A Riccati-Based Interior Point Method for Efficient Model Predictive Control of SISO Systems

Hagdrup, Morten; Johansson, Rolf; Jørgensen, John Bagterp

Published in:
IFAC-PapersOnLine

Link to article, DOI:
10.1016/j.ifacol.2017.08.2184

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 10672–10678

A Riccati-Based Interior Point Method for Efficient Model Predictive Control of SISO Systems

Morten Hagdrup * **Rolf Johansson** **
John Bagterp Jørgensen *

* *DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: jbjo@dtu.dk)*
** *Department of Automatic Control, Lund University, PO Box 118, SE-221 00 Lund, Sweden (e-mail: Rolf.Johansson@control.lth.se)*

Abstract: This paper presents an algorithm for Model Predictive Control of SISO systems. Based on a quadratic objective in addition to (hard) input constraints it features soft upper as well as lower constraints on the output and an input rate-of-change penalty term. It keeps the deterministic and stochastic model parts separate. The controller is designed based on the deterministic model, while the Kalman filter results from the stochastic part. The controller is implemented as a primal-dual interior point (IP) method using Riccati recursion and the computational savings possible for SISO systems. In particular the computational complexity scales linearly with the control horizon. No warm-start strategies are considered. Numerical examples are included illustrating applications to Artificial Pancreas technology. We provide typical execution times for a single iteration of the IP algorithm and the number of iterations required for convergence in different situations.

Keywords: Predictive control, constrained optimization, quadratic programming, interior point methods, Riccati iteration, closed-loop control, linear systems, Artificial Pancreas.

1. INTRODUCTION

Model Predictive Control (MPC) is a control methodology that uses a model of the system to be controlled to predict its output over some future horizon. At each time instance a control sequence is computed online by solving an open-loop optimal control problem (OCP) based on the model, the estimated current state, and a reference trajectory. Only the first element of the control sequence is applied to the system and feedback is obtained by repeating this procedure when the next measurement is received. For a comprehensive introduction to MPC the reader is referred to (Rawlings and Mayne, 2009).

MPC has its origins in the process industries but the last decade has seen a widening of the scope of application to encompass also biomedical systems (Zavitsanou et al., 2016). A case in point being the application to Artificial Pancreas (AP) technology for Type 1 Diabetes Mellitus (Bátora et al., 2015) and (Schmidt et al., 2015). Common for the applications to AP is the requirement for the algorithms to be able to run on small portable platforms. This drives a search for efficient implementation of optimization algorithms tailored to the control problem under consideration.

This paper addresses those needs by proposing a linear MPC which

- keeps deterministic and stochastic model parts separate so that optimization is performed only for the deterministic part.
- is based on a Riccati recursion technique that takes advantage of the long horizon relative to system dimension.
- handles soft constraints (both upper and lower) very efficiently.

Wang and Boyd (2010) and Domahidi et al. (2012) report related work on computationally efficient implementations of MPC. Our implementation however stays closer in spirit to the seminal work of Rao et al. (1998). We provide a transcription of the control problem including soft constraints reducing it to a form where the Riccati recursion is applicable. Wherever possible use is made of the fact that the system considered is a SISO system. Sokoler et al. (2015), Frison and Jørgensen (2013) and Jørgensen et al. (2012) provide further examples of applications of the Riccati iteration technique to problems in MPC.

In the comprehensive review (Zavitsanou et al., 2016) of existing embedded control technology for AP no mention is made of Riccati-based implementations. The implementation described in the present contribution therefore seems novel in the context of AP despite the passing of close to 20 years since the publication of (Rao et al., 1998). The implementation described here differs from that of (Rao et al., 1998) by not including the cost-to-go term. Reasons for this choice are given in Section 7. In addition we allow for the specification of a reference trajectory r for the output to track.

The work presented in this paper should be seen as a continuation of that of the contribution (Hagdrup et al.,

10.1016/j.ifacol.2017.08.2184

2016). While the latter paper dealt with filtering, prediction and tuning of the MPC-based control system for AP, here we delve into the detailed implementation of the controller. The reader may find it helpful to consult (Hagdrup et al., 2016) for further elaboration of the signal model used, the description of which will here by necessity be relatively brief.

The paper is structured as follows. Section 2 presents the signal model used. Building on this, Section 3 formulates the OCP and transcribes it to a convex quadratic program (QP). The optimality conditions of the QP are formulated in Section 4 and we show how to solve for the minimizer using an IP algorithm. The efficient implementation of the IP algorithm by means of Riccati recursion is the subject of Section 5. Section 6 provides simulation examples and CPU times and Section 7 concludes the paper.

2. SIGNAL MODEL

We consider a linear system described in continuous time in terms of transfer functions $G(s)$ and $H(s)$ and with discrete measurements $y_k = y(t_k)$ at times $t = t_k = kT_s$:

$$\begin{aligned} Z(s) &= G(s)\,U(s) + H(s)\,W(s) \\ y(t_k) &= z(t_k) + v_k, \qquad k = 0, 1, 2, ... \end{aligned} \tag{1}$$

Here, U denotes the input to the deterministic part of the model and W the white noise input to the stochastic part of the model. G and H are assumed *strictly proper*. Finally, $\{v_k\} \sim N_{iid}(0, r^2)$ is a sequence of independent and identically distributed Gaussian random variables representing the measurement noise. The deterministic input u is subject to the Zero-Order-Hold condition (ZOH) while the discretization of stochastic part involves sampling a Stochastic Differential Equation (SDE) as described in (Hagdrup et al., 2016).

