Constraint Reduction with Exact Penalization for Model-Predictive Rotorcraft Control*

Meiyun Y. He,† André L. Tits,†
Aaron L. Greenfield,‡ and Vineet Sahasrabudhe‡

February 1, 2012

Abstract

Model Predictive Control (also known as Receding Horizon Control (RHC)) has been highly successful in process control applications. Its use for aerospace applications has been hindered by its high computational requirements. In the present paper, we propose using enhanced primal-dual interior-point optimization techniques in the convex-quadratic-program-based RHC control of a rotorcraft. Our enhancements include a previously proposed "constraint-reduction" scheme that takes advantage of the very large number of inequality constraints (compared to the number of variables), and the ensuing redundancy of a large majority of these constraints. Other enhancements include the use of a penalty function, with automatic adaptation of the penalty parameter (also previously analyzed in the context of constraint-reduction), allowing for the use of "infeasible" "warm starts", and of the partition of the constraints into "hard" to be imperatively satisfied, and "soft" whose violations can possibly be traded-off, with appropriate supporting algorithm. The heart of the paper is the application of all these techniques to an aggressive-trajectory-following problem for a model of a utility-class helicopter. The results are encouraging, and demonstrate that RHC control of rotorcraft should soon become a mature technology.

Key Words— model predictive control, MPC, receding horizon control, RHC, constraint reduction, exact penalty function, warm start, quadratic programming, interior-point methods, affine scaling.

*This work was supported in part by a grant from Sikorsky Aircraft Corporation to the University of Maryland, and by the US Department of Energy under Grant DESC0002218. The background material in the present paper is largely borrowed from [HKT+10].

†M.Y. He and A.L. Tits are with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA myhe@umd.edu, andre@umd.edu

‡A.L. Greenfield and V. Sahasrabudhe are with Sikorsky Aircraft Corporation, Stratford, CT 06614, USA agreenfield@sikorsky.com, vSahasrabudhe@sikorsky.com

1 Introduction

Model Predictive Control (also known as Receding Horizon Control (RHC))[1] has been highly successful in process control applications; see, e.g., [CB04, KLS01, QB03]. Its use for aerospace applications has been hindered by its high computational requirements: at each time step, determination of the control value to be applied at the next time step involves the solution of an optimization problem. Still, RHC-based control for spacecraft formation keeping and attitude control is studied in [MAG+99]; in [WBB01], RHC is used in conjunction with a neural network feedback controller, to control a six-degree-of-freedom model of an autonomous helicopter; and in [KB06] RHC is successfully applied in flight tests on unmanned aerial vehicles. In [HKT+10] (a stepping stone to the present paper), control of a helicopter in hover is considered.

In the present paper, our focus is on enhancing the efficiency of the optimization solver to be used at each time step, specifically a convex quadratic programming (CQP) solver. Like [HKT+10], the present paper features a "constraint-reduction" technique for primal dual interior-point (PDIP) optimization, along the lines of recent work by the first two authors and their co-authors: [TAW06, WNTO11, HT11] for linear programming and [JOT12, HT12] for convex quadratic programming. Constraint reduction significantly reduces the computational cost per iteration of the optimization process by exploiting the presence of a large number of inequality constraints (much larger than the number of decision variables) and the resulting redundancy of a large majority of these constraints. In [JOT12] and [HT11, HT12], constraint reduction is complemented by an exact penalty scheme that allows for infeasible initial points, in particular infeasible "warm starts", a common occurrence in the RHC context, and in [HT11, HT12] an adaptive mechanism is proposed and analyzed for determining an appropriate penalty parameter value. As demonstrated in the present paper, this property proves remarkably valuable in the context of RHC for rotorcraft control. An additional, specific contribution of the present paper is a scheme to handle "hard" and "soft constraints": hard constraints must be satisfied by the initial condition (at each time step), and a technique is introduced that guarantees that they will be met throughout; soft constraint are allowed to be initially violated, and the algorithm tends to reduce that violation, and if doable generate a final iteration that satisfies all of them. The heart of the paper is the application of all these techniques to an aggressive-trajectory-following problem for a model of a utility-class helicopter. This problem is significantly more challenging than the hover control considered in [HKT+10]. Here constraints on the values of the control inputs and its derivatives are defined to be hard constraints, while all others are made soft.

The paper is organized as follows. Section 2 sets up the optimization problem for RHC-based trajectory following. In section 3, an optimization methodology is proposed that uses constraint reduction and an exact penalty function. Section 4 reports simulation results for a 38-state rotorcraft. We conclude in Section 5.

[1] We will mostly refrain from using acronym MPC, ubiquitous in the model predictive control literature. This is because, by an unfortunate coincidence, that same acronym is commonly used with a different meaning—Mehrotra's Predictor Corrector algorithm, also briefly considered in the present paper—in the interior-point optimization literature. Indeed the present paper is targeted at both audiences.

2 Notation and Problem Set Up

We consider an RHC controller which, during each time interval $(t-1,t)$, t an integer, solves the convex quadratic program (CQP)

$$\min \sum_{k=0}^{M-1} u_k^T R u_k + \sum_{k=1}^{N} (x_k - x_k^{\mathrm{r}})^T Q (x_k - x_k^{\mathrm{r}}) \tag{1}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1, \tag{2}$$

$$x_0 = Ax(t-1) + Bu(t-1), \tag{3}$$

$$u_k = u_{M-1},\ k = M, \ldots, N-1, \tag{4}$$

$$u_{\min} \le u_k \le u_{\max},\ k = 0, \ldots, M-1, \tag{5}$$

$$\delta u_{\min} \le u_k - u_{k-1} \le \delta u_{\max},\ k = 1, \ldots, M-1, \tag{6}$$

$$\delta u_{\min} \le u_0 - u(t-1) \le \delta u_{\max}, \tag{7}$$

$$x_{\min} \le x_k \le x_{\max},\ k = 1, \ldots, N, \tag{8}$$

$$\delta x_{\min} \le x_k - x_{k-1} \le \delta x_{\max},\ k = 1, \cdots, N, \tag{9}$$

where minimization is with respect to $x_1, \cdots, x_N$, all in $\mathbb{R}^n$ and $u_0, \cdots, u_{M-1}$, all in $\mathbb{R}^m$; $R \in \mathbb{R}^{m\times m}$ is positive definite and $Q \in \mathbb{R}^{n\times n}$ positive semidefinite; the (vector) bounds $u_{\min}, u_{\max}, \delta u_{\min}, \delta u_{\max}, x_{\min}, x_{\max}, \delta x_{\min}, \delta x_{\max}$ are given, with components in $\mathbb{R} \cup \{\pm\infty\}$. Vector x_k is an estimate of the state of the rotorcraft at time $t+k$, based on model (2)-(3), given the "true" (measured) state $x(t-1)$ at time $t-1$, under the control sequence $\{u(t-1), u_0, u_1, \cdots, u_{k-1}\}$, where $u(t-1)$ is the control value that was applied at time $t-1$. The "prediction horizon" N is typically larger than the "control horizon" M. Constraints (6)–(7) and (9) restrict the rate of change of the control inputs and that of the states, respectively. Path x^{r} is a specified reference trajectory. Following the Matlab Model-Predictive-Control Toolbox [MR94], control values beyond the control horizon are constrained to be the value at the control horizon (see (4)).

We adopt the Matlab-inspired notation $[v_1; v_2; \ldots; v_p]$ to denote a vertical concatenation of vectors or matrices v_i, $1 \le i \le p$. Let vectors $\mathbf{u}$ and $\mathbf{x}$ be defined by

$$\mathbf{u} := [u_0; \ldots; u_{M-1}] \in \mathbb{R}^{Mm}, \quad \mathbf{x} := [x_1; \ldots; x_N] \in \mathbb{R}^{Nn}.$$

and let

$$\Gamma := \begin{bmatrix} B & 0 & \cdots & 0 & 0 \\ AB & B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{M-1}B & A^{M-2}B & \cdots & AB & B \\ A^M B & A^{M-1}B & \cdots & A^2 B & AB+B \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & A^{N-M+1}B & \sum_{i=0}^{N-M} A^i B \end{bmatrix}$$

and $\Omega := \begin{bmatrix} A; A^2; \ldots; A^N \end{bmatrix}$. Then, constraints (2)–(4) becomes

$$\mathbf{x} = \Gamma \mathbf{u} + \Omega(Ax(t-1) + Bu(t-1)). \tag{10}$$

Further, define matrices

$$\mathbf{Q} := \mathrm{diag}(Q, \ldots, Q) \in \mathbb{R}^{Nn \times Nn}, \qquad \mathbf{R} := \mathrm{diag}(R, \ldots, R) \in \mathbb{R}^{Mm \times Mm},$$

