Receding Horizon Control of an F-16 Aircraft: a Comparative Study *

Tamás Keviczky * and Gary J. Balas

Department of Aerospace Engineering and Mechanics, University of Minnesota, 110 Union Street S.E., Minneapolis, MN 55455, USA

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 a scheduled 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 scheduled 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.

Key words: Predictive control, Flight control, Optimization

1 Introduction

Receding horizon control (RHC) techniques, also known as model predictive control or model based predictive control methods, have been in the limelight of significant research efforts, motivated by several successful industrial applications (Camacho and Bordons, 1995; Bemporad and Morari, 1999; Allgöwer and Zheng, 2000; Maciejowski, 2002). 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

* Research supported under DARPA Software Enabled Control program, contract number USAF/AFMC F33615-99-C-1497, program manager Dr. John Bay, technical contract monitor Dale Van Cleave.

* Corresponding author. Tel.: +1-612-625-6561, Fax.: +1-612-626-1558

Email addresses: keviczky@aem.umn.edu (Tamás Keviczky), balas@aem.umn.edu (Gary J. Balas).
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. Online optimization performed within RHC schemes allow control engineers to incorporate in the problem formulation and address physical process constraints explicitly. Furthermore, the flexibility of this solution approach lends itself to online reconfiguration and adaptation to changes in the environment such as updates to process models and constraints. These characteristics make receding horizon control schemes attractive for potential application in areas, where process constraints and performance are critical, yet currently addressed with classical control techniques.

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 (Papageorgiou et al., 1997). As more and more of these cutting edge systems (e.g. active suspension by Donahue (2001), gas turbine engine by Fuller and Meisner (2000), civil aircraft by Schram et al. (1997), 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. This control problem represents a high bandwidth application with the objective of providing altitude and velocity tracking autopilot functions using a pitch-rate tracking inner-loop legacy controller. The selection of the receding horizon algorithms was motivated by the following aspects of predictive controller design: process modeling, 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 modeling

The nonlinear model of the F-16 aircraft (Fig. 1) used in simulations and the problem formulation was obtained from the book by Stevens and Lewis (1992) and is available at the web-site (Russell, 2003) as a low fidelity model. The
The dynamics of the continuous time aircraft model is represented as

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

(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 (Nguyen et al., 1979). 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

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

(2)

where $h$ stands for altitude [ft], $\theta$ for pitch angle [rad], $V_t$ for total airspeed [ft/s], $\alpha$ for angle of attack [rad], $q$ for pitch rate [rad/s], $\delta_{th}$ for thrust [lb] and $\delta_e$ for elevator deflection [deg]. Actuators for the control surface and engine are modeled as first-order systems, with characteristics described in Table 1. The aerodynamic data are valid up to Mach 0.9 and angle of attack range of $-10$ deg $\leq \alpha \leq 45$ deg.

### 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 be numerically very inaccurate and cause numerical problems in optimization.
software (Maciejowski, 2002). Hence, an unstable prediction model should be avoided whenever possible. 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 inner-loop controller was chosen to provide a similar level of performance throughout a sufficiently large flight envelope. This inner-loop controller is a linear $H_\infty$-controller designed with parameter space techniques (Blue et al., 2001, 2002) and implemented in a two-degree-of-freedom structure as two second order transfer functions with a first order command prefilter. The two inputs to the controller are the commanded and measured pitch rate, the output is elevator deflection change from the trim value:

$$\Delta \delta_e = K(s) \begin{bmatrix} q_{dem} \\ q_{meas} \end{bmatrix}. \quad (3)$$

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 scheduled RHC algorithm described in the following section, this approach serves as a viable alternative to a 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 (Blet et al., 2002).
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 scheduled RHC scheme of Section 3.1. Each linearized inner-loop model has the commanded thrust and pitch rate as inputs, and altitude, velocity, vertical acceleration ($n_z$), 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 $r$.

\[
y = \begin{bmatrix} h \\ V_t \end{bmatrix}, \quad z = n_z, \quad u = \begin{bmatrix} \delta_{th} \\ \frac{\partial \delta_{th}}{\partial t} \\ \delta_e \\ \frac{\partial \delta_e}{\partial t} \end{bmatrix}, \quad r = \begin{bmatrix} \delta_{th\,cmd} \\ q_{cmd} \end{bmatrix}
\]  