The deterministic part of the system description may be realized as a state space model of the form

$$Z_d(s) = G(s)\,U(s) \sim \begin{cases} x^d_{k+1} &= A_d x^d_k + B_d u_k \\ z^d_k &= C_d x^d_k \end{cases} \tag{2}$$

since G is assumed strictly proper. The stochastic part may likewise be realized as a state-space model

$$Z_s(s) = H(s)W(s) \sim \begin{cases} x^s_{k+1} &= A_s x^s_k + B_s \boldsymbol{w}_k \\ z^s_k &= C_s x^s_k \end{cases} \tag{3}$$

The conditional expectation $\hat{x}^s_{k|k} := E[x^s_k|Y_k]$ of the state vector x^s_k given the observations $Y_k = \{y_0, y_1, \ldots, y_k\}$, is obtained from the Kalman Filter. The Kalman Filter and Predictor result from the transfer functon $H(s)$ as described in (Hagdrup et al., 2016).

3. OPTIMAL CONTROL PROBLEM

We now turn to the issue of optimal control of systems with dynamics as described in Eq. (2). We define the output penalty function π_z by

$$\pi_z(z, r, \chi, \theta) = |z - r|^2 + \kappa\,|\chi|^2 + \eta|\theta|^2 \tag{4}$$

and the objective function by

$$\begin{aligned} \phi = \frac{1}{2}\sum_{j=1}^{N-1} \pi_z(\hat{z}_{k+j|k}, \hat{r}_{k+j|k}, \hat{\chi}_{k+j|k}, \hat{\theta}_{k+j|k}) + \rho|\Delta u_{k+j|k}|^2 \\ + \rho|\Delta u_{k|k}|^2 + |\hat{z}_{k+N|k} - \hat{r}_{k+N|k}|^2 \end{aligned} \tag{5}$$

In (4) and (5) the first term of π_z penalizes deviations of the predicted outputs, $\{\hat{z}_{k+j|k}\}_{j=1}^N$, from the anticipated set-points, $\{\hat{r}_{k+j|k}\}_{j=1}^N$. The second and third terms of π_z represent the penalties associated with the soft lower and upper bounds $\underline{z}_{k+j|k}$ and $\overline{z}_{k+j|k}$ on the output variable:

$$\hat{z}_{k+j|k} \geq \underline{z}_{k+j|k} - \hat{\chi}_{k+j|k} \tag{6a}$$

$$\hat{z}_{k+j|k} \leq \overline{z}_{k+j|k} + \hat{\theta}_{k+j|k} \tag{6b}$$

for $j = 1, ..., N-1$. The final term of ϕ penalizes rate-of-change, $\Delta u_k = u_k - u_{k-1}$, in the manipulated variable. Furthermore the system is subject to (hard) input constraints

$$\underline{u} \leq u_{k+j} \leq \overline{u}. \tag{7}$$

As elaborated in (Hagdrup et al., 2016) the finite horizon optimal control problem (OCP) with objective function (5), subject to Kalman predictions for (1), input constraints (7) and soft output constraints (6) has an equivalent formulation in terms of the modified objective ϕ^*

$$\begin{aligned} \phi^* = \frac{1}{2}\sum_{j=1}^{N-1} \pi_z(\hat{z}^d_{k+j|k}, \hat{r}^*_{k+j|k}, \hat{\chi}_{k+j|k}, \hat{\theta}_{k+j|k}) + \rho|\Delta u_{k+j|k}|^2 \\ + \rho|\Delta u_{k|k}|^2 + |\hat{z}^d_{k+N|k} - \hat{r}^*_{k+N|k}|^2 \end{aligned} \tag{8}$$

and modified reference trajectory ($1 \leq j \leq N$)

$$r^*_{k+j|k} = r_{k+j|k} - \hat{z}^s_{k+j|k}. \tag{9}$$

In fact by introducing the notation

$$\underline{z}^*_{k+j|k} = \underline{z}_{k+j|k} - \hat{z}^s_{k+j|k} \tag{10a}$$

$$\overline{z}^*_{k+j|k} = \overline{z}_{k+j|k} - \hat{z}^s_{k+j|k} \tag{10b}$$

and adjusting the soft constraints to

$$\hat{z}^d_{k+j|k} \geq \underline{z}^*_{k+j|k} - \hat{\chi}_{k+j|k} \tag{11a}$$

$$\hat{z}^d_{k+j|k} \leq \overline{z}^*_{k+j|k} + \hat{\theta}_{k+j|k} \tag{11b}$$

the original OCP is equivalent to solving the constrained optimization problem

$$\begin{aligned} \min_{\{\hat{z}^d_{j+k}, u_{j+k-1}, \hat{\chi}_{j+k}, \hat{\theta}_{j+k}\}_{j=1}^N} \quad & \phi^* \\ s.t. \qquad & (2), (7), (11) \quad . \end{aligned} \tag{12}$$

3.1 Formulation of OCP (12) as a QP

For the formulation of (12) as a QP it is convenient to introduce the augmented state variables

$$\tilde{x}_j = \begin{bmatrix} x_j \\ u_{j-1} \end{bmatrix} \tag{13}$$

as well as the augmented weight matrices

$$Q = \begin{bmatrix} C'C & 0 \\ 0 & \rho \end{bmatrix} \quad \bar{Q} = \begin{bmatrix} C'C & 0 \\ 0 & 0 \end{bmatrix} \quad M = -\begin{bmatrix} 0 \\ \rho \end{bmatrix} \tag{14}$$

and system matrices

$$A = \begin{bmatrix} A_d & 0 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} B_d \\ I \end{bmatrix} \qquad C = [C_d \;\; 0] \tag{15}$$