$$E := \begin{bmatrix} I_m & 0 & \cdots & 0 \\ -I_m & I_m & \cdots & 0 \\ & \ddots & \ddots & \\ 0 & \cdots & -I_m & I_m \end{bmatrix} \in \mathbb{R}^{Mm \times Mm}, \qquad F := \begin{bmatrix} I_n & 0 & \cdots & 0 \\ -I_n & I_n & \cdots & 0 \\ & \ddots & \ddots & \\ 0 & \cdots & -I_n & I_n \end{bmatrix} \in \mathbb{R}^{Nn \times Nn},$$

$$E_1 := [I_m; 0; \ldots; 0] \in \mathbb{R}^{m \times Mm}, \qquad F_1 := [I_n; 0; \ldots; 0] \in \mathbb{R}^{n \times Nn},$$

and vectors

$$\begin{aligned} \mathbf{x}_{\max} &:= [x_{\max}; \ldots; x_{\max}] \in \mathbb{R}^{Nn}, & \mathbf{x}_{\min} &:= [x_{\min}; \ldots; x_{\min}] \in \mathbb{R}^{Nn}, \\ \delta\mathbf{x}_{\max} &:= [\delta x_{\max}; \ldots; \delta x_{\max}] \in \mathbb{R}^{Nn}, & \delta\mathbf{x}_{\min} &:= [\delta x_{\min}; \ldots; \delta x_{\min}] \in \mathbb{R}^{Nn}, \\ \mathbf{u}_{\max} &:= [u_{\max}; \ldots; u_{\max}] \in \mathbb{R}^{Mm}, & \mathbf{u}_{\min} &:= [u_{\min}; \ldots; u_{\min}] \in \mathbb{R}^{Mm}, \\ \delta\mathbf{u}_{\max} &:= [\delta u_{\max}; \ldots; \delta u_{\max}] \in \mathbb{R}^{Mm}, & \delta\mathbf{u}_{\min} &:= [\delta u_{\min}; \ldots; \delta u_{\min}] \in \mathbb{R}^{Mm}, \\ \mathbf{x}^{\mathrm{r}} &:= [x_1^{\mathrm{r}}; \cdots, x_N^{\mathrm{r}}] \in \mathbb{R}^{Nn}. \end{aligned}$$

Substituting (10) into (1), (8) and (9) then yields a CQP problem in inequality form,

$$\min_{\mathbf{u}} \mathbf{f}^T\mathbf{u} + \frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u} \quad \text{s.t.} \quad \mathbf{A}\mathbf{u} \geq \mathbf{b} \tag{11}$$

where

$$\mathbf{f} := \Gamma^T\mathbf{Q}\left(\Omega(Ax(t-1) + Bu(t-1)) - \mathbf{x}^{\mathrm{r}}\right), \quad \mathbf{H} := \mathbf{R} + \Gamma^T\mathbf{Q}\Gamma$$

and

$$\mathbf{A} := \begin{bmatrix} -I_{Mm} \\ I_{Mm} \\ -\Gamma \\ \Gamma \\ -E \\ E \\ -F\Gamma \\ F\Gamma \end{bmatrix} \in \mathbb{R}^{4(Mm+Nn) \times Mm}, \quad \mathbf{b} := \begin{bmatrix} -\mathbf{u}_{\max} \\ \mathbf{u}_{\min} \\ -\mathbf{x}_{\max} + \Omega(Ax(t-1) + Bu(t-1)) \\ \mathbf{x}_{\min} - \Omega(Ax(t-1) + Bu(t-1)) \\ -(\delta\mathbf{u}_{\max} + E_1u(t-1)) \\ \delta\mathbf{u}_{\min} + E_1u(t-1) \\ -\delta\mathbf{x}_{\max} + (F\Omega - F_1)(Ax(t-1) + Bu(t-1)) \\ \delta\mathbf{x}_{\min} - (F\Omega - F_1)(Ax(t-1) + Bu(t-1)) \end{bmatrix}. \tag{12}$$

Because $\mathbf{H}$ is positive definite, problem (11) has a unique solution $\mathbf{u}^* = [u_0^*; \ldots; u_{M-1}^*]$ whenever it is feasible. In the spirit of model predictive control, control input $u(t) = u_0^*$ is applied to the rotorcraft at sample time t; the other sub-vectors of $\mathbf{u}^*$ are discarded but used as a "warm-start" towards solving the next CQP (see section 4.2 below).

The number of variables in problem (11) is Mm and, when all the state and control input variables are constrained (all components of $u_{\min}, u_{\max}, \delta u_{\min}, \delta u_{\max}, x_{\min}, x_{\max}, \delta x_{\min}, \delta x_{\max}$ are finite), the total number of constraints is $4Mm + 4Nn$, which is much larger than Mm—recall that $N > M$ and $n > m$. Even though, in our numerical example (see section 4), many state variables are unconstrained, the number of constraints is still significantly larger than that of variables, making constraint reduction, discussed below, beneficial.

3 An Optimization Methodology

3.1 Constraint reduction

Constraint-reduced PDIP was first considered in [TAW06] in the context of linear programming, where it was later refined in [WNTO11]. It was extended to CQP in [JOT12]. The algorithm proposed in [JOT12], when applied to problem (11), uses search directions associated to the reduced problem

$$\min_{\mathbf{u}} \mathbf{f}^T\mathbf{u} + \frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u} \quad \text{s.t.} \quad \mathbf{A}_{\mathcal{Q}}\mathbf{u} \geq \mathbf{b}_{\mathcal{Q}} \tag{13}$$

where $\mathbf{A}_{\mathcal{Q}}$ and $\mathbf{b}_{\mathcal{Q}}$ respectively denotes the submatrix of $\mathbf{A}$ and sub-vector of $\mathbf{b}$ with only those rows that are indexed in a certain working set $\mathcal{Q} \subseteq \{1, \cdots, 4(Mm+Nn)\}$. In (13), $\mathbf{u}$ must be strictly feasible, i.e., $\mathbf{s} := \mathbf{A}\mathbf{u} - \mathbf{b} > 0$, and $\mathcal{Q}$, which is updated at each iteration, is selected to include all the most nearly active constraints, usually at least Mm (dimension of $\mathbf{u}$) of them, while insuring that $[\mathbf{H}; \mathbf{A}_{\mathcal{Q}}]$ has full column rank. In the "affine-scaling" version analyzed in [JOT12], a "primal-dual" search direction $(\Delta\mathbf{u}, \Delta\mathbf{s}_{\mathcal{Q}}, \Delta\boldsymbol{\lambda}_{\mathcal{Q}})$ is computed by solving the system

$$\begin{aligned} \mathbf{H}\Delta\mathbf{u} - \mathbf{A}_{\mathcal{Q}}^T\Delta\boldsymbol{\lambda}_{\mathcal{Q}} &= -(\mathbf{H}\mathbf{u} + \mathbf{f} - \mathbf{A}_{\mathcal{Q}}^T\boldsymbol{\lambda}_{\mathcal{Q}}), \\ \mathbf{A}_{\mathcal{Q}}\Delta\mathbf{u} - \Delta\mathbf{s}_{\mathcal{Q}} &= \mathbf{0}, \qquad (14) \\ \mathbf{S}_{\mathcal{Q}}\Delta\boldsymbol{\lambda}_{\mathcal{Q}} + \boldsymbol{\Lambda}_{\mathcal{Q}}\Delta\mathbf{s}_{\mathcal{Q}} &= -\mathbf{S}_{\mathcal{Q}}\boldsymbol{\lambda}_{\mathcal{Q}}, \qquad (15) \end{aligned}$$

where $\mathbf{s}_{\mathcal{Q}}$ and $\boldsymbol{\lambda}_{\mathcal{Q}}$ are obtained from the slack vector $\mathbf{s}$ and Karush-Kuhn-Tucker (KKT) multiplier estimate (or "dual variable") $\boldsymbol{\lambda} > 0$ by keeping only the rows indexed in $\mathcal{Q}$, and where $\mathbf{S}_{\mathcal{Q}} := \mathrm{diag}(\mathbf{s}_{\mathcal{Q}})$ and $\boldsymbol{\Lambda}_{\mathcal{Q}} := \mathrm{diag}(\boldsymbol{\lambda}_{\mathcal{Q}})$. Eliminating $\Delta\mathbf{s}_{\mathcal{Q}}$ and $\Delta\boldsymbol{\lambda}_{\mathcal{Q}}$ yields

$$\mathcal{M}_{(\mathcal{Q})}\Delta\mathbf{u} = -(\mathbf{H}\mathbf{u} + \mathbf{f}) \tag{16}$$

where

$$\mathcal{M}_{(\mathcal{Q})} := \mathbf{H} + \mathbf{A}_{\mathcal{Q}}^T\mathbf{S}_{\mathcal{Q}}^{-1}\boldsymbol{\Lambda}_{\mathcal{Q}}\mathbf{A}_{\mathcal{Q}}.$$

The dominant cost lies in forming matrix $\mathcal{M}_{(\mathcal{Q})}$, taking approximately $(Mm)^2|Q|$ flops—solving (16) via Cholesky decompostion then only takes $(Mm)^3/3$ flops. Compared to the original cost $(Mm)^2(4Mm+4Nn)$ (corresponding to $|Q| = 4Mm+4Nn$), constraint reduction affords a speedup in the computation of $\Delta\mathbf{u}$ of close to $(4Mm+4Nn)/|Q|$.

