \mathcal{D}_y \Delta r(k), \\ \mathcal{Y}(k) &= \left[\hat{y}\,(k+1 \mid k), \ldots, \hat{y}\,(k+H_p \mid k)\right]^T \end{aligned}$$

using only the corresponding $\mathcal{C}_y$ and $\mathcal{D}_y$ matrices in expression (5). The prediction for these outputs has the form

$$\mathcal{Y}(k) = \Psi_y \hat{\xi}(k) + \Theta_y \Delta\mathcal{R}(k) \tag{6}$$

Substituting the predicted output in (6) into the cost function of (3), we get a quadratic expression in terms of the control changes $\Delta\mathcal{R}(k)$:

$$J(k) = \Delta\mathcal{R}(k)^T \mathcal{H} \Delta\mathcal{R}(k) - \Delta\mathcal{R}(k)^T \mathcal{G} + const + \rho\varepsilon \tag{7}$$

where $\mathcal{H}_k = \Theta_y^T Q_e \Theta_y + R_e$, $\mathcal{G}_k = 2\Theta_y^T Q_e \mathcal{E}(k)$, $const = \mathcal{E}^T(k) Q_e \mathcal{E}(k)$ and $\mathcal{E}(k)$ is defined as a tracking error between the future target trajectory and the free response of the system, i.e. $\mathcal{E}(k) = \mathcal{Y}_{ref}(k) - \Psi_y \hat{\xi}(k)$. Q_e and R_e are block diagonal matrices of appropriate dimensions with Q and R on the main diagonal, respectively.

$$\underbrace{\begin{bmatrix} \hat{w}(k+1 \mid k) \\ \hat{w}(k+2 \mid k) \\ \vdots \\ \hat{w}(k+H_c \mid k) \\ \hat{w}(k+H_c+1 \mid k) \\ \vdots \\ \hat{w}(k+H_p \mid k) \end{bmatrix}}_{\mathcal{W}(k)} = \underbrace{\begin{bmatrix} \mathcal{CA} \\ \mathcal{CA}^2 \\ \vdots \\ \mathcal{CA}^{H_c} \\ \mathcal{CA}^{H_c+1} \\ \vdots \\ \mathcal{CA}^{H_p} \end{bmatrix}}_{\Psi} \hat{\xi}(k) + \underbrace{\begin{bmatrix} \mathcal{CB} & \mathcal{D} & \cdots & 0 \\ \mathcal{CAB} & \mathcal{CB} & \mathcal{D} & \vdots \\ \vdots & \vdots & \ddots & \ddots \\ \mathcal{CA}^{H_c-1}\mathcal{B} & \mathcal{CA}^{H_c-2}\mathcal{B} & \cdots & \mathcal{CB} \\ \mathcal{CA}^{H_c}\mathcal{B} & \mathcal{CA}^{H_c-1}\mathcal{B} & \cdots & \mathcal{CAB} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{CA}^{H_p-1}\mathcal{B} & \mathcal{CA}^{H_p-2}\mathcal{B} & \cdots & \mathcal{CA}^{H_p-H_c}\mathcal{B} \end{bmatrix}}_{\Theta} \underbrace{\begin{bmatrix} \Delta r(k \mid k) \\ \vdots \\ \Delta r(k+H_c-1 \mid k) \end{bmatrix}}_{\Delta\mathcal{R}(k)} \tag{5}$$

As in most applications, there are level and rate limits on actuators. These are enforced as hard constraints

$$\underline{u} \leq \hat{u}(k+1 \mid k), \ldots, \hat{u}(k+H_p \mid k) \leq \overline{u} \tag{8}$$

since the RHC algorithm has almost direct control over some of them (thrust level and rate) and the effect of pitch rate demand on elevator deflection and rate is also known with high accuracy. Another type of constraint is also considered in this specific application example represented by certain maneuvering limits on the aircraft. These limits might be system-state dependent or change according to different stages of a mission. We assume the existence of such limitations on the vertical acceleration (nz) of the aircraft in Section 5.3. It is vital that these limits are treated as soft constraints, since disturbances and model mismatch can easily lead to infeasibility problems if hard constraints are put on these type of output signals.