To keep the notation simple, in the following we shall omit the asterisks for the modified reference trajectory and limits on z. Furthermore we assume that $k = 0$ and write conditional expectations as $x_j = \hat{x}_{0+j|0}$. Using these

conventions, routine calculations show that the governing dynamics may be expressed as

$$\tilde{x}_{j+1} = A\tilde{x}_j + Bu_j \tag{16}$$

and that up to a constant term, the objective function ϕ^* equals the function

$$\begin{aligned}\Psi = \frac{1}{2}\sum_{j=1}^{N-1}\begin{bmatrix}\tilde{x}_j\\u_j\end{bmatrix}'\begin{bmatrix}Q & M\\M' & \rho\end{bmatrix}\begin{bmatrix}\tilde{x}_j\\u_j\end{bmatrix} + \kappa|\chi_j|^2 + \eta|\theta_j|^2 \\ + \frac{1}{2}\tilde{x}_N'\bar{Q}\tilde{x}_N + \frac{1}{2}\rho u_0^2 - \sum_{j=1}^{N} r_j'C\tilde{x}_j - \rho u_{-1}u_0\end{aligned} \tag{17}$$

Since $\kappa, \eta \geq 0$, $\rho > 0$ and $\bar{Q}$ is positive semidefinite, convexity of Ψ results from the easily verifiable positive semidefiniteness of the Schur complement of element ρ of $\begin{bmatrix}Q & M\\M' & \rho\end{bmatrix}$. For the highly structured quadratic function (17) it is useful to view the Hessian as a block diagonal matrix

$$\mathcal{H} = \mathrm{diag}(\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{N-1}, \mathcal{H}_N) \tag{18}$$

where for $1 \leq j \leq N-1$

$$\mathcal{H}_j = \begin{bmatrix}Q & M & 0 & 0\\M' & \rho & 0 & 0\\0 & 0 & \kappa & 0\\0 & 0 & 0 & \eta\end{bmatrix} \tag{19}$$

while $\mathcal{H}_0 = \rho$ and $\mathcal{H}_N = \bar{Q}$. By similarly introducing

$$\xi = \begin{bmatrix}\xi_0' & \xi_1' & \cdots & \xi_N'\end{bmatrix}' \tag{20}$$

where for $1 \leq j \leq N-1$

$$\xi_j = \begin{bmatrix}\tilde{x}_j' & u_j' & \chi_j' & \theta_j'\end{bmatrix}' \tag{21}$$

while $\xi_0 = u_0$ and $\xi_N = \tilde{x}_N$ we see that the sum of the 2^{nd} order terms in (17) may be written compactly as $\frac{1}{2}\xi'\mathcal{H}\xi$. Now define $g = \begin{bmatrix}g_0' & g_1' & \cdots & g_N'\end{bmatrix}'$ where for $1 \leq j \leq N-1$

$$g_j = [-r_jC \;\; 0 \;\; 0 \;\; 0]' \tag{22}$$

while $g_0 = -\rho u_{-1}$ and $g_N = -(r_NC)'$. By doing so the sum of the 1^{st} order terms in (17) becomes $g'\xi$. Altogether this brings the objective Ψ into the standard form

$$\Psi(\xi) = \frac{1}{2}\xi'\mathcal{H}\xi + g'\xi. \tag{23}$$

As for the constraints the equalities (16) may be expressed in the form $\mathcal{A}'\xi = b$ for suitably chosen $\mathcal{A}$ and b. For $N = 2$

$$\mathcal{A}' = \begin{array}{c}\begin{matrix}u_0 & \tilde{x}_1 & u_1 & \chi_1 & \theta_1 & \tilde{x}_2\end{matrix}\\\begin{bmatrix}-B & I & 0 & 0 & 0 & 0\\0 & -A & -B & 0 & 0 & I\end{bmatrix}\end{array}, \qquad b = \begin{bmatrix}A\tilde{x}_0\\0\end{bmatrix}. \tag{24}$$

will be our choice of sign convention and which we stick to also for higher N. The structure of $\mathcal{A}'$ is hinted at by indicating which variable each coefficient corresponds to. This partitioning into stages will be crucial to exploiting the inherent structure of the optimization problem. Turning to the inequality constraints, we note that for each stage $1 \leq j \leq N-1$ the constraints (7) and (11) may be cast in the form $\mathcal{C}_j'\xi_j \geq d_j$ where

$$\left[\,\mathcal{C}_j' \middle| d_j\,\right] = \begin{array}{c}\begin{matrix}\tilde{x}_j & u_j & \chi_j & \theta_j\end{matrix}\\\left[\begin{array}{cccc|c}C & 0 & I & 0 & \underline{z}_j\\-C & 0 & 0 & I & -\overline{z}_j\\0 & I & 0 & 0 & \underline{u}\\0 & -I & 0 & 0 & -\overline{u}\end{array}\right]\end{array}\begin{matrix}\\\underline{z}_j\\\overline{z}_j\\\underline{u}_j\\\overline{u}_j\end{matrix} \tag{25}$$

The rightmost column contains the index which we shall use to refer to that particular inequality. It provides a convenient way of addressing the slack variables s, and associated dual variables t. For example $\lambda_{\underline{z}_j}$ denotes the Lagrange multiplier pertaining to the inequality constraint expressed by the first row of (25). This notation will be used consistently in Section 5. For $j = 0$ the matrices $\mathcal{C}_0'$ and d_0 comprise only the data pertaining to u_0, that is rows 3 and 4. Stage N is subject to no inequality constraints so $\mathcal{C}_N' = 0$ and $d_N = 0$. By introducing the block diagonal matrix $\mathcal{C}$ and the stacked column vector d

$$\begin{aligned}\mathcal{C} &= \mathrm{diag}(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_N)\\ d &= [d_0 \;\; d_1 \;\; \ldots \;\; d_N]'\end{aligned} \tag{26}$$

we may now express our OCP (12) as a convex QP in the standard form

$$\begin{aligned}\min_{\xi} \quad & \frac{1}{2}\xi'\mathcal{H}\xi + g'\xi\\ s.t. \quad & \mathcal{A}'\xi = b\\ & \mathcal{C}'\xi \geq d\end{aligned} \tag{27}$$