With $\Delta\mathbf{u}$ in hand, $(\Delta\mathbf{s}_{\mathcal{Q}}, \Delta\boldsymbol{\lambda}_{\mathcal{Q}})$ are obtained by solving (at low cost)

$$\Delta\mathbf{s} = \mathbf{A}\Delta\mathbf{u}, \quad \mathbf{S}\Delta\boldsymbol{\lambda} + \boldsymbol{\Lambda}\Delta\mathbf{s} = -\mathbf{S}\boldsymbol{\lambda},$$

where $\mathbf{S} := \mathrm{diag}(\mathbf{s})$. Appropriate step sizes are then taken along $\Delta\mathbf{u}$ and $\Delta\boldsymbol{\lambda}$. See [JOT12] for full details.

3.2 Enhancements

The constrained-reduced PDIP algorithms of [TAW06, WNTO11] require that a strictly feasible initial point be available (and so do earlier constraint-reduced dual interior-point methods, such as the one considered in [Ye90, DY91, dHRT92]). Such point may not be available in an RHC context due in particular to disturbances and modeling errors. This difficulty is partially solved in [JOT12] (section 3) and [HKT+10] by introducing an ℓ_1/ℓ_∞ penalty function; specifically, in the ℓ_∞ case: problem (11) is relaxed to

$$\begin{array}{ll} \min\limits_{\mathbf{u},\mathbf{z}} & \mathbf{f}^T\mathbf{u} + \rho z + \frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u} \\ \text{s.t.} & \mathbf{A}\mathbf{u} + z\mathbf{e} \geq \mathbf{b},\ z \geq 0, \end{array} \tag{17}$$

where $\rho > 0$ is the penalty parameter and $\mathbf{e}$ is the vector of all ones. Given any desired warm start $\mathbf{u}$, a strictly feasible point $[\mathbf{u}; z]$ for (17) is trivially available by setting z to large enough values. Because problem (17) has only one more variable and constraint than (13), the computational cost per iteration is not increased.

It is not guaranteed that the solution $[\mathbf{u}^*; z^*]$ to (17) will be such that $\mathbf{u}^*$ is feasible for (11) (and $z^* = 0$). Still, our ℓ_∞ penalty function is "exact", i.e., if (11) is feasible, then $\mathbf{u}^*$ will be feasible indeed (and will solve (11)) provided ρ is larger than some threshold value ρ_* (e.g., Theorem 40, [FM90]). However, ρ_* is not known *a priori*. In [JOT12], the penalty parameter must be provided by the user, and in [HKT+10], an appropriate value of ρ was obtained by trial-and-error. Both are impractical in the context of model predictive control where a large number of successive problems are run in real time. An automatic, adaptive penalty adjustment scheme was proposed in [HT11] for linear programs and extended to the case of CQP in Algorithm IrQP of [HT12] which was used in the tests reported in this paper. Specifically, in [HT12], at the end of each iteration, ρ is increased by a constant factor, i.e.,

$$\rho := \sigma\rho \tag{18}$$

with $\sigma > 1$, when either

$$z \geq \gamma_1\rho \tag{19}$$

or

$$\text{(i) } \|[\Delta\mathbf{u}; \Delta z]\| \leq \gamma_2, \text{ AND (ii) } \boldsymbol{\lambda}_\mathcal{Q} + \Delta\boldsymbol{\lambda}_\mathcal{Q} \geq -\gamma_3\mathbf{e}, \text{ AND (iii) } y + \Delta y < \gamma_4 \tag{20}$$

is satisfied, where y is the dual variable associated to constraint $z \geq 0$, and γ_i, $i = 1, 2, 3, 4$ are positive parameters. Condition (19) prevents z from growing without bound. Condition (20) triggers an increase of ρ when the current iterate is close to the solution of (17) but z is not close to zero.

Problem (17) *always* has an optimal (feasible) solution. When (11) is infeasible, the optimal z for (17) has some nonzero components. In that sense, the original constraints are made "soft" and, by minimizing z, (17) trades off the constraint violation against each other.

To force satisfaction of certain "hard" constraints, the following intermediate formulation can be used

$$\min_{\mathbf{u},\mathbf{z}} \quad \mathbf{f}^T\mathbf{u} + \rho z + \frac{1}{2}\mathbf{u}^T\mathbf{H}\mathbf{u} \tag{21}$$

$$\text{s.t.} \quad \mathbf{A}_S\mathbf{u} + z\mathbf{e} \geq \mathbf{b}_S,\ z \geq 0, \tag{22}$$

$$\mathbf{A}_{\bar{S}}\mathbf{u} \geq \mathbf{b}_{\bar{S}}, \tag{23}$$

where S is the index set of soft constraints, and its complement $\bar{S}$ that of hard constraints. (Of course, a warm start $\mathbf{u}$ that violates (23) must then be prohibited. In the context of RHC, it is natural to make constraints (5) and (6) hard.)

IrQP normally terminates when

$$\max\left\{\frac{\|[\mathbf{H}\mathbf{u} - \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{f}; \mathbf{e}^T\boldsymbol{\lambda} + y - \rho]\|_\infty}{\max\{\|\mathbf{H}\|_\infty, 1 + \|\mathbf{A}\|_\infty, \|\mathbf{f}\|_\infty, \rho\}}, \frac{\mathbf{s}^T\boldsymbol{\lambda} + z^T y}{4(Mm + Nn) + 1}\right\} \leq tol, \tag{24}$$

where *tol* is a small positive value selected by the user.

In [HT12] (following [TAW06, WNTO11, JOT12]), global convergence is proved under general guidelines on how to select $\mathcal{Q}$; much freedom is left to be exploited based on the application. In the present implementation, during the first few (10 in our experiments) iterations in the optimization process of current problem, we forced the set $\mathcal{Q}$ to always contain (as a subset) the active set at the solution of *previous* problem. Indeed, the active set at the solution of two successive problems usually do not change much.

It is proved in [HT12] that, under mild assumptions (in particular, strict feasibility of (11)), that the sequence $\{\mathbf{u}_k\}$ generated by Algorithm IrQP converges to an optimal solution of (11). When (11) is infeasible (which commonly occurs in an RHC context), ρ increases without bound. In such case, the sequence generated by IrQP tends to minimize the constraint violations. Very large values of ρ tend to significantly slow down the optimization process. For that reason, we modified IrQP by placing an upper bound on the number of times ρ can be increased.

4 Application to Rotorcraft Control

4.1 Rotorcraft models

We started from model M2,[2] a linear time-invariant, continuous-time model of a utility-class helicopter, of the form

$$\dot{x} = A^{\text{ct}}x + B^{\text{ct}}u. \tag{25}$$

This model was provided to us by R. Celi, who generated it as described in [TC02] for the case of constant forward velocity $V = 80$ kts. It has 31 states and 4 control inputs; see Table 1 below, as well as Tables 3 and 4 in the appendix. For the purpose of the present

[2]The name reflects the fact that the model accounts for two flexible-blade modes (rigid body flap and lag).

paper, model M2 was assumed to be exact and appropriate components $x(\cdot)$ of its states were substituted in (3) for the measured state of the rotorcraft.

Table 1: Description of the original model M2 from [TC02]. States and control inputs estimated/generated by the controller are marked in boldface.

States 1–3 (u, v, w)	**Longitudinal (positive forward), lateral (positive right), and vertical (positive down) velocity coordinates**
States 4–6 (p, q, r)	**Roll, pitch and yaw rates**
States 7–9 (ϕ, θ, ψ)	**Roll, pitch and yaw angles**
States 10–17	Blade rigid body flap
States 18–25	Blade lag
States 26, 27	Torsion, torsion rate
State 28	Constant portion (offset) of main rotor inflow
States 29, 30	First harmonic (cos and sin) of main rotor inflow
State 31	Tail rotor inflow
Control inputs	**Lateral cyclic, longitudinal cyclic**, collective and **pedal**.