Constraint softening is accomplished by introducing an additional slack variable that allows some level of constraint violation if no feasible solution exists

$$\underline{z} - \varepsilon \leq \hat{z}(k+1 \mid k), \ldots, \hat{z}(k+H_p \mid k) \leq \overline{z} + \varepsilon, \quad 0 \leq \varepsilon \tag{9}$$

Using the linear prediction model in (5), all of the constraints in (8) and (9) can be posed as linear constraints on the optimization variables $\Delta\mathcal{R}$ and ε. Finally, the QP to be solved at each time step has the following form

$$\begin{aligned} \min_{\Delta\mathcal{R},\,\varepsilon} \quad & \Delta\mathcal{R}^T\mathcal{H}\Delta\mathcal{R} - \Delta\mathcal{R}^T\mathcal{G} + const + \rho\varepsilon \\ \text{s. t.} \quad & \begin{bmatrix} \Omega_{hard} \\ \Omega_{soft} \end{bmatrix} \Delta\mathcal{R} \leq \begin{bmatrix} \omega_{hard} \\ \omega_{soft} \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix} \\ & 0 \leq \varepsilon \end{aligned} \tag{10}$$

3.2 Adaptive RHC

A natural extension of the linear RHC is to base the prediction at a certain time step on a linear model that best describes the plant (inner-loop) at the actual flight condition, if flight condition dependent linear models are available for prediction. A fixed LTI model is used over the entire prediction horizon but it is updated according to the values of some flight condition dependent scheduling parameters every time the horizon is propagated and the optimization is resolved based on new measurement data. This approach leads to the same QP problem to be solved as in (10), however the state matrices describing the internal model change in each implementation cycle according to their current values: $\mathcal{A}_k$, $\mathcal{B}_k$, $\mathcal{C}_k$, $\mathcal{D}_k$. This flight condition dependent description of the inner-loop dynamics could be obtained either by freezing the scheduling parameters of a quasi-LPV model [10], or interpolating over a database of linearized models. The latter approach is used in all the examples of this paper to illustrate the general applicability of this approach motivated by the remarks on restricted model availability in Section 2.

We note if an accurate prediction of the parameters that the linear models depend on is available, this would allow for the prediction model to vary over the prediction horizon. The optimization problem could still be formulated as a quadratic program using different state matrices of the internal model at each time step. Obtaining a reasonable prediction of the scheduling parameters is not always easy, there are several alternative approaches that might be used for this purpose [11].

3.3 Nonlinear RHC

In general, the discrete time version of a nonlinear trajectory optimization problem can be posed in Bolza form [5] as

$$\min_{r(i)} \quad J = \phi[x(N)] + \sum_{i=0}^{N-1} L[x(i), r(i), i] \tag{11a}$$

using discrete time nonlinear system dynamics, initial conditions and terminal constraints defined as

$$x(i+1) = f[x(i), r(i), i], \quad x(0) = x_0, \quad \psi[x(N)] = 0. \tag{11b}$$

The receding horizon principle is realized by solving the Nonlinear Programming (NP) optimization problem (11) for the sequence of control vectors $r(i)$ for $i = 0, \ldots, N-1$, then implementing the first control values in the sequence and resolving the problem again at each subsequent time step, when new state estimate information is available.

The standard form (11) of the nonlinear RHC problem is modified slightly to match the control space parametrization and cost function of the QP-based linear approaches as closely as possible. The discretized version of the nonlinear inner-loop model described in Section 2 is augmented with integrators on the input to redefine the decision variables of the optimization problem to changes in control $\Delta r(i)$. A constant disturbance acting on the respective outputs is also added to the model similarly to the equations in (4). The decision variables $\Delta r(i)$ of the reformulated optimization problem were selected to be exactly those values that the QP-based methods optimize over with the same sampling time interval. The cost function is calculated exactly as the quadratic expression in (3).

The modified nonlinear programming problem outlined above is solved using Sequential Quadratic Programming (SQP) implemented by the NPSOL optimization software package [9].

There are many alternatives to the problem formulation in (11), e.g. using a continuous time nonlinear model and parametrizations of the control signals (selection of basis functions). However, it should be emphasized that the choice of control space formulation and cost function is intended to serve purely as a basis for comparison between the QP based linear, adaptive RHC schemes and the nonlinear RHC. In general, solving the nonlinear RHC problem requires more elaborate development of a suitable control space and cost function formulation, for which the work in [2] provides detailed guidelines using B-spline functions to parameterize the control space and other important implementation issues of the nonlinear RHC algorithm based on NPSOL that are omitted here for brevity.

Figure 1 illustrates the general receding horizon control setup common in all three approaches described in this section.