4. OPTIMALITY CONDITIONS

The convexity of (27) means that a value ξ of the decision variable is a minimizer of (27) if and only if ξ and its pair of associated dual variables satisfy the *Karush-Kuhn-Tucker* (KKT) conditions (Nocedal and Wright, 2006). To formulate these optimality conditions for (27) we introduce non-negative slack variables s allowing us to recast the inequality constraints as

$$-\mathcal{C}'\xi + s + d = 0 \tag{28}$$

It now follows that ξ is a minimizer if and only if there exist vectors p and $s, t \geq 0$ (component-wise) such that

$$F(\xi, p, t, s) = \begin{bmatrix}r_L\\r_{\mathcal{A}}\\r_{\mathcal{C}}\\r_{\mathcal{ST}}\end{bmatrix} = \begin{bmatrix}\mathcal{H}\xi + g - \mathcal{A}p - \mathcal{C}t\\-\mathcal{A}'\xi + b\\-\mathcal{C}'\xi + s + d\\\mathcal{ST}e\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{29}$$

Here $\mathcal{S}$ denotes the diagonal matrix formed by the elements $s_1, s_2, \ldots, s_{4N-2}$ of s. Matrix $\mathcal{T}$ is defined analogously and e denotes the column vector whose entries all equal 1.

4.1 Interior Point Method

This subsection describes an Interior Point Method (Wright, 1997) for iterative solution of (29). The algorithm tracks the so-called *central path* connecting an initial point (ξ^0, p^0, s^0, t^0) to a solution (ξ, p, s, t) of (29). Given a current iterate (ξ, p, s, t) we define a *complementarity measure*

$$\mu = \frac{s't}{\mathrm{card}(s)} = \frac{s't}{4N-2} \tag{30}$$

and consider the *perturbed KKT conditions* (Nocedal and Wright, 2006):

$$F(\xi,p,t,s)=\begin{bmatrix}\mathcal{H}\xi+g-\mathcal{A}p-\mathcal{C}t\\ -\mathcal{A}'\xi+b\\ -\mathcal{C}'\xi+s+d\\ \mathcal{S}\mathcal{T}e\end{bmatrix}=\begin{bmatrix}0\\0\\0\\ \mu\sigma e\end{bmatrix} \tag{31}$$

The solutions of (31) for all positive values of σ and μ define the so-called *central path*, which is a trajectory that leads to the solution of (29) as the product $\mu\sigma$ tends to zero. To track the central path we employ a variant of Mehrotra's predictor-corrector method (Mehrotra, 1992), (Nocedal and Wright, 2006). The method consists in repeating a two-step procedure until convergence. The first so-called *affine step* updates the centering paramter σ and computes second-order correction terms. Next a *corrector step* is determined and a new iterate is produced. The direction of the affine step equals that of the pure Newton step for (31) with parameter $\sigma=0$

$$J_F(\zeta^k)\Delta\zeta^k_{\text{aff}}=-F(\zeta^k) \tag{32}$$

where $J_F(\zeta^k)$ denotes the current value of the Jacobian of F, $\Delta\zeta^k_{\text{aff}}$ is the affine direction, and ζ^k is the current iterate

$$\zeta^k=\begin{bmatrix}(\xi^k)' & (p^k)' & (s^k)' & (t^k)'\end{bmatrix}' \tag{33}$$

With $\Delta\zeta^k_{\text{aff}}$ in hand the affine variables s^k_{aff} and t^k_{aff} are computed

$$s^k_{\text{aff}}:=s^k+\alpha^k_{\text{aff}}\Delta s^k_{\text{aff}}\qquad t^k_{\text{aff}}:=t^k+\alpha^k_{\text{aff}}\Delta t^k_{\text{aff}} \tag{34}$$

The scaling factor α^k_{aff} is introduced to ensure that the constraints $(s,t)\geq 0$ remain satisfied:

$$\alpha^k_{\text{aff}}:=\max\left\{a_{\text{aff}}\in[0;1]|\begin{bmatrix}s^k\\t^k\end{bmatrix}+a_{\text{aff}}\begin{bmatrix}\Delta s^k\\ \Delta t^k\end{bmatrix}\geq 0\right\} \tag{35}$$

We emphasize that we apply the same damping factor to the primal and the dual variables. This differs from what is usually done for linear programming where two separate damping factors are specified Nocedal and Wright (2006). Following Mehrotra (1992), Nocedal and Wright (2006) we update the centering parameter σ by

$$\sigma^k:=\left(\frac{(s^k_{\text{aff}})'t^k_{\text{aff}}}{(s^k)'t^k}\right)^3. \tag{36}$$

In the second step, we obtain $\Delta\zeta^k$ by solving (32) for a modified right hand side, namely

$$-\begin{bmatrix}r_L\\r_A\\r_C\\ \bar{r}_{\mathcal{S}\mathcal{T}}\end{bmatrix}=-\begin{bmatrix}r_L\\r_A\\r_C\\ r_{\mathcal{S}\mathcal{T}}+\Delta\mathcal{S}^k_{\text{aff}}\Delta\mathcal{T}^k_{\text{aff}}e-\sigma^k\mu^k e.\end{bmatrix} \tag{37}$$

Matrix $\Delta\mathcal{S}^k_{\text{aff}}$ is the diagonal matrix formed by the elements of vector Δs^k_{aff}. As in (35) for the affine step, we select the largest scaling parameter α such that $s^k+\alpha\Delta s^k$ and $t^k+\alpha\Delta t^k$ remain non-negative. We update the iterates after damping α with a factor $\tau\in[0.95;0.999]$ to ensure that iterates stay in the interior of the feasible set:

$$\begin{aligned}\xi^k&:=\xi^k+\tau\alpha^k\Delta\xi^k & p^k&:=p^k+\tau\alpha^k\Delta p^k\\ s^k&:=s^k+\tau\alpha^k\Delta s^k & t^k&:=t^k+\tau\alpha^k\Delta t^k\end{aligned} \tag{38}$$