To keep the RHC controller reasonably simple and fast, we decided to only simulate the first nine states (the "classic nine") in the built-in simulator (2)-(3). The reduction from 31 to nine states was effected as follows. Denote by $A^{\rm ct}_{\alpha,\beta}$ the submatrix of $A^{\rm ct}$ that consists of only those rows in set α and only those columns in set β, and denote by $B^{\rm ct}_{\alpha}$ the submatrix of $B^{\rm ct}$ that consists of only those rows in set α (and all columns). Let r consist of the kept states in the reduced model, and d of the deleted states. By setting the derivatives of the deleted states to zero, system (25) becomes

$$\dot{x}_{\rm r} = A^{\rm ct}_{\rm r,r} x_{\rm r} + A^{\rm ct}_{\rm r,d} x_{\rm d} + B^{\rm ct}_{\rm r} u, \tag{26}$$

$$0 = \dot{x}_{\rm d} = A^{\rm ct}_{\rm d,r} x_{\rm r} + A^{\rm ct}_{\rm d,d} x_{\rm d} + B^{\rm ct}_{\rm d} u. \tag{27}$$

Noting that $A^{\rm ct}_{\rm d,d}$ in M2 is non-singular, we can solve (27) for $x_{\rm d}$ and substitute into (26), yielding the reduced model

$$\dot{x}_{\rm r} = \overline{A^{\rm ct}} x_{\rm r} + \overline{B^{\rm ct}} u \tag{28}$$

where

$$\overline{A^{\rm ct}} = A^{\rm ct}_{\rm r,r} - A^{\rm ct}_{\rm r,d}(A^{\rm ct}_{\rm d,d})^{-1} A^{\rm ct}_{\rm d,r}, \quad \overline{B^{\rm ct}} = B^{\rm ct}_{\rm r} - A^{\rm ct}_{\rm r,d}(A^{\rm ct}_{\rm d,d})^{-1} B^{\rm ct}_{\rm d}.$$

Our objective is to track a reference trajectory consisting of a nose-down followed by an aggressive nose-up maneuver (see Figure 4), subject to actuator and other limitations. One such limitation is a bound on the load factor N_z associated to pitch. N_z satisfies the differentiable equation

$$\dot{N}_z = \frac{V}{g} Z_w q - Z_w N_z. \tag{29}$$

Here, g is the acceleration of gravity, V is the forward speed of the helicopter and Z_w is the negative of the $(3,3)$ entry of $\overline{A^{\rm ct}}$ (associated with state w) in the reduced model. According to the model M2 data, $V = 80$ kts $= 134.96$ ft/sec and $Z_w = 0.6920$ sec^{-1}; and we used $g := 32.174$ ft/sec^2.

To be able to place a constraint on N_z, we augmented both (25) and (28) with (29), yielding a 32-state model considered below as an exact representation of the rotorcraft, and a 10-state model to be used by the controller. Both models have 4 control inputs. Discrete-time models were then generated from both (using Matlab command `c2d`). The first one was fed the sequence $u(\cdot)$ generated by the controller, and appropriate components $x(\cdot)$ of its state were fed back (see (12)). The second one, shown in Table 5 in the appendix, was used in controller's estimator (2)–(3).

4.2 Optimization details

The desired pitch trajectory was set as shown by the solid lines on Figure 3 (pitch rate) and Figure 4 (pitch attitude); all other components of the desired trajectories were set to zero. Because use of the collective is not essential for tracking such trajectory, and for sake of controller simplicity, we decided to set it to zero throughout, so that each u_k effectively has three components rather than four. Values $R = \mathrm{diag}(1,1,1)$ and $Q = \mathrm{diag}(0,0,0,10^3,10^4,10,10,10^4,10,0)$ were found to be appropriate for the cost function (no tracking is attempted for u, v, w and N_z). Constraint bounds were set to $u_{\max} = -u_{\min} = [5;5;5]$ inches, $\delta u_{\max} = -\delta u_{\min} = [4;4;4]$ inches/sec, $\phi_{\max} = -\phi_{\min} = 5$ degrees, $\psi_{\max} = -\psi_{\min} = 4$ degrees, $N_z^{\max} = -N_z^{\min} = 1$ g, $\delta p_{\max} = -\delta p_{\min} = 1$ rad/sec^2 and $\delta q_{\max} = -\delta q_{\min} = 1$ rad/sec^2. Bound constraints on all control inputs and their rates of change, i.e., (5)–(7), were set to be "hard" while all other constraints were made "soft".

With three control inputs ($m = 3$), our CQP problem (11) has $3M$ variables. Only three out of ten state variables have finite prescribed upper and lower bounds, which gives us $6N$ constraints from (8). Also, two out of 10 state variables have finite upper and lower bounds on their rates of change, giving another $4N$ constraints from (9). In addition, all $4 \times 3M$ bounds on control inputs and their rates of change are finite. Hence, the total number of constraints in problem (11) is $12M + 10N$. With our current choice of $M = 30$ and $N = 100$, we thus have 90 optimization variables and 1360 constraints.

For the first CQP (where $t = 0$), the initial values $\mathbf{u}^0$ of $\mathbf{u}$ was set to the zero vector, as were $x(t-1)$ and $u(t-1)$ (used in (12)). For all other CQPs, the warm start $\mathbf{u}^0$ was set to $0.998 \times \mathbf{u}^*$ where $\mathbf{u}^*$ is the solution of previous problem (thus enforcing *strict* satisfaction of the hard constraints), and we used the warm start $z^0 := \|\max\{\mathbf{0}, \mathbf{b}_S - \mathbf{A}_S\mathbf{u}^0\}\|_\infty + 0.001$ for variable z, where the "max" is taken component-wise. The dual variables $[\boldsymbol{\lambda}; y]$ were set to all ones for the first CQP, and to $[\boldsymbol{\lambda}^*; y^*]$ for all others, where $[\boldsymbol{\lambda}^*; y^*]$ is the KKT multiplier vectors associated to the solution of CQP (17) at previous time step. The initial value of the penalty parameter was set to 2×10^7 for the first CQP, and for all others, it was set to $\min\{2 \times 10^7, 10\|\boldsymbol{\lambda}^*\|_\infty\}$. Parameters in (18)–(20) were set to $\sigma = 10$, $\gamma_1 = 0.1$, $\gamma_2 = 100$ and $\gamma_4 = 1$. The penalty parameter was allowed to increase at most five times—see our comment at the end of section 3. The optimization runs were terminated when (24) was satisfied with

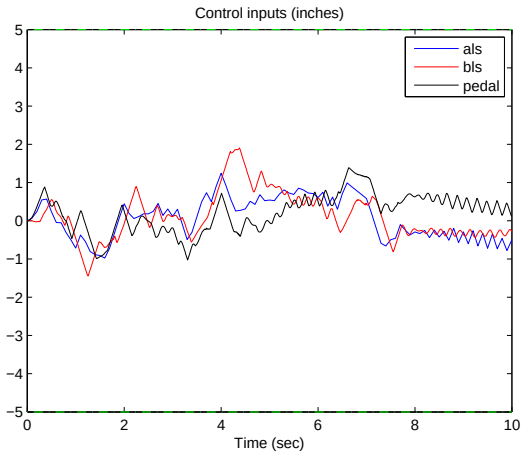

Figure 1: Control inputs: lateral cyclic, longitudinal cyclic and pedal, marked by blue circles, and red and magenta lines, respectively.

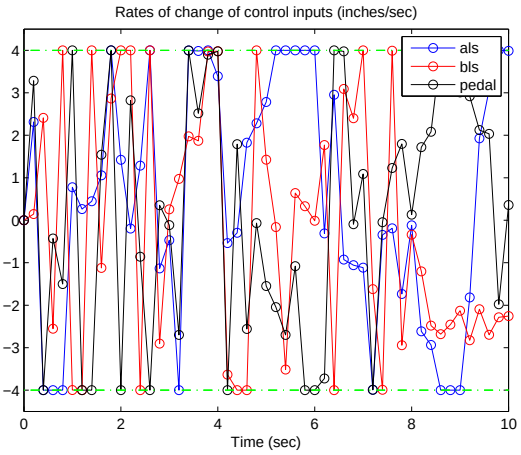

Figure 2: Rates of change of control inputs. The upper bound of 4 inches/sec and lower bound of -4 inches/sec are marked by dashed green lines. To reduce clutter, only every 20th value is plotted.

$tol = 10^{-4}$ or when a maximum iterate count IterMax $= 100$ was reached. In one of our tests (see the last paragraph of section 4.3), they were terminated on elapsed time.