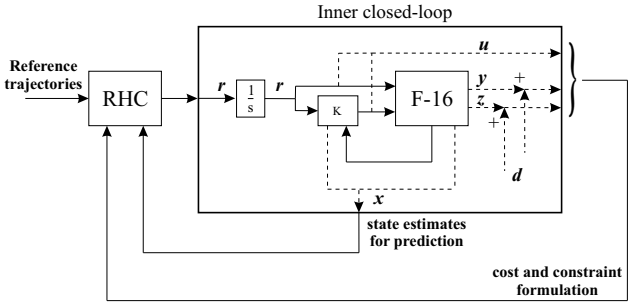


Figure 1: General RHC framework of the F-16 control problem

4 Simulation details

A database of linearized inner-loop models was created to be used by the interpolation routine in the adaptive RHC scheme. Dynamic pressure $(\bar{q})$ and Mach number (M) were selected as flight condition dependent scheduling parameters that determine which linear model to use for prediction. The nonlinear aircraft dynamics was linearized at steady level flight trim condition, at 38 different points of the flight envelope. More details about the flight envelope and the interpolation over the linearized models can be found in [11].

The augmented linear models in equation (4) of the inner-loop had 17 states and were discretized at 20 Hz [11]. All three RHC schemes were implemented with a 50 ms sampling time, 4 second prediction horizon $(H_p = 80)$ and 1.5 second control horizon with future control changes at every 0.5 second $(H_c = 31, \delta H_c = 10)$. The discrete time nonlinear model of the inner-loop, used for prediction in the nonlinear RHC scheme, was discretized with a sampling time of 1 ms. The values of the weighting matrices Q and R in the cost formulation were tuned based on the linear RHC scheme.

Three simulation scenarios are presented to illustrate the benefits and drawbacks of the RHC schemes under investigation. The first example is a disturbance rejection scenario, in which the objective is to keep steady level flight at trim altitude and velocity in the presence of vertical wind gusts that occur in the form of a 50 ft/s step disturbance on velocity at 5 seconds and a 100 ft step disturbance on altitude at 50 seconds into the simulation. (The single LTI model based linear RHC scheme uses a prediction model that corresponds to a different flight condition to demonstrate the inherent problems with this approach.)

In contrast to the first example, which intends to investigate local behaviour of the RHC methods, the second example represents a large envelope comparison of the three approaches.

The third example aims at pointing out the aggressive maneuvering capabilities enabled by the adaptive RHC approach, as well as system-state dependent constraint enforcement represented by total airspeed dependent varying vertical acceleration limits. This simulation demonstrates two scenarios. First, a relatively aggressive reference altitude and velocity trajectory is flown without any maneuvering constraints on vertical acceleration. Then soft constraints are enforced during the same maneuver on vertical acceleration to keep the aircraft within velocity dependent upper and lower acceleration limits, which might be motivated by the stall characteristics of the aircraft.

In all of these examples, the general goal of the outer-loop RHC controllers is to accomplish "higher level" control objectives, by exploiting *a priori* reference information. The controllers have to ensure that the aircraft's inputs are held within saturation limits in the presence of wind gusts and respect flight envelope constraints and system-state dependent maneuver limits by acting as a system/mission-state dependent variable inner-loop command prefilter.