Iteration continues until

$$\begin{aligned}\mu^k&\leq \texttt{tol}_\mu\\ ||(r^k_L,r^k_{\mathcal{A}},r^k_{\mathcal{C}})||_\infty&\leq \texttt{tol}_r||\mathcal{H},\mathcal{A},\mathcal{C},b,d,g||_\infty\end{aligned} \tag{39}$$

where $\texttt{tol}_r$ and $\texttt{tol}_\mu$ are user-defined small tolerances whose default values are 10^{-8} (Gertz and Wright, 2003).

5. RICCATI ITERATION PROCEDURE

In interior point methods such as the one presented in this paper, the main computational effort is spent solving the linear system (32) and its counterpart with a modified right hand side (37). We therefore seek to exploit the inherent structure of the problem. The explicit formulation of (32) becomes

$$\begin{bmatrix}\mathcal{H}&-\mathcal{A}&-\mathcal{C}&0\\ -\mathcal{A}'&0&0&0\\ -\mathcal{C}'&0&0&I\\ 0&0&\mathcal{S}&\mathcal{T}\end{bmatrix}\begin{bmatrix}\Delta\xi\\ \Delta p\\ \Delta t\\ \Delta s\end{bmatrix}=-\begin{bmatrix}r_L\\ r_{\mathcal{A}}\\ r_{\mathcal{C}}\\ r_{\mathcal{S}\mathcal{T}}\end{bmatrix}. \tag{40}$$

By eliminating Δs and Δt one obtains

$$\begin{aligned}\Delta t&=-(\mathcal{S}^{-1}\mathcal{T})\mathcal{C}'\Delta\xi+(\mathcal{S}^{-1}\mathcal{T})(r_{\mathcal{C}}-\mathcal{T}^{-1}r_{\mathcal{S}\mathcal{T}})\\ \Delta s&=-\mathcal{T}^{-1}r_{\mathcal{S}\mathcal{T}}-\mathcal{T}^{-1}\mathcal{S}\Delta t\end{aligned} \tag{41}$$

and the resulting so-called *augmented system* (Nocedal and Wright, 2006) to solve

$$\begin{bmatrix}\mathcal{H}+\mathcal{C}(\mathcal{S}^{-1}\mathcal{T})\mathcal{C}'&-\mathcal{A}\\ -\mathcal{A}'&0\end{bmatrix}\begin{bmatrix}\Delta\xi\\ \Delta p\end{bmatrix}=\begin{bmatrix}r_\xi\\ r_p\end{bmatrix} \tag{42}$$

where

$$\begin{bmatrix}r_\xi\\ r_p\end{bmatrix}=\begin{bmatrix}-r_L+\mathcal{C}(\mathcal{S}^{-1}\mathcal{T})(r_{\mathcal{C}}-\mathcal{T}^{-1}r_{\mathcal{S}\mathcal{T}})\\ -r_{\mathcal{A}}\end{bmatrix} \tag{43}$$

We note that $r_\xi=\begin{bmatrix}r'_{\xi,0} & r'_{\xi,1} & \dots & r'_{\xi,N}\end{bmatrix}'$ where

$$r_{\xi,j}=\begin{bmatrix}r'_{\tilde{x}_j} & r_{u_j} & r_{\chi_j} & r_{\theta_j}\end{bmatrix}' \tag{44}$$

for $1\leq j\leq N-1$ while $r_{\xi,0}=r_{u_0}$ and $r_{\xi,N}=r_{\tilde{x}_N}$.

Computing Δt and the term $(\mathcal{S}^{-1}\mathcal{T})(r_{\mathcal{C}}-\mathcal{T}^{-1}r_{\mathcal{S}\mathcal{T}})$ only involves cheap element-wise operations on vectors. The latter's multiplication with $\mathcal{C}$ is handled by a stage-wise approach utilizing the structure $\mathcal{C}=\text{diag}(\mathcal{C}_0,\mathcal{C}_1,\dots,\mathcal{C}_{N-1})$. The block-diagonal structure of $\mathcal{C}$ ensures that $\mathcal{H}+\mathcal{C}(\mathcal{S}^{-1}\mathcal{T})\mathcal{C}'$ has exactly the same block-diagonal structure as $\mathcal{H}$. By introducing the diagonal matrix $\Lambda=\mathcal{S}^{-1}\mathcal{T}$ one may write $\mathcal{H}+\mathcal{C}\Lambda\mathcal{C}'=\text{diag}(\mathcal{H}_0+\mathcal{C}_0\Lambda_0\mathcal{C}'_0,\dots,\mathcal{H}_{N-1}+\mathcal{C}_{N-1}\Lambda_{N-1}\mathcal{C}'_{N-1},\mathcal{H}_N)$ where

$$\mathcal{H}_j+\mathcal{C}_j\Lambda_j\mathcal{C}'_j=\begin{matrix}\\ \tilde{x}_j\\ u_j\\ \chi_j\\ \theta_j\end{matrix}\begin{matrix}\begin{matrix}\tilde{x}_j & u_j & \chi_j & \theta_j\end{matrix}\\ \begin{bmatrix}Q_j&M&E'_j&F'_j\\ M'&R_j&0&0\\ E_j&0&G_j&0\\ F_j&0&0&K_j\end{bmatrix}\end{matrix} \tag{45}$$

for $1\leq j\leq N-1$ while $\mathcal{H}_0+\mathcal{C}_0\Lambda_0\mathcal{C}_0=R_0$ and $\mathcal{H}_N=Q_N=\bar{Q}$. The quantities appearing in (45) are given by

$$\begin{aligned}E_j&=\lambda_{\underline{z}_j}C & G_j&=\kappa+\lambda_{\underline{z}_j}\\ F_j&=-\lambda_{\overline{z}_j}C & K_j&=\eta+\lambda_{\overline{z}_j}\end{aligned} \tag{46}$$

for $1\leq j\leq N-1$ while

$$\begin{aligned}R_j&=\rho+\lambda_{\underline{u}_j}+\lambda_{\overline{u}_j} & j&=0,\dots,N-1\\ Q_j&=Q+C'(\lambda_{\underline{z}_j}+\lambda_{\overline{z}_j})C & j&=1,\dots,N-1.\end{aligned} \tag{47}$$