4.3 Simulation results

Algorithm IrQP was implemented in Matlab R2010b. All tests were run on a desktop machine (Intel(R) Core(TM)i5-2400 CPU @ 3.10G Hz, 4GB of RAM, Windows 7 Enterprise 2009). As already mentioned, our RHC controller was applied to the full 38-state model (1-state augmentation of M2) in lieu of the actual rotorcraft. The initial state was set to zero (i.e., to trim). We simulated the system for 10 sec with a sample time $T_s = 0.01$ sec, so that 1000 CQPs were solved.

Figures 1 and 2 show the evolution of the control inputs and their rates of change (defined by $(u(t) - u(t-1))/T_s$). It can be seen from Figure 2 that the upper or lower bounds on the rates of change were reached at most time steps. Figures 3 and 4 show the comparison of the actual and reference trajectories for the pitch rate and pitch attitude, respectively. Evolution of other states are showed in Figures 5–8. From about 5 sec to 7 sec, the pitch reference trajectory was not followed well. This is due to the bounds imposed on the rates of change of control inputs and to the upper bound placed on the pitch load factor; see Figures 2 and 6. Figure 9 shows the rates of change of the roll and pitch rates, defined by $(p(t) - p(t-1))/T_s$ and $(q(t) - q(t-1))/T_s$, respectively.

We compared in Figures 10 and 11 the CPU time and number of iterations to solve the 1000 CQPs with $|\mathcal{Q}| = 120$—slightly larger than Mm (=90)—and with no constraint reduction (corresponding to $|\mathcal{Q}| = 1360$). To reduce clutter, the results for only every tenth

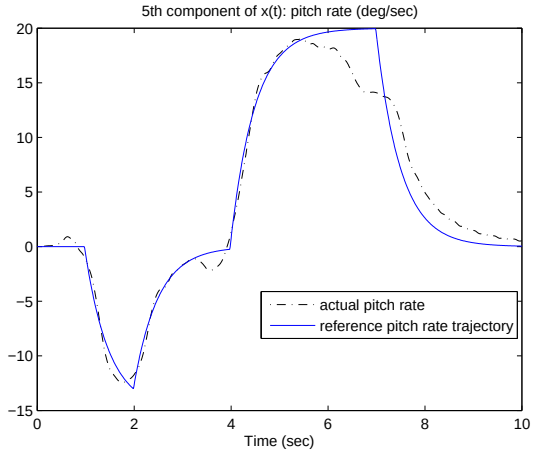

Figure 3: Pitch rate q (no bounds were imposed).

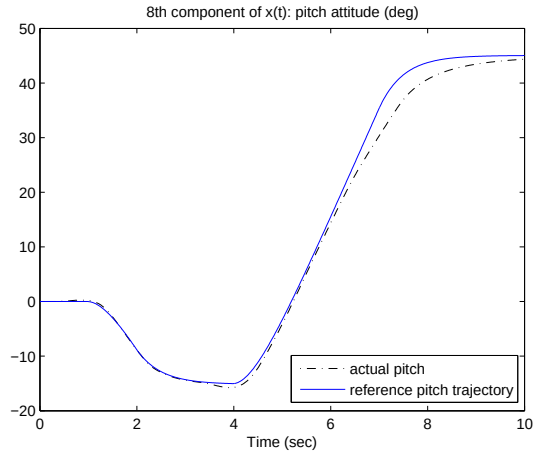

Figure 4: Pitch θ (no bounds were imposed).

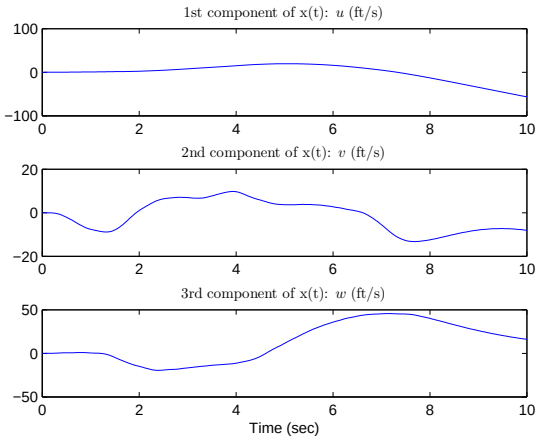

Figure 5: Velocity components (u, v, w) along the body axes (no bounds were imposed).

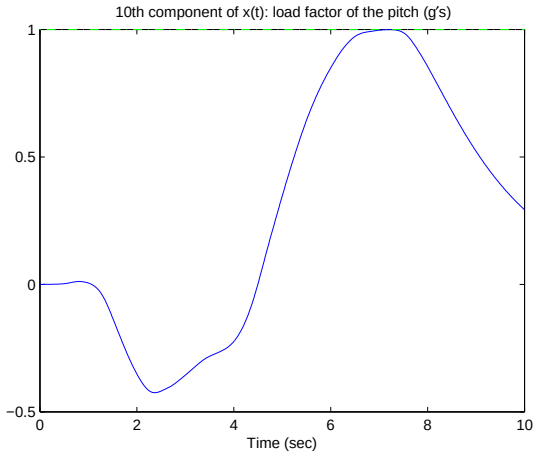

Figure 6: Pitch load factor N_z. The upper bound of 1g is marked by a green dashed line.

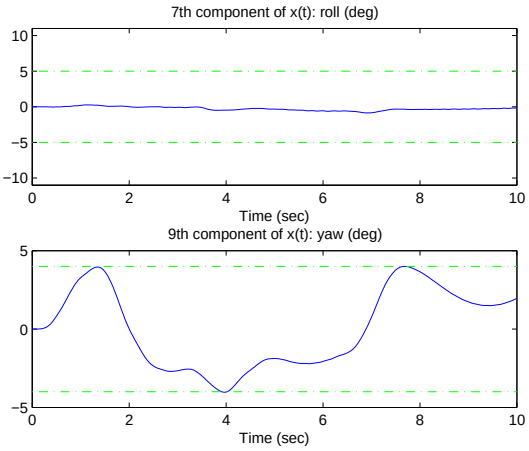

Figure 7: Roll and yaw angles ϕ and ψ. Bounds (of ± 5 degrees and ± 4 degrees, respectively) are marked by green dashed lines.

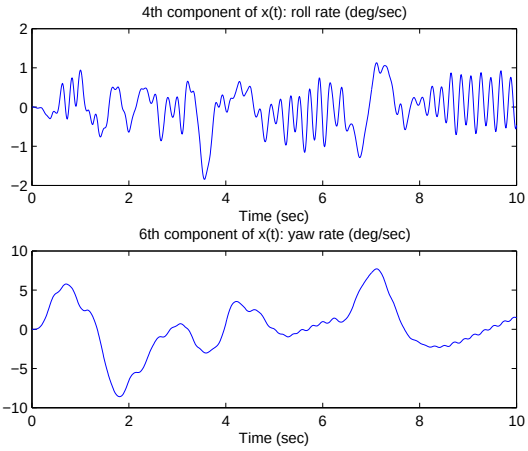

Figure 8: Roll and yaw rates p and r (no bounds were imposed).

CQP are showed. It can be seen from Figure 10 that for each CQP, constraint reduction affords a significant speedup.[3] We also observe from Figure 11 that constraint reduction does not increase the number of iterations compared with no constraint reduction. (Similar observations were made, e.g., in [TAW06, WNTO11].)

Table 2 shows a comparison of total time for solving all 1000 CQPs with four algorithms: functions `qpdantz` and `quadprog` in Matlab, and IrQP with $|\mathcal{Q}| = 120$ and $|\mathcal{Q}| = 1360$. While `qpdantz` (the solver used in the Matlab Model-Predictive-Control Toolbox) takes the longest time, constraint reduction takes the least time.

Table 2: Total CPU time for the solutions of all 1000 CQPs, with different solvers

Method	`qpdantz`	`quadprog`	$\|\mathcal{Q}\| = 1360$	$\|\mathcal{Q}\| = 120$
Time (sec)	5243.2	235.7	45.4	13.3

Figure 12 shows the total time and number of iterations needed to solve all 1000 CQPs with constraint reduction for various *fixed* values of the penalty parameter, i.e, with the adaptation scheme (18)–(20) turned off so the benefit drawn from the automatic penalty adaptation scheme can be assessed. (Algorithm IrQP then becomes Algorithm A of [JOT12].) We recorded the total time and number of iterations for solving the 1000 problems using various values of ρ, from 2×10^6 to 10^9. (The value $\rho = 2 \times 10^6$ turns out to be the smallest value that is large enough for all 1000 CQPs to be correctly solved, as was determined by trial and error. Of course, for other trajectories to be tracked, a larger value may be needed, so a robust fixed-ρ controller would have to use a value much larger than 2×10^6.) It can be

[3]The speedup is notably less than $1360/120 \approx 11$, which only accounts for the computation of $\Delta\mathbf{u}$.

