A Structure Exploiting Interior-Point Method for Moving Horizon Estimation

Niels Haverbeke, Moritz Diehl and Bart De Moor

***Abstract*— In this article a primal barrier interior-point method for moving horizon estimation (MHE) is presented. It exploits the structure of the KKT systems yielding an algorithm with linear complexity in the horizon length. Ideas of square root covariance Kalman filtering are proposed in order to efficiently update covariance matrices occurring in the factorization of the KKT matrix. The algorithm is able to compute - without any additional costs - the covariance of the last estimate within the horizon, which reflects the accuracy of the estimate.**

***Index Terms*— State estimation; Matrix Riccati equations; Optimal filtering; Optimization problems**

I. INTRODUCTION

The efficient solution of quadratic programs that arise in MPC has been addressed in [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. In [13] a primal barrier method for MPC problems was presented.

Almost all of these publications focus on the control problem, i.e. MPC. The state estimation problem, i.e. MHE, is closely related to the control problem and has largely the same optimal control structure. However, there are two main differences; first, the initial state is free in MHE and second, the Hessian of the objective function is almost always positive semidefinite as opposed to positive definite in standard MPC formulations.

It turns out that MHE problems can be solved efficiently by a matrix Riccati recursion, more precisely a form of the covariance Kalman filter recursion, a forward and a backward vector solve. A well known problem with the normal Kalman filter recursions is that they can result in a covariance matrix which fails to be symmetric [14], [15]. To cope with this difficulty Potter and Stern [16] introduced the idea of expressing the Kalman filter recursions in terms of a square-root, more precisely a Cholesky factor of the error covariance matrix. By propagating such a Cholesky factor, the computed error covariance matrix remains symmetric and positive (semi-)definite at all times. Furthermore, due to the numerically stable operations such as Householder reflections and Given rotations usually employed in square-root implementations, they are numerically better conditioned than a direct implementation[17].

In this article a primal barrier interior-point method for MHE is presented. The contribution of this article is three-fold: first, we show how constrained MHE problems can be efficiently factorized using an interior-point method with square-root recursions for KKT solution; second, it is shown how covariances of within-horizon states, which are obtained as a by-product, are naturally modified by the barrier; finally, we demonstrate with some numerical simulations that MHE problems can be solved extremely fast.

The article is organized as follows. In Section II, the state estimation problem is introduced. In Section III, we give an interior-point algorithm for solving the quadratic program formulated in Section II. The key goal of this paper, namely the structure exploiting algorithm for factorizing and solving the underlying KKT system is outlined in Section IV and subsequently in Section V a numerical example is given to demonstrate the efficiency of the proposed algorithms.

Corresponding author Niels Haverbeke Tel. +32 16 328652. niels.haverbeke@esat.kuleuven.be

Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Heverlee (Leuven), Belgium.

II. BACKGROUND

A. Problem statement

In this section we describe the state estimation problem for linear Gauss-Markov models. Let the linear time-varying dynamical system be given by

$$x_{k+1} = f_k + A_k x_k + G_k w_k, \quad k = 0, 1, \dots \tag{1}$$

$$y_k = C_k x_k + H_k w_k + v_k, \quad k = 0, 1, \dots \tag{2}$$

where k denotes discrete time, $x_k \in \mathbf{R}^n$ is the state, $w_k \in \mathbf{R}^n$ is the state disturbance or process noise, $v_k \in \mathbf{R}^p$ is the output disturbance or sensor noise, and $y_k \in \mathbf{R}^p$ is the observed output. The system matrices $A_k \in \mathbf{R}^{n\times n}$, $G_k \in \mathbf{R}^{n\times m}$, $H_k \in \mathbf{R}^{p\times m}$ and $C_k \in \mathbf{R}^{p\times n}$ and the offsets $f_k \in \mathbf{R}^n$ are assumed to be known. The goal is to find the state sequence that is most likely, given the sequence of observations and the model described above.

Remark 1: The model described above is slightly more general than other linear (time-varying) models encountered in the MHE literature. This model arises from linearization of nonlinear MHE problems within the framework of Newton-type methods. Our interest in this article is in the underlying structure of the quadratic (sub)problems, hence, we will accept the time-varying model without worrying about its origin.