Each trim flight condition was characterized by the corresponding dynamic pressure ($\tilde{q}$) and Mach number ($M$). Denoting this vector of parameters with $\varrho(k) = [\tilde{q}(k) \ M(k)]^T$, the linearized discrete-time inner-loop models have the form

\[
x(k + 1) = A_k x(k) + B_k r(k) \\
y(k) \\
z(k) = C_k x(k) + D_k r(k), \\
u(k)
\]

where $A_k \in \mathbb{R}^{12 \times 12}$, $B_k \in \mathbb{R}^{12 \times 2}$, $C_k \in \mathbb{R}^{7 \times 12}$, $D_k \in \mathbb{R}^{7 \times 2}$ and the flight condition dependency of the prediction models is indicated by the subscript $k$, meaning

\[
A_k = A(\varrho(k)), \quad B_k = B(\varrho(k)), \\
C_k = C(\varrho(k)), \quad D_k = D(\varrho(k)).
\]

3 RHC problem formulation

A summary of the three receding horizon control approaches compared in this paper follows.
3.1 Linear and Scheduled RHC

This section describes two receding horizon methods based on linear prediction models. The terms ‘linear’ and ‘scheduled’ are used to distinguish between the two methods based on the prediction model used in the cost formulation. Both approaches use the same problem formulation, which will be described in this section. They differ only in the prediction model, which is a single linear time-invariant (LTI) model within the ‘Linear RHC’ approach, and linear parameter-varying (LPV) model within the ‘Scheduled RHC’ approach. Some additional remarks about each method are given at appropriate points in the following description of their common problem formulation.

The optimization problem setup follows the formulation of Maciejowski (2002) with some modifications. In most linear predictive controllers, the performance is specified by the following quadratic cost function to be minimized, which will also be adopted here:

\[
J(k) = \sum_{i=1}^{H_p} \|\hat{y}(k+i | k) - y_{\text{ref}}(k+i | k)\|_Q^2 + \sum_{i=0(\delta_{H_c})}^{H_c-1} \|\Delta r(k+i | k)\|_R^2 + \rho \varepsilon, \tag{6}
\]

where \(\hat{y}(k+i | k)\) is the \(i\)-step ahead prediction of the outputs based on data up to time \(k\). \(H_p\) denotes the number of steps in the output prediction horizon. These predictions of the outputs are functions of future control increments \(\Delta r(k+i | k)\) 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 the “blocking parameter” \(\delta_{H_c}\), which represents a “uniform blocking policy”. The control signal is set to be constant for all \(i \geq H_c\). This means that the future control signal has the form of a stairstep function with steps occurring at \(\delta_{H_c}\) intervals. The reference signal \(y_{\text{ref}}\) represents the desired outputs, \(Q\) and \(R\) are suitably chosen weighting matrices. The slack variable \(\varepsilon\) and its weight \(\rho\) is used for softening constraints. The exact purpose of the slack variable and weight in the problem formulation will be clarified shortly.

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 \(\Delta r\) 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.
\[
\begin{pmatrix}
\hat{w}(k+1 | k) \\
\hat{w}(k+2 | k) \\
\vdots \\
\hat{w}(k+H_c | k) \\
\hat{w}(k+H_c+1 | k) \\
\vdots \\
\hat{w}(k+H_p | k)
\end{pmatrix}
= \begin{pmatrix}
C_kA_k & C_kA_k^2 & \cdots & C_kA_k^{H_c} & C_kA_k^{H_p-1} & C_kA_k^{H_p-H_c}B_k \\
C_kA_k^2 & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_kA_k^{H_p-1} & C_kA_k^{H_p-2}B_k & \cdots & C_kA_kB_k \\
C_kA_k^{H_p-1} & C_kA_k^{H_p-2}B_k & \cdots & C_kA_kB_k \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_kA_k^{H_p-H_c}B_k & C_kA_k^{H_p-H_c}B_k & \cdots & C_kA_kB_k
\end{pmatrix}
\begin{pmatrix}
\xi(k) \\
\Delta r(k | k) \\
\Delta r(k+H_c-1 | k)
\end{pmatrix}
\tag{9}
\]

in equation (8), which assumes constant disturbances are acting on outputs. The constant disturbance estimates \(\hat{d}(k)\) are obtained by using the difference between measured and predicted outputs in the following model:

\[
\hat{d}(k+1) = \hat{d}(k) + \begin{pmatrix}
y(k) - \hat{y}(k | k-1) \\
z(k) - \hat{z}(k | k-1)
\end{pmatrix}.
\tag{7}
\]