Using the indexing convention $p=\begin{bmatrix}p'_0 & p'_1 & \dots & p'_{N-1}\end{bmatrix}'$ we now group all the equatons appearing in (42) stage-wise and eliminate the quantities $\Delta\chi_j$ and $\Delta\theta_j$. We end up with the following equations to solve:

$$\begin{aligned}-\Delta p_{j-1}+\tilde{Q}_j\Delta\tilde{x}_j+M\Delta u_j+A\Delta p_j&=\tilde{r}^{\tilde{x}}_j\\ M'\Delta\tilde{x}_j+R_j\Delta u_j+B'\Delta p_j&=r^u_j\\ A\Delta\tilde{x}_j+B\Delta\tilde{u}_j-\Delta\tilde{x}_{j+1}&=r^p_j\end{aligned} \tag{48}$$

for $j = 1, \ldots, N-1$ in addition to the equations

$$\begin{aligned} R_0 \Delta u_0 + B' \Delta p_0 &= r_0^u \\ B \Delta u_0 - \Delta \tilde{x}_1 &= r_0^p \\ -\Delta p_{N-1} + \tilde{Q}_N \Delta \tilde{x}_N &= \tilde{r}_N^{\tilde{x}} \end{aligned} \tag{49}$$

In (48) and (49) matrices R_j are as specified in (47) while

$$\tilde{Q}_j = Q + \left(\frac{\lambda_{\underline{z}_j} \kappa}{\kappa + \lambda_{\underline{z}_j}} + \frac{\lambda_{\overline{z}_j} \eta}{\eta + \lambda_{\overline{z}_j}} \right) C'C \tag{50}$$

for $1 \leq j \leq N-1$ and $\tilde{Q}_N = \bar{Q}$. Using $r^{\tilde{x}_j}$ from (44) the quantities $\tilde{r}_j^{\tilde{x}}$ are calculated by $\tilde{r}_N^{\tilde{x}} = r_N^{\tilde{x}}$ and

$$\tilde{r}_j^{\tilde{x}} = r_j^{\tilde{x}} + \left(\frac{\lambda_{\overline{z}_j}}{\eta + \lambda_{\overline{z}_j}} r_j^{\theta} - \frac{\lambda_{\underline{z}_j}}{\kappa + \lambda_{\underline{z}_j}} r_j^{\chi} \right) C' \tag{51}$$

for $1 \leq j \leq N-1$. Having solved for $\Delta \tilde{x}_j$,Δu_j and Δp_j one computes $\Delta \chi_j$ and $\Delta \theta_j$ for $1 \leq j \leq N-1$ straightforwardly from

$$\begin{aligned} \Delta \chi_j &= \frac{1}{\kappa + \lambda_{\underline{z}_j}} (r_j^{\chi} - \lambda_{\underline{z}_j} C \Delta \tilde{x}_j) \\ \Delta \theta_j &= \frac{1}{\eta + \lambda_{\overline{z}_j}} (r_j^{\theta} + \lambda_{\overline{z}_j} C \Delta \tilde{x}_j) \end{aligned} \tag{52}$$

We note that (48) and (49) are in a form that permit their efficient solution by the Riccati recursion technique (Rao et al., 1998). In fact for the special case $N = 3$, the equations may be arranged in the well-known form:

$$\begin{bmatrix} R_0 & B' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -I & \tilde{Q}_1 & M & A' & 0 & 0 & 0 & 0 \\ 0 & 0 & M' & R_1 & B' & 0 & 0 & 0 & 0 \\ 0 & 0 & A & B & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -I & \tilde{Q}_2 & M & A' & 0 \\ 0 & 0 & 0 & 0 & 0 & M' & R_2 & B' & 0 \\ 0 & 0 & 0 & 0 & 0 & A & B & 0 & -I \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I & \tilde{Q}_3 \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta p_0 \\ \Delta \tilde{x}_1 \\ \Delta u_1 \\ \Delta p_1 \\ \Delta \tilde{x}_2 \\ \Delta u_2 \\ \Delta p_2 \\ \Delta \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} r_0^u \\ r_0^p \\ \tilde{r}_1^x \\ r_1^u \\ r_1^p \\ \tilde{r}_2^x \\ r_2^u \\ r_2^p \\ \tilde{r}_3^x \end{bmatrix} \tag{53}$$

6. NUMERICAL EXAMPLES

This section presents a case study relevant for an Artificial Pancreas (Bátora et al., 2015). We apply the MPC algorithm developed above to a family of transfer functions often used to describe linearized glucose-insulin dynamics. The (deterministic) model is given in terms of a transfer function of the form

$$G(s) = \frac{k_d}{(1 + \tau_d s)^{n_d}} \tag{54}$$

where n_d is a small integer, typically equal to 2 or 3 (Boiroux et al., 2015). We consider a stochastic model of similar form

$$H(s) = \frac{k_s}{(1 + \tau_s s)^{n_s}} \tag{55}$$

For the Artificial Pancreas it is not uncommon to consider prediction horizons of up to 24 hours and for a sampling rate of $T_s = 5\,\text{min}$ this amounts to 288 samples. In this example we consider $T_s = 1\,\text{min}$ and a horizon of $N = 300$. We consider system and model both to be of the form (1) with G and H of the form (54) and (55) respectively. The relevant simulation parameters are listed in the tables below. The value for the weight ρ on the rate-of-input is arrived at by considering a Pareto-plot as in (Hagdrup et al., 2016). This is obtained by mapping integrated squared output error against integrated squared rate-of-input for values of ρ ranging through a wide interval. We pick the value of ρ ensuring the best compromise for keeping both the mapped quanties low.