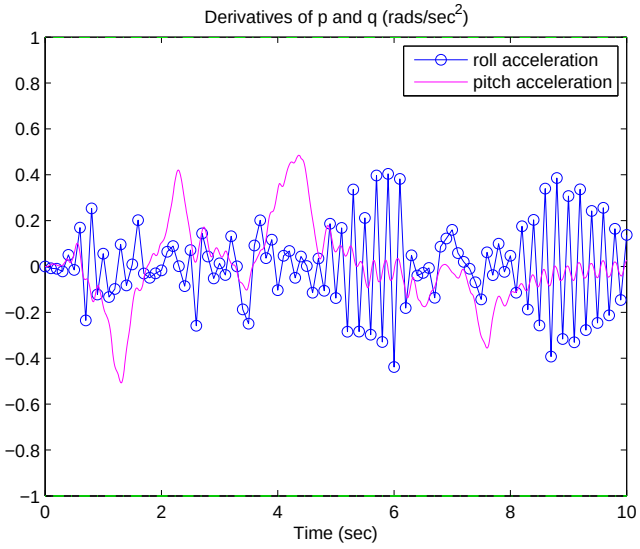

Figure 9: Roll and pitch accelerations $\dot{p}$ and $\dot{q}$. Both have an upper bound 1 rad/sec^2 and a lower bound -1 rad/sec^2 (marked in green dashed lines).

seen from the figure that the adaptive penalty scheme performs better than the scheme with fixed penalty values for most such values, and much better for large such penalty values. For $\rho = 2 \times 10^7$ (i.e., a safety factor of 10), fixed ρ takes roughly 20% more time than the adaptation scheme.

When an RHC-based controller is implemented, the optimization process must be terminated when the next stopping time is reached, at which time the best control value obtained so far is applied to the system being controlled. In Figures 13–16, we compare the performances of IrQP on our trajectory following problem with $|\mathcal{Q}| = 1360$ and $|\mathcal{Q}| = 120$, with the optimization stopped after 0.012 sec (0.010 sec was a little too short a time for our controller with the computer we used for our runs) and the other stopping criteria turned off. It can be seen that the trajectories are followed reasonably closely with constraint reduction, but not closely at all without (Figures 13–14), and the associated constraint violations are much smaller with constraint reduction than without (Figures 15–16).

5 Conclusion

A constraint-reduction scheme was used within a primal-dual affine-scaling interior-point algorithm to efficiently solve CQPs in RHC-based rotorcraft trajectory following. The need for strictly feasible starting points for those CQPs was addressed by introducing an exact ℓ_∞ penalty function. Exactness of the penalty function was ensured by an adaptive adjustment scheme that selected an appropriate large value for the penalty parameter. Numerical simulations show promise. While algorithm IrQP is affine-scaling-based, a Mehrotra-Predictor-Corrector version can also be developed, by combining IrQP with ideas from [WNTO11].

Acknowledgement: The first two authors wish to thank Professor Roberto Celi for his unlimited availability in helping them gain a detailed understanding of the M2 model.

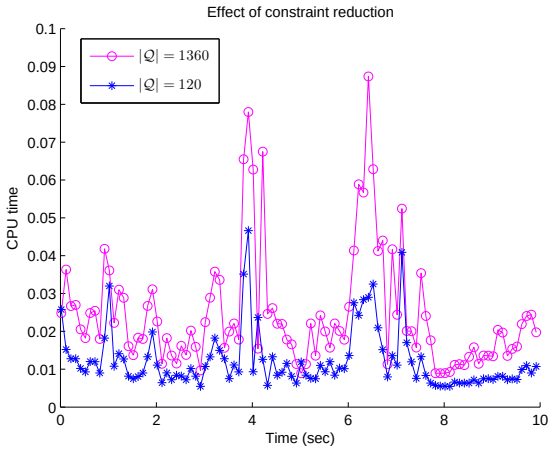

Figure 10: Comparison of CPU time between constraint reduction and no constraint reduction. (Results shown for every 10th CQP.)

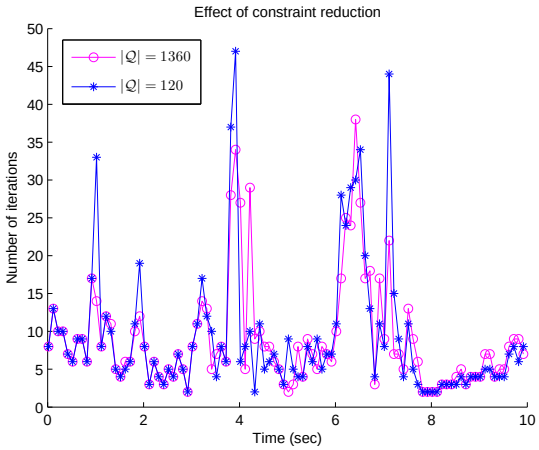

Figure 11: Comparison of iteration counts between constraint reduction and no constraint reduction. (Results shown for every 10th CQP.)

References

[CB04] E. F. Camacho and C. Bordons, *Model predictive control*, 2nd Edition, Springer, 2004.

[dHRT92] D. den Hertog, C. Roos, and T. Terlaky, *Adding and deleting constraints in logarithmic barrier method for linear programming problems*, Shell report, AMER 92-001 (1992).

[DY91] G. Dantzig and Y. Ye, *A build-up interior-point method for linear programming: Affine scaling form*, Tech. report, Department of Management Science, University of Iowa, 1991.

[FM90] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Society for Industrial and Applied Mathmatics, 1990.

[HKT+10] M. Y. He, M. Kiemb, A. L. Tits, A. Greenfield, and V. Sahasrabudhe, *Constraint-reduced interior-point optimization for model predictive rotorcraft control*, Proceedings of the 2010 American Control Conference, 2010, pp. 2088–2094.

[HT11] M. Y. He and A. L. Tits, *An infeasible constraint-reduced interior-point method for linear optimization*, Optimization Methods and Software (DOI: 10.1080/10556788.2011.589056, published on-line as of September 2011).

[HT12] _____, *An infeasible constraint-reduced interior-point method for convex quadratic optimization*, Tech. report, Department of Electrical and Computer Engineering, University of Maryland, In preparation, 2012.

[JOT12] J. H. Jung, D. P. O'Leary, and A. L. Tits, *Adaptive constraint reduction for convex quadratic programming*, Computational Optimization and Applications **5** (2012), no. 1, 125–157.

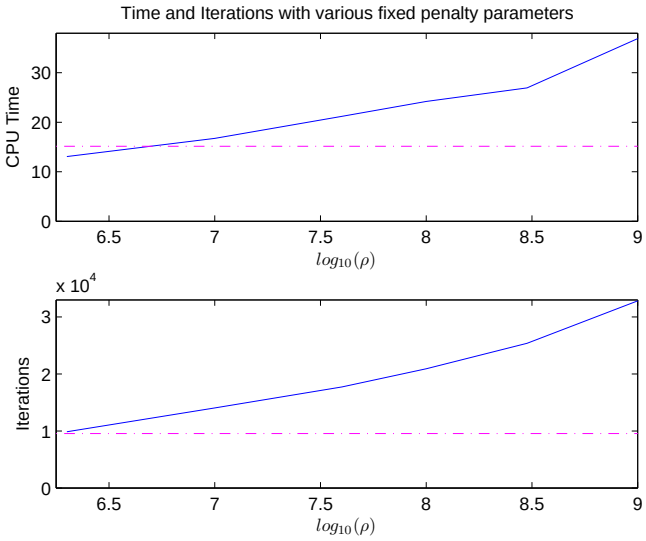

Figure 12: Total time and number of iterations with fixed penalty values ρ and $|Q|$=120. The magenta dash lines mark the total time and number of iterations with the adaptive scheme.

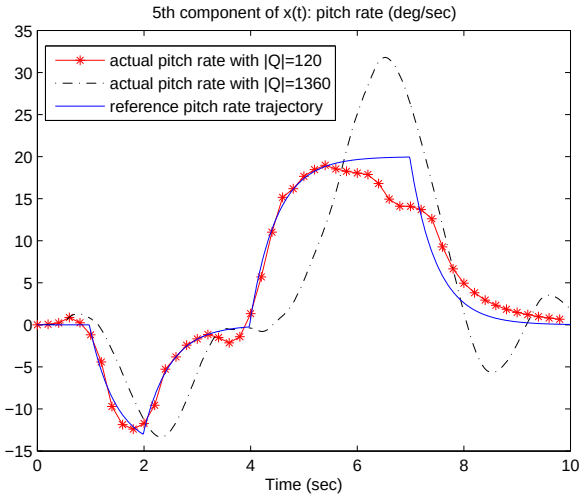

Figure 13: Trajectory following of the pitch rate q with a time limit of 0.012 sec on the solution of the CQPs.