The disturbance model also serves to mitigate the effect of model mismatch. The augmented linear inner-loop model has the following form

\[
\begin{pmatrix}
\hat{x}(k+1) \\
\hat{d}(k+1) \\
r(k)
\end{pmatrix}
= \begin{pmatrix}
A_k & 0 & B_k \\
0 & I_3 & 0 \\
0 & 0 & I_2
\end{pmatrix}
\begin{pmatrix}
\hat{x}(k) \\
\hat{d}(k) \\
r(k-1)
\end{pmatrix}
+ \begin{pmatrix}
B_k \\
0 \\
I_2
\end{pmatrix}
\Delta r(k)
\]

\[
\begin{pmatrix}
\hat{y}(k) \\
\hat{z}(k) \\
\hat{u}(k)
\end{pmatrix}
= \begin{pmatrix}
I_2 & 0_{2\times1} & I_1 \\
0_{4\times3} & D_k \\
0_{4\times3} & \xi(k)
\end{pmatrix}
\begin{pmatrix}
\hat{x}(k) \\
\hat{d}(k) \\
r(k-1)
\end{pmatrix}
+ \begin{pmatrix}
D_k \\
D_k
\end{pmatrix}
\Delta r(k)
\]

where the state matrices of the prediction model have the following dimensions:

\[A_k \in \mathbb{R}^{17 \times 17}, B_k \in \mathbb{R}^{17 \times 2}, C_k \in \mathbb{R}^{7 \times 17}, D_k \in \mathbb{R}^{7 \times 2}.
\]

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 (9).
Denote parts of the state matrices \( C_k \) and \( D_k \) in equation (9) that correspond to the predicted \( \hat{y}(k) \) outputs in \( \hat{w}(k) \), with an additional \( y \) subscript

\[
\begin{align*}
C_k &= \begin{bmatrix} C_{ky} \\ C_{kz} \\ C_{ku} \end{bmatrix}, & D_k &= \begin{bmatrix} D_{ky} \\ D_{kz} \\ D_{ku} \end{bmatrix}.
\end{align*}
\]

Consider only those predicted outputs that appear in the performance index

\[
\begin{align*}
\hat{y}(k) &= C_{ky} \dot{\xi}(k) + D_{ky} \Delta r(k), \\
\mathcal{Y}(k) &= [\hat{y}(k+1 \mid k), \ldots, \hat{y}(k+H_p \mid k)]^T,
\end{align*}
\]

using only the corresponding \( C_{ky} \) and \( D_{ky} \) matrices in expression (9). The prediction for these outputs has the form

\[
\mathcal{Y}(k) = \Psi_{ky} \dot{\xi}(k) + \Theta_{ky} \Delta \mathcal{R}(k). \tag{10}
\]

Substituting the predicted output in (10) into the cost function of (6), we get a quadratic expression in terms of the control changes \( \Delta \mathcal{R}(k) \):

\[
J(k) = \Delta \mathcal{R}(k)^T \mathcal{H}_k \Delta \mathcal{R}(k) - \Delta \mathcal{R}(k)^T \mathcal{G}_k + \text{const} + \rho \varepsilon, \tag{11}
\]

where

\[
\begin{align*}
\mathcal{H}_k &= \Theta_{ky}^T Q_e \Theta_{ky} + R_e, & \mathcal{G}_k &= 2 \Theta_{ky}^T Q_e \mathcal{E}(k), \\
\text{const} &= \mathcal{E}^T(k) Q_e \mathcal{E}(k),
\end{align*}
\]

and \( \mathcal{E}(k) \) is defined as a tracking error between the future target trajectory and the free response of the system, i.e. \( \mathcal{E}(k) = \mathcal{Y}_{ref}(k) - \Psi_{ky} \dot{\xi}(k) \). \( Q_e \) and \( R_e \) are block diagonal matrices of appropriate dimensions with \( Q \) and \( R \) on the main diagonal, respectively. (These could be chosen parameter-dependent also.)