Table 1. System and Model parameters plus measurement noise variance

Parameter	System	Model
n_d	2	2
τ_d	5	5
k_d	-1	-1
n_s	2	1
τ_s	3	2.5
k_s	0.3	1.2
σ_v^2	10^{-4}	10^{-4}

Table 2. Cost function weights, tolerances, prediction horizon, sampling interval and number of samples simulated

Parameter	Value
ρ	$10^{-4.75}$
κ	100
η	10
$\texttt{tol}_r$	10^{-8}
$\texttt{tol}_\mu$	10^{-8}
N	300
T_s	1
N_{sim}	600

Table 3. Limits on u (hard) and z (soft)

Parameter	Value
$u_{\min}$	-50
$u_{\max}$	50
$z_{\min}$	-3
$z_{\max}$	3

We perform closed-loop simulations for a tracking scenario where the system output is required to track a reference trajectory which is identically zero except for the intervals between samples 50 and 100 and 450 and 500 respectively. On those two intervals the reference value is 3, coinciding thus with the upper soft bound on the output. Figure 1 shows the result of the simulations. One notes that the input constraints are active around the changes in reference value. We plot in Fig. 2 the number of iterations required in the interior point algorithm when at each time instance of the closed-loop simulation the (open-loop) input profile is calculated. The iteration number is seen to vary and remembering that the prediction horizon is $N = 300$ we see that the times for which most iterations (10 or 11) are needed seem to coincide with the times where an input constraint is active somewhere in the prediction window. The lowest number of iterations, namely 4, occurs when we just require to track the constant value 0 over the entire prediction horizon. Of course we do not imply that the iteration numbers here are necessarily typical. They

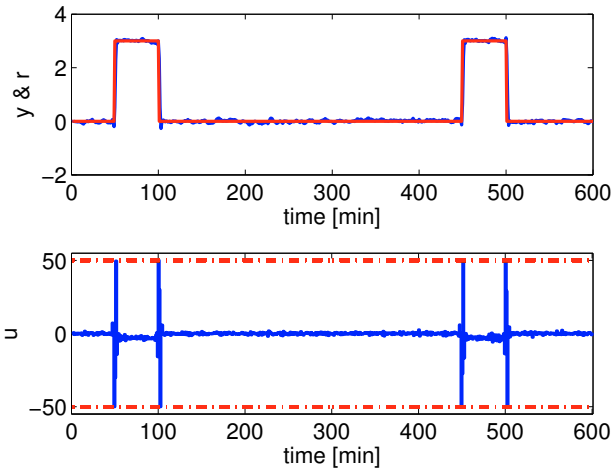

Fig. 1. Top: process output y (blue) and reference r (red). Bottom: control signal u (blue) with hard input limits shown by dashed red lines.

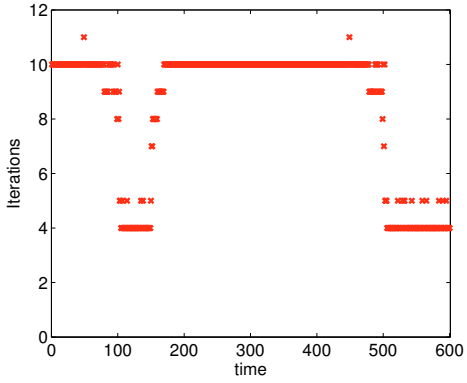

Fig. 2. Number of iterations spent in the interior-point algorithm calculating the input profile at a given time instance. Horizon $N = 300$ and system order $n_s = 2$.

merely serve to illustrate the variation in iteration number which may occur. It should also be noted that the number of iterations required to reach a specified tolerance may depend upon how well-conditioned the resulting matrices are which appear in the systems to be solved. The entire MPC simulation chain including the Riccati based IP solver was implemented in Matlab R2014a. For algorithm performance evaluation we note that the calculations were performed on a DELL laptop equipped with an Intel(R) Core (TM) i5-4310U CPU@2.00 GHz processor and 16.0 GB of memory. Due to the computer's internal scheduling the time spent processing a single interior point iteration will of course fluctuate. This is also clear from the histogram in Fig. 3. Generally speaking though each iteration seems to take approximately 50 ms. This is for a prediction horizon of length 300 and a linear model whose deterministic part is of order $n_d = 2$. For the implemented Riccati iteration based optimization scheme one observes indeed by varying the horizon N an approximately linear growth in CPU time with N. For the scenario described above we now allow the horizon to vary while retaining the value of the tuning parameter ρ at its value corresponding to $N = 300$. For each value of N we map in Fig. 4 the

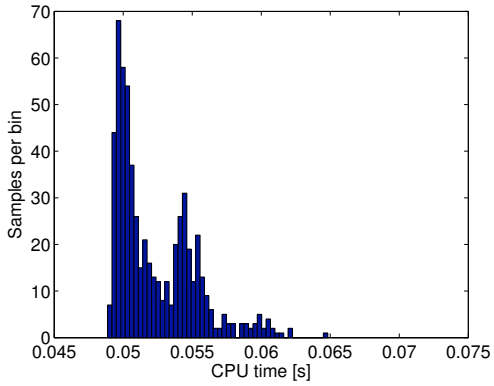

Fig. 3. Histogram of average times spent per interior point iteration calculating the input profile. Horizon $N = 300$ and system order $n_d = 2$.