The objective is defined as

$$\mathcal{J}(N, \mathbf{x}, \mathbf{w}) = \mathcal{J}_{\text{ic}}(x_0) + \mathcal{J}_{\text{proc}}(N, \mathbf{w}) + \mathcal{J}_{\text{sens}}(N, \mathbf{x}, \mathbf{w}),$$

where N is the estimation horizon and where $\mathbf{x} \in \mathbf{R}^{(N+1).n}$ and $\mathbf{w} \in \mathbf{R}^{N.m}$ denote, respectively, the stacked vectors of states and disturbances. The first term, $\mathcal{J}_{\text{ic}}$ is a penalty on the initial condition. Usually, it is assumed that some prior information is available in the form of an initial state estimate

$\bar{x}$ and corresponding covariance matrix $\bar{P}$, in which case we define

$$\mathcal{J}_{\mathrm{ic}}(x_0) = \|S_0^{-T}(x_0 - \bar{x})\|_2^2,$$

where the matrix S_0 is a square root of $\bar{P}$, i.e. $\bar{P} = S_0^T S_0$; a Cholesky factorization for instance generates a (lower) triangular S_0. The larger the covariance of the initial state error, the less weight it is given in the objective function.

The second term, $\mathcal{J}_{\mathrm{proc}}$, is a penalization of the state disturbances. We assume that w_k, for different values of k, are independently distributed, which implies

$$\mathcal{J}_{\mathrm{proc}}(N, \mathbf{w}) = \sum_{k=0}^{N-1} \|W_k^{-T} w_k\|_2^2.$$

Here, W_k is a weighting matrix, which in a stochastic setting can be regarded as the square root of the covariance matrix corresponding to the process noise w_k, i.e., $Q_k = W_k^T W_k$ with Q_k the covariance matrix.

The third term, $\mathcal{J}_{\mathrm{sens}}$, is a penalization of the sensor noise. Again, we assume v_k to be independently distributed, which implies

$$\mathcal{J}_{\mathrm{sens}}(N, \mathbf{x}, \mathbf{w}) = \sum_{k=0}^{N} \|V_k^{-T} v_k\|_2^2,$$

where V_k is also a weighting matrix, which in a stochastic setting can be regarded as the square root of the covariance matrix corresponding to v_k, i.e., $R_k = V_k^T V_k$ with R_k the covariance matrix. In this article we will assume both W_k and V_k to be identity matrices, without loss of generality.

Next, v_k can be eliminated using output equation (2), leading to

$$\mathcal{J}_{\mathrm{sens}}(N, \mathbf{x}, \mathbf{w}) = \sum_{k=0}^{N-1} \|y_k - C_k x_k - H_k w_k\|_2^2 + \|y_N - C_N x_N\|_2^2.$$

Finally, mixed linear inequality constraints are considered:

$$T_k^x x_k + T_k^w w_k \leq t_k, \quad k = 0, 1, \ldots, N-1, \tag{3}$$

$$T_N^x x_N \leq t_N, \tag{4}$$

where $T_k^x \in \mathbf{R}^{r_k \times n}$, $T_k^w \in \mathbf{R}^{r_k \times m}$, $t_k \in \mathbf{R}^{r_k}$, and $T_N^x \in \mathbf{R}^{r_N \times n}$, $t_N \in \mathbf{R}^{r_N}$ are given, and where $\leq$ denotes vector (componentwise) inequality.

Before continuing, let us summarize the state estimation problem treated in this paper:

$$\begin{array}{ll} \min_{\mathbf{x},\mathbf{w}} & \|S_0^{-T}(x_0 - \bar{x})\|_2^2 + \sum_{k=0}^{N-1} \|w_k\|_2^2 \\ & + \sum_{k=0}^{N-1} \|y_k - C_k x_k - H_k w_k\|_2^2 \\ & + \|y_N - C_N x_N\|_2^2 \\ \text{s.t.} & x_{k+1} = f_k + A_k x_k + G_k w_k, \quad k = 0, \ldots, N-1, \\ & T_k^x x_k + T_k^w w_k \leq t_k, \qquad\quad k = 0, \ldots, N-1, \\ & T_N^x x_N \leq t_N, \end{array} \tag{5}$$

This problem is a convex QP. In Sections III and IV we present a structure exploiting interior-point method for solving these quadratic programs efficiently.

B. Moving horizon estimation

The least squares batch estimator described above cannot be applied to online estimation problems in general because the problem grows unbounded with increasing horizon. In order to bound the problem, people have proposed a moving horizon strategy [18], [19] relying on the ideas of dynamic programming. The idea is to summarize past information by an initial state estimate and a corresponding weighting matrix. Evidently, one should use the best initial estimate and weighting available. From the previously computed optimal state trajectory, a good initial estimate is already available. The weighting matrix is propagated using a Kalman filter update (see e.g.,[20], [18]). The moving horizon problem is identical to (5) with growing horizon N until a horizon N_{hor} is reached after which MHE problems of length N_{hor} are being solved with propagated $\bar{x}, \bar{P}$.