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

\[
u \leq \hat{u}(k+1 \mid k), \ldots, \hat{u}(k+H_p \mid k) \leq \pi, \tag{12}
\]

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. Actuator level and rate constraints are also implemented in the nonlinear aircraft model used for implementation. Another type of constraint is also considered in this specific application example represented by certain maneuvering limits on the aircraft. These limits that 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 \( (n_z) \) 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

\[
\dot{z} - \varepsilon \leq \dot{z} (k + 1 \mid k), \ldots, \dot{z} (k + H_p \mid k) \leq \dot{z} + \varepsilon
\]

\[
0 \leq \varepsilon.
\]  

(13)

It is beneficial to use an \( \infty \)-norm (maximum violation) penalty on constraint violations (as shown in (6) and (13)), because it gives an “exact penalty” method if the weight \( \rho \) is large enough (Kerrigan and Maciejowski, 2000). This means that constraint violations will not occur unless no feasible solution exists to the original “hard” problem. If a feasible solution exists, the same solution will be obtained as with the “hard” formulation. Using the linear prediction model in (9), all of the constraints in (12) and (13) can be posed as linear constraints on the optimization variables \( \Delta R \) and \( \varepsilon \). Finally, the QP to be solved at each time step has the following form

\[
\begin{align*}
\min_{\Delta R, \varepsilon} & \quad \Delta R^T H_k \Delta R + \Delta R^T G_k + \text{const} + \rho \varepsilon \\
\text{s. t.} & \quad \begin{bmatrix}
\Omega_{k,\text{hard}} \\
\Omega_{k,\text{soft}}
\end{bmatrix} \Delta R \leq \begin{bmatrix}
\omega_{k,\text{hard}} \\
\omega_{k,\text{soft}}
\end{bmatrix} + \begin{bmatrix}
0 \\
\varepsilon
\end{bmatrix} \\
& \quad 0 \leq \varepsilon.
\end{align*}
\]  

(14)

3.1.1 Remarks

The problem formulation presented in this section is a natural extension of a fixed LTI model based RHC. The prediction at a certain time step is based on a linear model that best describes the plant (inner-loop) at the actual flight condition, assuming that 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 flight condition dependent scheduling parameters \( \varrho(k) \) every time the horizon is propagated and the optimization is resolved based on new measurement data. This approach leads to the QP problem in (14), and the state matrices describing the internal model change in each implementation cycle according to their current values: \( A_k, B_k, C_k, D_k \). This flight condition dependent description of the inner-loop dynamics could be obtained either by freezing the scheduling parameters of a quasi-LPV model (Huzmezan and Maciejowski, 1998), or interpolating over a database of linearized models. The latter approach is used in this paper to illustrate the
general applicability of this approach motivated by the remarks on restricted model availability in Section 2.

We note that if an accurate prediction of the parameters upon which the linear models depend is available, then a varying prediction model over the horizon is allowable. 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, one could experiment with solving the problem first with the fixed LTI model based RHC method and use the solution as the prediction for the scheduling parameters. Our investigations indicate that this extra effort does not lead to significant improvement for the specific application example and horizon lengths considered. Moreover, even though the optimization problem complexity is retained, the additional computational overhead from the large number of interpolations was significant enough to undermine real-time implementation of these ideas.

3.2 Nonlinear RHC

In general, the discrete time version of a nonlinear trajectory optimization problem can be posed in Bolza form (Bryson, 1999) as

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

(15a)

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

\[
x(i + 1) = f [x(i), r(i), i], \quad x(0) = x_0, \quad \psi [x(N)] = \mathcal{X}_f.
\]

(15b)

The receding horizon principle is realized by solving the Nonlinear Programming (NP) optimization problem (15) 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 (15) 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 (8). 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. No terminal constraints were used ($\mathcal{X}_f = \mathbb{R}^{17}$) in (15b) and the cost function is calculated exactly as the quadratic expression in (6), which means that the terminal cost was set to zero ($\phi[x(N)] = 0$) in (15a).

The modified nonlinear programming problem outlined above is solved using Sequential Quadratic Programming (SQP) implemented by the NPSOL optimization software package (Gill et al., 1998).

There are many alternatives to the problem formulation in (15), 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, scheduled 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 Bhattacharya et al. (2002) 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, which are omitted here.