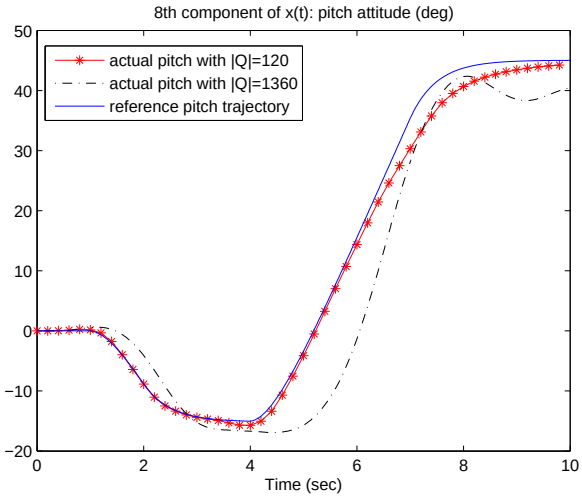

Figure 14: Trajectory following of the pitch θ with a time limit of 0.012 sec on the solution of the CQPs.

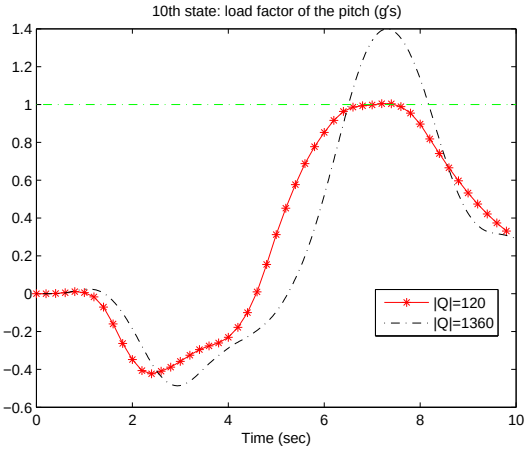

Figure 15: Load factor Nz during the pitch maneuver with a time limit of 0.012 sec on the solution of the CQPs.

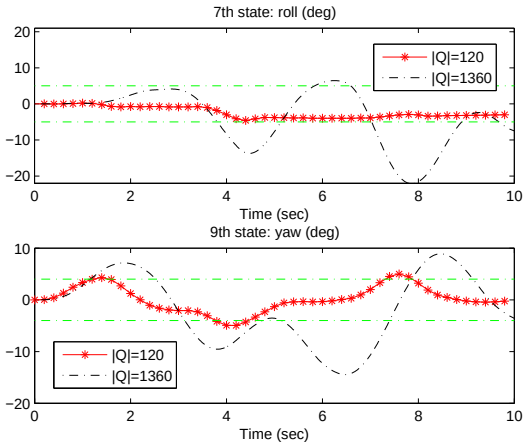

Figure 16: Roll and yaw angles ϕ and ψ with a time limit of 0.012 sec on the solution of the CQPs.

[KB06] T. Keviczky and G. Balas, *Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance*, Journal of Guidance, Control and Dynamics **29** (2006), no. 3, 680–694.

[KLS01] R. Kulhavy, J. Lu, and T. Samad, *Emerging technologies for enterprise optimization in the process industries*, Aiche Symposium Series **98** (2001), no. 326, 352–363.

[MAG+99] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra, and F. Y. Hadaegh, *A model predictive control-based approach for spacecraft formation keeping and attitude control*, Proceedings of the 1999 American Control Conference, 1999, pp. 4258–4262.

[MR94] M. Morari and N. L. Ricker, *Model predictive control toolbox*, Natick, MA: The MathWorks Inc., 1994.

[QB03] S. J. Qin and T. A. Badgwell, *A survey of industrial model predictive control technology*, Control Engineering Practice **11** (2003), no. 7, 733–764.

[TAW06] A. L. Tits, P. A. Absil, and W. P. Woessner, *Constraint reduction for linear programs with many inequality constraints*, SIAM Journal on Optimization **17** (2006), no. 1, 119–146.

[TC02] C. Theodore and R. Celi, *Helicopter flight dynamic simulation with refined aerodynamic and flexible blade modeling*, Journal of Aircraft **39** (2002), no. 4, 577–586.

[WBB01] E. A. Wan, A. A. Bogdanov, and E. A. Bogdanov, *Model predictive neural control with applications to a 6 DoF helicopter model*, Proceedings of the 2001 American Control Conference, 2001, pp. 488–493.

[WNTO11] L. B. Winternitz, S. O. Nicholls, A. L. Tits, and D. P. O'Leary, *A constraint-reduced variant of Mehrotra's predictor-corrector algorithm*, Computational Optimization and Applications (DOI: 10.1007/s10589-010-9389-4, published on-line as of January 2011).

[Ye90] Y. Ye, *A "build-down" scheme for linear programming*, Mathematical Programming **46** (1990), no. 1–3, 61–72.

Appendix. See Tables 3–5 in the next two pages. For states 1 to 3, the units are feet per second, for states 4 to 6, they are radians per second, and for states 7 to 9, radians. (Radians had to be converted to degrees for specifying bounds as well as in the plots.) For the first, second, and fourth control inputs, the units are inches. (The other units are unimportant for the purpose of the present paper.)

Note: If at all possible (and if this paper is accepted for publication), this appendix would best be made available to TCST readers as "electronic supplementary material" rather than being "printed" with the paper.

				Columns	1—16										
-0.0201	0.0019	-0.0040	-0.9649	-5.2500	13.5300	0.0000	-32.1500	0	-1.4150	-12.3100	13.7900	-0.1803	0.1052	-228.1000	-125.1000
-0.0018	-0.0795	-0.0068	4.7780	-1.1960	-133.8000	32.1500	0.0000	0	0.2658	10.7800	10.2900	1.2200	-0.8335	-107.8000	173.6000
-0.0284	0.0017	-0.0593	-13.9800	132.9000	2.2810	0.0002	-1.1550	0	-1.7950	-1.2040	0.7835	0.0161	-3.2560	-15.8300	-20.4200
0.0021	-0.0198	-0.0150	0.0960	-0.6645	0.0882	-0.0000	-0.0000	0	0.4345	8.6080	7.9360	0.8997	-0.8140	-85.9400	140.3000
-0.0038	0.0119	-0.0086	0.0587	-0.4368	0.0005	0.0000	0.0000	0	0.2096	1.1260	-1.3190	0.0402	-0.2193	22.2800	11.7600
-0.0054	0.0095	0.0042	0.0298	0.0722	-0.2163	-0.0000	0.0000	0	-4.1810	1.2830	1.2440	-0.0293	-7.4480	-23.1200	27.0500
0	0	0	1.0000	-0.0000	0.0359	0.0000	-0.0000	0	0	0	0	0	0	0	0
0	0	0	0	1.0000	0.0000	0.0000	0	0	0	0	0	0	0	0	0
0	0	0	0	-0.0000	1.0010	0.0000	-0.0000	0	0	0	0	0	0	0	0
-0.0053	0.0086	0.0502	0.4265	-1.2510	-0.0965	-0.0000	0.0000	0	-16.5700	2.3050	2.4190	-0.4013	-47.6700	-64.0100	60.9900
-0.0078	0.0049	-0.0026	0.3950	0.6735	0.1247	0.0000	-0.0000	0	3.5420	-14.5100	-58.5300	-3.5070	0.4969	743.9000	-371.9000
-0.0127	0.0053	0.0347	0.5807	-1.1010	0.0083	-0.0000	-0.0000	0	3.5380	58.1400	-11.9000	3.8220	-0.7711	341.7000	747.2000
0.0022	-0.0001	0.0059	-0.3309	-0.2658	-0.0087	0.0000	0.0000	0	-0.3135	-1.6580	1.8030	-12.3100	0.0809	-47.5200	-44.0000
0	0	0	0	0	0	0	0	0	1.0000	-0.0000	0.0000	-0.0000	0	-0.0000	-0.0000
0	0	0	0	0	0	0	0	0	-0.0000	1.0000	0.0000	0.0000	0	-0.0000	-0.0000
0	0	0	0	0	0	0	0	0	0.0000	0.0000	1.0000	0.0000	0	0.0000	0.0000
0	0	0	0	0	0	0	0	0	-0.0000	0.0000	0.0000	1.0000	0	-0.0000	0.0000
-0.1199	0.0357	1.2830	4.7740	-1.0880	-1.2170	-0.0000	-0.0000	0	2.0710	-0.8374	-0.3666	-0.1640	7.1380	29.8100	-21.5500
0.0550	0.2303	0.2402	57.0600	-10.3600	4.9720	-0.0000	-0.0000	0	-1.1200	9.4590	2.0940	2.2060	37.2600	-3.1510	165.7000
0.1833	-0.0988	0.3045	-9.8130	-56.4400	-0.0081	-0.0000	0.0000	0	-0.9585	5.4050	10.1100	-0.3650	-0.7794	-150.5000	105.7000
-0.0075	0.0015	-0.0202	0.7550	-2.4050	0.2921	-0.0000	0.0000	0	-0.1630	0.6914	-0.5043	2.1740	0.6652	-6.1910	19.3800
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.0019	-0.0000	0.0068	0.0216	0.0034	-0.0204	0	0	0	0.0211	0.0002	-0.0029	-0.0009	0.0546	0.1788	-0.0066
-0.0007	0.0030	0.0052	0.0357	0.3269	-0.0058	0	0	0	0.0123	0.0445	-0.0297	0.0096	0.4739	0.6512	0.7721
0.0022	0.0029	0.0063	0.3431	-0.0252	0.0378	0	0	0	-0.0186	0.0288	0.0373	-0.0113	0.3236	-0.5418	0.6203
-0.0087	-0.0136	0.0155	-0.2545	0.4497	1.0150	0	0	0	0	0	0	0	0	0	0
-0.0019	0	-0.0336	0	0.0041	0	0	0	0	0	0	0	0	0	0	0
-0.0090	0.0959	0.0027	0	-0.0048	0	0	0	0	0	0	0	0	0	0	0