5 Results

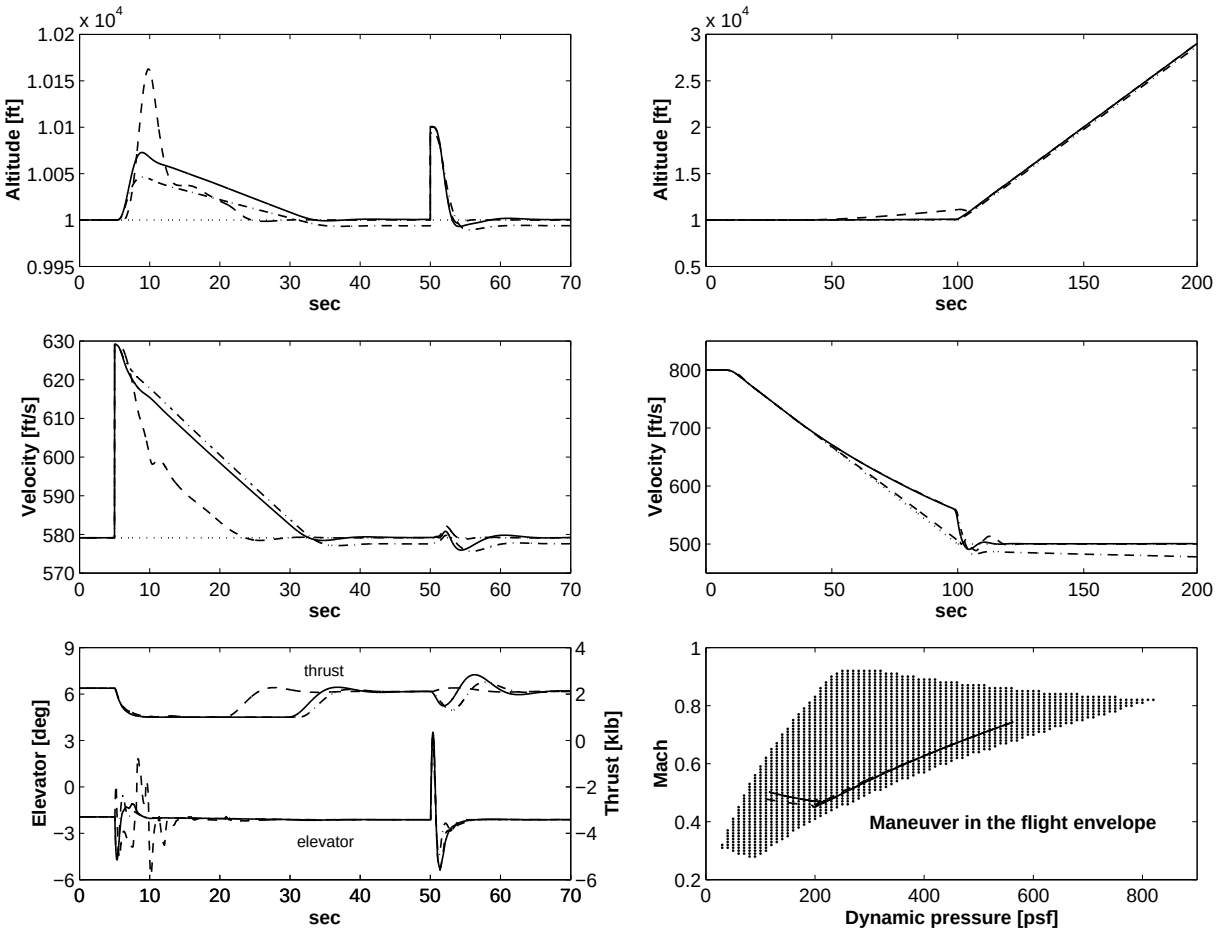
All RHC simulations were run on a 1.2 GHz Pentium III machine running RedHat Linux. Using the specific parameters described in Section 4 and [11] to formulate the optimization problem, the resulting QP had 8 decision variables (9 with the slack variable in example 3). The number of linear constraints were 664 (826 with the soft constraints).

5.1 Example 1

The results of the disturbance rejection scenario are shown in Figure 2(a) for the three approaches. Comparing the achieved performance of the three schemes, it is interesting to note that the nonlinear RHC approach achieves the fastest disturbance rejection decay time. On the other hand, in case of the velocity disturbance at 5 seconds, the peak altitude error is larger than with the other two methods. The nonlinear RHC approach results in a larger angle of attack disturbance response and the controller also uses larger and more oscillatory control effort. This could be attributed to the numerical problems mentioned earlier in Section 3, which stem from the fact that the control space and cost formulation were selected for comparison purposes and not according to other established guidelines [2], which would alleviate most of the numerical difficulties. Nevertheless, the nonlinear RHC scheme handles the altitude disturbance occurring at 50 seconds with faster settling time and significantly less use of control authority than the other two methods, achieving an overall lower cost value throughout the simulation. The linear RHC leads to steady state error, whereas the adaptive approach provides acceptable performance even in comparison with the nonlinear technique.

The linear RHC scheme requires the least amount of computational time, the underlying QP problem can be solved analytically, if no constraints are active. This means that calculation of the next control signal value takes approximately 0.03 second to complete (with 0.05 second sampling time) on the platform used for computations. If constraints are active, Matlab's QP solver is used, which provided a solution in 0.05 second on average. These numbers indicate that this approach is readily implementable in real time, even using the Matlab environment. Implementing the algorithm in C code would speed up execution by a factor of ten.

The optimization problem complexity is exactly the same in the adaptive RHC scheme as in the linear one, i.e. the QP can



(a) Example 1 results.

(b) Example 2 results.