Figure 2 illustrates the general receding horizon control setup common in all three approaches described in this section.

![Fig. 2. General RHC framework of the F-16 control problem.](image)

4 Simulation details

A database of linearized inner-loop models was created to be used by the interpolation routine in the scheduled RHC scheme. Based on the validity range of the aerodynamic coefficients in the aircraft model, the operating
flight envelope was chosen to be between 5000 to 40000 ft in altitude and 300 to 900 ft/s in true airspeed. Dynamic pressure ($\tilde{q}$) and Mach number ($M$) were selected as flight condition dependent scheduling parameters that determine which linear model to use for prediction. The flight envelope is shown in Fig. 3 in terms of dynamic pressure and Mach. The nonlinear aircraft dynamics were linearized at steady level flight trim condition, at 38 different points of the flight envelope. Given a $\tilde{q} - M$ value pair, triangular interpolation is performed over the coefficients of the inner-loop state matrices based on the grid depicted in Fig. 3.

![Fig. 3. F-16 model flight envelope with the interpolation grid of linearized models.](image)

Level and rate limits, as well as time constants of the actuators used in the nonlinear aircraft model are shown in Table 1.

<table>
<thead>
<tr>
<th>Actuator</th>
<th>Upper level limit</th>
<th>Lower level limit</th>
<th>Upper rate limit</th>
<th>Lower rate limit</th>
<th>Time constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throttle, $\delta_{th}$</td>
<td>19000 lb</td>
<td>1000 lb</td>
<td>+10000 lb/s</td>
<td>-10000 lb/s</td>
<td>1 s</td>
</tr>
<tr>
<td>Elevator, $\delta_{e}$</td>
<td>+25 deg</td>
<td>-25 deg</td>
<td>+60 deg/s</td>
<td>-60 deg/s</td>
<td>0.0495 s</td>
</tr>
</tbody>
</table>

The augmented linear models in equation (8) of the inner-loop had 17 states
and were discretized at 20 Hz. Five states were associated with the aircraft dynamics, an additional five states represented the $H_\infty$ inner-loop controller. Engine and elevator actuators together contributed two more states. The five remaining states were introduced by the augmentation of the inner-loop state-space description with integrators in equation (8). Three of these extra states were associated with the simple disturbance predictor and two integrators were used to convert control changes $\Delta r$ into actual controls $r$. 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 $\{0, 0.5, 1.0, 1.5\}$ seconds ($H_c = 31, \delta_{H_c} = 10$). 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 MPC scheme in Maciejowski (2002) and had constant values of $Q = diag\{0.4, 10\}$ and $R = diag\{0.1, 1\}$.

Three simulation scenarios are presented to illustrate the benefits and drawbacks of the RHC schemes under investigation.

(1) 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 nonzero trim states and controls at this flight condition are illustrated in Table 2. The single 

<table>
<thead>
<tr>
<th>Variables</th>
<th>Trim values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>10000 ft</td>
</tr>
<tr>
<td>$V_t$</td>
<td>579.12 ft/s</td>
</tr>
<tr>
<td>$\alpha = \theta$</td>
<td>2.315 deg</td>
</tr>
<tr>
<td>$\delta_{th}$</td>
<td>2247.7 lb</td>
</tr>
<tr>
<td>$\delta_c$</td>
<td>-1.945 deg</td>
</tr>
</tbody>
</table>

LTI model based linear RHC scheme uses a prediction model that corresponds to a different flight condition of ($h = 6000$ ft, $V_t = 800$ ft/s) to better illustrate the inherent problems with this approach.

(2) 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 maneuver in this example requires an altitude change of approximately 20000 ft and a reduction of the true airspeed by 300 ft/s.

(3) The third example aims at pointing out the aggressive maneuvering ca-
pabilities enabled by the scheduled RHC approach, as well as system-
state dependent constraint enforcement represented by vertical accelera-
tion limits that vary with true airspeed. This simulation demonstrates
two scenarios. First, a relatively aggressive reference altitude and velocity
trajectory is flown without any maneuvering constraints on vertical ac-
celeration. 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. The trim conditions are the same as
in Example 1.