				Columns	17—31									
-0.8540	0.1289	0.1086	-0.1233	-0.4795	66.3600	32.7800	-19.3300	-4.9590	-1.9490	2.5330	0.6207	-3.2850	0.0272	-0.0034
1.2070	-0.6143	-0.0194	-0.1346	-0.0569	-1.1310	-10.6500	-21.3500	15.7800	15.2500	0.1689	-0.6935	62.1600	0.0063	0.0302
1.1780	-2.1990	-0.0729	-0.1344	0.1053	-582.2000	5.9380	-1.5790	-1.9600	64.8500	0.3351	-5.1790	-27.4800	0.1048	0.0052
2.5670	-0.4784	-0.0326	-0.2127	-0.1057	-0.3960	-25.9900	-40.8800	10.9500	19.9600	1.5990	-3.2940	37.7800	0.0036	0.0023
0.0822	0.0433	-0.0209	0.0153	0.0397	6.9000	-6.4340	4.3740	0.5728	11.8400	-0.5314	-0.1271	-10.0200	0.0345	0.0020
0.1263	-0.0694	0.0088	0.0067	-0.0010	-0.0537	-2.5010	-2.5110	1.0560	-8.9260	-0.2102	-0.3010	-27.8400	-0.0028	-0.0104
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.0275	1.9660	0.8319	-0.9231	-0.1009	-0.7346	6.8890	5.2150	-3.7180	-42.1500	8.3160	-8.3060	-30.9100	-0.0030	-0.0107
-0.5689	1.8400	2.1080	0.0925	-1.5270	-35.0200	8.2020	75.8600	31.5200	0.5991	-13.5700	-8.8930	-15.2600	0.0062	-0.0006
0.8833	-2.0100	-0.0292	1.8520	-1.8310	-33.7300	-65.0700	-11.0100	-29.5900	-20.9100	12.3900	-19.2100	6.1660	0.0112	0.0017
-40.1900	-0.1091	-0.8175	-0.9471	1.9980	-4.1000	43.7600	-39.6900	-0.5813	-3.9600	2.6540	-7.2490	0.1870	-0.0004	-0.0000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.7186	-26.6900	-2.4810	-2.8830	0.5612	-812.2000	18.4500	-12.6200	18.1500	-927.5000	-8.4610	-120.3000	-0.7374	0.0032	0.0002
-36.0900	-5.1830	-27.1800	-54.1000	5.0500	-114.3000	-96.0300	-746.9000	128.1000	-163.4000	-537.1000	-395.5000	12.1100	0.0284	0.0025
1.9110	-6.1460	54.0000	-26.2000	-5.9330	105.1000	690.1000	-71.6000	117.6000	-210.6000	366.0000	-528.9000	33.4200	-0.0161	0.0003
7.2850	0.5592	2.4820	-2.8670	-26.5600	18.2600	137.2000	122.0000	-779.8000	13.3100	69.3700	-18.4100	-0.1782	-0.0000	-0.0000
0	1.0000	-0.0000	0.0000	0.0000	0	-0.0000	-0.0000	0	0	0	0	0	0	0
0	-0.0000	1.0000	0.0000	0.0000	0	-0.0000	0.0000	0	0	0	0	0	0	0
0	0.0000	0.0000	1.0000	0.0000	0	-0.0000	0.0000	0	0	0	0	0	0	0
0	0.0000	0.0000	0.0000	1.0000	0	-0.0000	0.0000	0	0	0	0	0	0	0
-0.0045	-0.1119	-0.0110	-0.0128	0.0019	-0.0461	-0.0388	0.0350	0.0972	-9.4040	-5.7250	-1.1180	0	0	0
-0.2694	-0.0112	-0.2464	0.1658	0.0627	-2.2990	-5.2330	-6.3070	0.3759	39.2700	-42.6600	-2.3470	0	0	0
-0.2219	-0.1089	-0.1785	-0.2380	-0.0136	0.2478	6.1940	-4.6850	1.4610	-0.0298	-2.3350	-21.9800	0	0	0
0	0	0	0	0	0	-0.4351	0.3807	0.0000	-27.3400	0	0	-68.4800	-0.0244	0.0766
0	0	0	0	0	0.0000	-0.1087	0.0951	0.0000	23.8400	0	0	0	-8.0610	0
0	0	0	0	0	0.0000	-0.2701	0.2363	0.0000	-7.1410	0	0	0	0	-8.1920

Table 3: Matrix A^{ct} (31×31) in the continuous-time model M2

Table 4: Matrix B^{ct} (31×4) in the continuous-time model M2

0.0320	-0.0830	-0.3060	0.0447
0.0014	0.0060	-0.0869	-0.0041
0.0500	0.4830	1.3100	-0.2781
0.0508	0.1123	-0.0998	-0.0635
-0.0105	0.0044	-0.0025	-0.0116
0.0013	0.0319	-0.0020	-0.0147
0	0	0	0
0	0	0	0
0	0	0	0
0.2688	0.6665	-0.5589	-0.3794
0.5284	-0.2956	0.2280	0.1672
-0.0835	-0.1345	0.7821	0.0748
0.2979	0.2854	0.1319	-0.1639
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
-0.7208	10.8600	19.0900	-6.2420
-15.5600	18.0100	6.3620	-10.3600
9.3770	28.6700	2.3830	-16.4700
3.6290	2.2650	-0.9356	-1.3010
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
-0.0036	0.0485	0.0817	-0.0279
-0.2484	-0.0263	0.0529	0.0151
-0.0219	0.4660	0.0814	-0.2678
0	0	0.3796	-1.3140
0	0	0	0
0	0	0	0

Table 5: Matrices A and B in the discrete-time model (2)

matrix A (10×10)									
0.9997	0.0001	0.0005	-0.0118	-0.0372	0.1334	0.0000	-0.3215	0	0
-0.0001	0.9990	-0.0001	0.0475	-0.0007	-1.3244	0.3213	0.0000	0	0
-0.0003	-0.0001	0.9931	-0.1122	1.3212	0.0218	-0.0000	-0.0115	0	0
-0.0001	-0.0004	0.0000	0.9747	-0.0040	0.0040	-0.0001	0.0000	0	0
-0.0000	0.0001	0.0001	0.0026	0.9884	-0.0015	0.0000	0.0000	0	0
-0.0000	0.0001	-0.0001	-0.0003	0.0002	0.9943	0.0000	0.0000	0	0
-0.0000	-0.0000	0.0000	0.0099	-0.0000	0.0004	1.0000	-0.0000	0	0
-0.0000	0.0000	0.0000	0.0000	0.0099	-0.0000	0.0000	1.0000	0	0
-0.0000	0.0000	-0.0000	-0.0000	0.0000	0.0100	0.0000	0.0000	1.0000	0
-0.0000	0.0000	0.0000	0.0000	0.0287	-0.0000	0.0000	0.0000	0	0.9931

matrix B (10×4)			
-0.0006	-0.0062	0.0068	0.0049
0.0020	-0.0021	0.0005	-0.0131
0.0062	-0.0384	-0.0829	0.0282
0.0093	-0.0009	0.0012	-0.0063
-0.0000	0.0038	0.0018	-0.0004
0.0004	-0.0001	-0.0002	0.0046
0.0000	-0.0000	0.0000	-0.0000
-0.0000	0.0000	0.0000	-0.0000
0.0000	-0.0000	-0.0000	0.0000
-0.0000	0.0001	0.0000	-0.0000