Figure 2: Simulation examples 1 and 2 (reference: dotted , linear RHC: dash-dot, adaptive RHC: solid, nonlinear RHC: dash)

be solved in approximately real-time, even if constraints are active. However, a significant computational overhead comes from the need for interpolation over the linearized inner-loop models. The amount of time this prediction model lookup requires depends heavily on the implementation of the interpolation routine and the size of the linear model database. The Matlab interpolation algorithm (`griddata`) performs this task in approximately 1 second. Considering the amount of speed-up gained from C implementation and other possible avenues of decreasing computation time (using different interpolation routines, less number of linear models, variable time interval formulation [15] or a number of other options), real-time implementation of this scheme is also deemed achievable.

Solving the nonlinear programming problem involved in the nonlinear RHC scheme places a significantly larger computational burden on the implementation platform, with average CPU times of 40 seconds for each 0.05 second sample interval. Clearly, this method is not applicable to fast and complex systems with the current state-of-the-art computational tools.

5.2 Example 2

In this large envelope example, depicted in Figure 2(b), the linear and adaptive RHC schemes behave similarly by following the reference altitude with slowly increasing velocity error and lower saturated thrust control value. However, a large velocity error appears in the climb phase of the maneuver using the linear RHC. The adaptive approach clearly outperforms the single linear model RHC. Errors introduced by the single model RHC scheme become more significant as larger excursions are made in the flight envelope. The main underlying reason for this is the absence of a trim map in the single model approach, however mismatch in plant dynamics is also a strong contributor to these errors, especially in the low dynamic pressure and Mach region of the flight envelope. The nonlinear RHC method, interestingly enough, trades off altitude errors for better velocity tracking and uses larger control values.

5.3 Example 3

The third example shown in Figure 3, was performed only using the adaptive RHC approach and demonstrates that aggressive maneuvers, as well as system-state dependent maneuvering limits can be enforced by the flexibility offered in this methodology. (The flight path angle peaks near 25 degrees during the maneuver and angle of attack approaches 15 degrees in the unconstrained case, and 8 degrees in the constrained case.) It is interesting to note, that in the case of soft constraints on vertical acceleration, the aircraft violates the lower limit to a small extent between 15 and 20 seconds, which indicates that the actual maneuver would have led to infeasibility if hard constraints were imposed on vertical acceleration. This is due to the ∞-norm (maximum violation) penalty on constraint violations (as shown in (3) and (9)) and has been verified by running the algorithm with hard constraints. This 'exact penalty' method means that constraint violations will not occur unless no feasible solution exists to the original 'hard' problem. If a

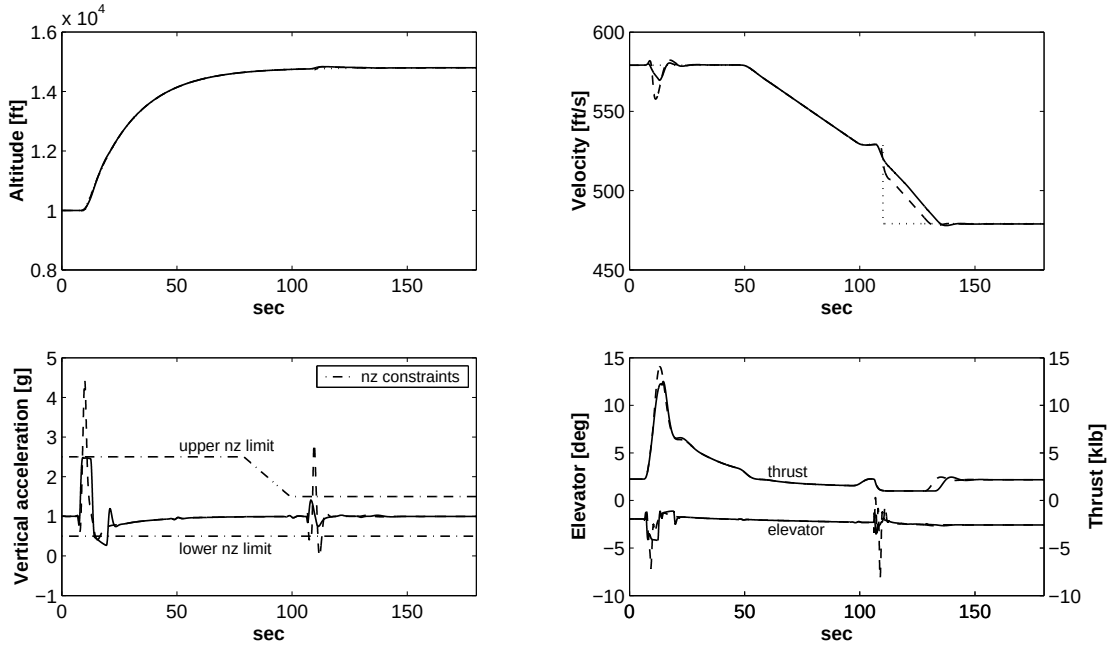


Figure 3: Simulation results of example scenario 3 (reference: dotted, adaptive RHC w/o constraints: dash-dot, adaptive RHC with soft constraints: solid)

feasible solution exists, the same solution will be obtained as with the 'hard' formulation. The simulation also demonstrates how the controller enforces smaller upper limits, as the total airspeed is reduced.