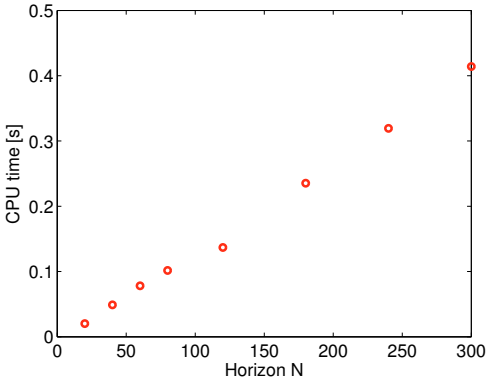

Fig. 4. Average CPU time spent calculating the input sequence at each step.

average CPU time spent calculating the open-loop input sequence at each step. The CPU time is seen to grow approximately linearly with N.

7. CONCLUSION

In applications of MPC it is desirable to use long prediction horizons for stability reasons. Hence one would rather err on the side of caution than pick a horizon N that turns out to be too short. It is therefore useful to have an implementation whose computational complexity only grows linearly with N. One might argue that similar benefits could be achieved by using a dual-mode approach including a suitable end-of-horizon cost-to-go term. This approach, however, presents an additional challenge in the form of selection of a proper weighting matrix for the cost-to-go term. The presence of constraints makes this selection a non-trivial task. The weighting matrix would thus become an extra tuning parameter, something we may avoid by choosing the prediction horizon sufficiently large. This is a viable option thanks to the fast Riccati-based implementation.

A computationally efficient controller obviously has a beneficial effect on battery lifetime, an issue of potential concern when implemented on a small portable platform. While it also makes it easier to meet real-time constraints, there is even a third benefit to having a Riccati-based

controller. Most development and testing of AP technology takes place *in silico* and the ability to increase the number of realizations when performing stochastic simulation is clearly welcome.

The implementation also implies a significant reduction in memory consumption. By condensing and using dense storage, for a scenario with horizon N, the Hessian would still be of size $(3N \times 3N)$. Our approach uses significantly less than that. It has been noted that the algorithm presented in this paper does not include any warm-start strategy. The inclusion of warm-starting might well further reduce computation times for the controller but remains an item for future study.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the constructive criticism given by the anonymous reviewers. Their comments and suggestions have certainly served to improve this paper.

REFERENCES

Bátora, V., Tárnik, M., Murgaš, J., Schmidt, S., Nørgaard, K., Poulsen, N.K., Madsen, H., and Jørgensen, J.B. (2015). Bihormonal control of blood glucose in people with type 1 diabetes. *Proceedings of the 14th European Control Conference (ECC 2015)*, 25–30.

Boiroux, D., Bátora, V., Hagdrup, M., Tárnik, M., Murgaš, J., Schmidt, S., Nørgaard, K., Poulsen, N.K., Madsen, H., and Jørgensen, J.B. (2015). Comparison of prediction models for a dual-hormone artificial pancreas. In *9th IFAC Symposium on Biological and Medical Systems (BMS2015)*, 7 – 12.

Domahidi, A., Zgraggen, A.U., Zeilinger, M.N., Morari, M., and Jones, C.N. (2012). Efficient interior point methods for multistage problems arising in receding horizon control. *Proceedings of the IEEE Conference on Decision and Control, Including the Symposium on Adaptive Processes*, 668–674.

Frison, G. and Jørgensen, J.B. (2013). Efficient Implementation of the Riccati Recursion for Solving Linear-Quadratic Control Problems. *2013 IEEE Multiconference on Systems and Control*, 1117–1122.

Gertz, E. and Wright, S. (2003). Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29(1), 58–81.

Hagdrup, M., Boiroux, D., Mahmoudi, Z., Madsen, H., Kjølstad Poulsen, N., Poulsen, B., and Jørgensen, J.B. (2016). On the significance of the noise model for the performance of a linear MPC in closed-loop operation. *IFAC-papersonline*, 49(7), 171–176. doi:10.1016/j.ifacol.2016.07.241.

Jørgensen, J.B., Frison, G., Gade-Nielsen, N.F., and Dammann, B. (2012). Numerical Methods for Solution of the Extended Linear Quadratic Control Problem. *Nonlinear Model Predictive Control*, 4(1), 187–193.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4), 575–601.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization, 2nd Edition*. Springer, New York, USA.

Rao, C.V., Wright, S., and Rawlings, J.B. (1998). Application of Interior-Point Methods to Model Predictive Control. *Journal of Optimization Theory and Applications*, 99(3), 723 – 757.

Rawlings, J. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, USA.

Schmidt, S., Boiroux, D., Ranjan, A., Jørgensen, J.B., Madsen, H., and Nørgaard, K. (2015). An artificial pancreas for automated blood glucose control in patients with type 1 diabetes. *Therapeutic Delivery*, 6(5), 211–221. doi:10.4155/tde.15.12.

Sokoler, L.E., Frison, G., Skajaa, A., Halvgaard, R.F., and Jørgensen, J.B. (2015). A Homogeneous and Self-Dual Interior-Point Linear Programming Algorithm for Economic Model Predictive Control. *IEEE Transactions on Automatic Control*, PP(99), 2226–2231.

Wang, Y. and Boyd, S. (2010). Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.

Wright, S. (1997). *Primal-dual interior-point methods*. SIAM.

Zavitsanou, S., Chakrabarty, A., Dassau, E., and Doyle, F. (2016). Embedded control in wearable medical devices: Application to the artificial pancreas. *Processes*, 4(4), 35.