In all of these examples, the general goal of the outer-loop RHC controllers is
to accomplish “higher level” control objectives, by exploiting \textit{a priori} refer-
ence 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 within Matlab on a 1.2 GHz Pentium III ma-
chine running RedHat Linux. Using the specific parameters described in Sec-
tion 4 to formulate the optimization problem, the resulting QP had 8 decision
variables (9 with the slack variable in example 3). The number of linear con-
straints were 664 (826 with the soft constraints).

5.1 Example 1

The results of the disturbance rejection scenario are shown in Fig. 4 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 (Bhattacharya et al., 2002), 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 designed for a different flight condition leads to steady state error, whereas the scheduled approach provides acceptable performance even in comparison with the nonlinear technique. It should be noted that if the single LTI pre-
diction model is chosen according to the actual flight condition where the disturbance rejection scenario takes place, then the performance of the single LTI model based RHC scheme is almost identical to the results obtained with the scheduled RHC approach. However, in general it is desirable to have similar disturbance rejection properties throughout the flight envelope, which is not achieved using a single LTI model for prediction, as illustrated by this example.

Computational requirements of the different approaches are reported in Table 3, by summarizing the average and maximum execution times of the various controllers at each time step in the entire simulation. The overall execution time is further subdivided into the effort spent on solving the optimal control problems and into other computations required for setting up the problem and interpolating over prediction models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Optimization average</th>
<th>Optimization maximum</th>
<th>Other calculations average</th>
<th>Other calculations maximum</th>
<th>Total execution average</th>
<th>Total execution maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear RHC</td>
<td>0.022 s</td>
<td>0.045 s</td>
<td>0.01 s</td>
<td>0.011 s</td>
<td>0.032 s</td>
<td>0.056 s</td>
</tr>
<tr>
<td>Scheduled RHC</td>
<td>0.022 s</td>
<td>0.045 s</td>
<td>1.13 s</td>
<td>1.49 s</td>
<td>1.152 s</td>
<td>1.511 s</td>
</tr>
<tr>
<td>Nonlinear RHC</td>
<td>30.97 s</td>
<td>102.45 s</td>
<td>0.1 s</td>
<td>0.16 s</td>
<td>31.08 s</td>
<td>102.55 s</td>
</tr>
</tbody>
</table>

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 0.01 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 0.04 second on average. The numbers in Table 3 indicate that this approach is readily implementable in real time, even using the Matlab environment.

The optimization problem complexity is exactly the same in the scheduled RHC scheme as in the linear one, i.e. the QP can be solved in approximately real-time, even if constraints are active. However, Table 3 shows that 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) performed 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 as in Schram et al. (1997) or a number of other options), real-time implementation
of this scheme is also deemed achievable. In fact, compared to execution times in Matlab we observed an average speed-up factor of 7.89 when implementing the scheduled RHC approach in C/C++ for flight testing of a different aircraft control problem reported in Keviczky et al. (2004).

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 31 seconds for each 0.05 second sample interval. Although the main emphasis in this study was not efficient implementation of algorithms but their comparability, 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 Fig. 5, the linear and scheduled 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 scheduled approach maintains the same performance throughout the entire maneuver. 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, achieves lower cost by trading off altitude errors for better velocity tracking and uses larger control values.

5.3 Example 3

The third example shown in Fig. 6, was performed only using the scheduled 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 $\infty$-norm (maximum violation) penalty on constraint violations (as shown in (6) and (13)) and
Fig. 5. Simulation results of example 2 (reference: dotted, linear RHC: dash-dot, scheduled RHC: solid, nonlinear RHC: dash).

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 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,
Fig. 6. Simulation results of example scenario 3 (reference: dotted, scheduled RHC w/o constraints: dash, scheduled RHC with soft constraints: solid).

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. In fact, a receding horizon guidance controller based on the scheduled scheme and using online linearized prediction models was successfully flight tested in June 2004 as part of the DARPA SEC program final demonstration (Keviczky et al., 2004; Keviczky and Balas, 2005). The proposed scheduled 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 the Defense Advanced Research Projects Agency under the Software Enabled Control program, Dr. John Bay Program Manager. The contract number is USAF/AFMC F33615-99-C-1497, Dale Van Cleave was the Technical Contract Monitor.

References


URL http://www.aem.umn.edu/people/faculty/balas/darpa_sec/SEC.Publications.html


URL http://www.aem.umn.edu/people/faculty/balas/darpa_sec/SEC.Software.html