6 Conclusions

As simulation results demonstrate for the F-16 longitudinal axis control example, computationally efficient receding horizon schemes can be developed for highly nonlinear, complex systems based on linear prediction models to keep the optimization problem manageable. Using flight condition dependent linearized models or a quasi-LPV system for prediction, the modest complexity of the predictive control problem can still be retained (QP) with improved accuracy and extended operation limits. Even though additional computational overhead is introduced by interpolating over linearized models, our experience suggests that real-time implementation is plausible. The proposed adaptive RHC scheme has the required flexibility that these type of applications, such as aerospace systems often require. In this aspect, it provides significantly more than single LTI model based linear RHC schemes, with performance comparable to a full nonlinear RHC solution. On-line constraint modification allows straightforward incorporation of system-state dependent, and time-varying constraints.

Acknowledgements

This work was funded by DARPA under the Software Enabled Control program, Dr. John Bay Program Manager. The contract number is USAF/AFMC F33615-99-C-1497, Dale Van Cleave is the Technical Monitor.

References

[1] Bemporad, A. and M. Morari, "Robust Model Predictive Control: A Survey", *Robustness in Identification and Control*, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1999, pp. 207-226.

[2] Bhattacharya, R., G.J. Balas, M.A. Kaya and A. Packard, "Nonlinear Receding Horizon Control of an F-16 Aircraft", *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, 2002, pp. 924-931.

[3] Blet, N., D. Megías, J. Serrano and C. de Prada, "Nonlinear MPC Versus MPC Using On-Line Linearisation - A Comparative Study", *15th IFAC World Congress*, 2002.

[4] Blue, P., D. Odenthal and M. Muhler, "Designing Robust Large Envelope Flight Controllers for High-Performance Aircraft", *AIAA Guidance, Navigation, and Control Conference*, 2002.

[5] Bryson, A.E.Jr., *Dynamic Optimization*, Addison-Wesley, 1999.

[6] Camacho, E.F. and C. Bordons, "Model Predictive Control in the Process Industry", *Advances in Industrial Control*, Springer, London, 1995.

[7] Donahue, M.D., *Implementation of an Active Suspension, Preview Controller for Improved Ride Comfort*, M.S. Thesis, University of California at Berkeley, 2001.

[8] Fuller, J. and R. Meisner, "Optimization-based Control for Flight Vehicles", *AIAA Guidance, Navigation, and Control Conference*, 2000.

[9] Gill, P., W. Murray, M. Saunders and M. Wright, NPSOL – Nonlinear Programming Software, Stanford Business Software, Inc., Mountain View, CA.

[10] Huzmezan, M. and J.M. Maciejowski, "Reconfiguration and Scheduling in Flight Using Quasi-LPV High-Fidelity Models and *MBPC* Control", *American Control Conference*, 1998.

[11] Keviczky, T. and G.J. Balas, "Software Enabled Flight Control Using Receding Horizon Techniques", *AIAA Guidance, Navigation, and Control Conference*, 2003.

[12] Maciejowski, J.M., *Predictive Control with Constraints*, Prentice Hall, 2002.

[13] Nguyen, L., M. Ogburn, W. Gilbert, K. Kibler, P. Brown and P. Deal, "Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane with Relaxed Longitudinal Static Stability", NASA Technical Report 1538, Dec. 1979.

[14] Papageorgiou, G., K. Glover, M. Huzmezan and J.M. Maciejowski, "A Combined *MBPC*/$\mathcal{H}_\infty$ Automatic Pilot for a Civil Aircraft", *American Control Conference*, 1997, pp. 118-122.

[15] Schram, G., R.A.J. de Vries, E. Cevaal and T.J.J. van den Boom, "Predictive Control Applied to a Civil Aircraft Benchmark Problem", *European Control Conference*, 1997.

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

[17] `http://www.aem.umn.edu/people/faculty/balas/ /darpa_sec/SEC.Software.html`