III. PRIMAL BARRIER INTERIOR-POINT METHOD

In this section a basic primal barrier interior-point for solving (5) is described. Let us first define an overall optimization variable

$$z = (x_0, w_0, x_1 \ldots, w_{N-1}, x_N) \in \mathbf{R}^{(N+1)n+Nm},$$

Then the QP may be expressed as

$$\begin{array}{ll} \min_z & \frac{1}{2} z^T H z + g^T z \\ \text{s.t.} & Pz \leq h, \quad Cz = b, \end{array} \tag{6}$$

with

$$H = \begin{bmatrix} S_0^{-1} S_0^{-T} + C_0^T C_0 & C_0^T H_0 & & \\ H_0^T C_0 & I_m + H_0^T H_0 & & \\ & & \ddots & \\ & & & C_N^T C_N \end{bmatrix},$$

$$g = \begin{bmatrix} -S_0^{-1} S_0^{-T} \bar{x} + C_0^T h_0 \\ H_0^T h_0 \\ \vdots \\ H_{N-1}^T h_{N-1} \\ C_N^T h_N \end{bmatrix}, b = -\begin{bmatrix} f_0 \\ \vdots \\ f_{N-1} \end{bmatrix},$$

$$C = \begin{bmatrix} A_0 & G_0 & -I_n & & \\ & & \ddots & & \\ & & & A_{N-1} & G_{N-1} & -I_n \end{bmatrix},$$

$$P = \begin{bmatrix} T_0^x & T_0^w & & & \\ & \ddots & & & \\ & & T_{N-1}^x & T_{N-1}^w & \\ & & & & T_N^x \end{bmatrix}, h = \begin{bmatrix} t_0 \\ \vdots \\ t_{N-1} \\ t_N \end{bmatrix},$$

where I_n denotes the unit matrix of dimension n.

A. Primal barrier method

We will use a primal barrier method to solve the QP ([21, Chap. 11] [22]). The inequality constraints in the QP (6) are replaced with a barrier term in the objective, to get the approximate problem

$$\begin{array}{ll} \min_z & z^T H z + g^T z + \kappa \phi(z) \\ \text{s.t.} & Cz = b, \end{array} \tag{7}$$

where $\kappa > 0$ is a barrier parameter, and ϕ is the logarithmic barrier associated with the inequality constraints, defined as

$$\phi(z) = \sum_{i=1}^{N_p} -\log(h_i - p_i^T z),$$

where $p_1^T, \ldots, p_{N_p}^T$ are the rows of P. The problem (7) is a convex optimization problem with smooth objective and linear equality constraints, and can be solved by Newton's method.

In a basic primal barrier method, a sequence of problems of the form (7) is solved, using Newton's method starting from the previously computed point, for a decreasing sequence of values of κ. As κ approaches zero, the solution of (7) converges to a solution of the QP (6).

B. Newton method

We now focus on solving the problem (7) using a Newton method [21, §10.3.2]. Let us introduce a dual variable $\nu \in \mathbf{R}^{Nn}$ associated with the equality constraint $Cz = b$. The optimality conditions for (7) are then

$$\begin{aligned} r_d &= Hz + g + \kappa P^T d + C^T \nu = 0 \\ r_p &= Cz - b = 0 \end{aligned} \tag{8}$$

with r_d the dual and r_p the primal residual and where $d_i = 1/(h_i - p_i^T z)$, and p_i^T denotes the ith row of P. The term $\kappa P^T d$ is the gradient of $\kappa\phi(z)$. We also have the implicit constraint here that $Pz < h$.

The algorithm is initialized with a z^0 point that strictly satisfies the inequality constraints but need not satisfy the equality constraints. An arbitrary initial value can be used for ν^0.

We maintain an approximate z (with $Pz < h$) and ν at each step. If the residuals are small enough, we quit; otherwise we refine our estimate by linearizing the optimality conditions (8) and computing primal and dual steps Δz, $\Delta\nu$ for which $z + \Delta z$, $\nu + \Delta\nu$ give zero residuals in the linearized approximation.

The primal and dual search steps Δz and $\Delta\nu$ are found by solving a system of linear equations:

$$\begin{bmatrix} H + \kappa P^T \mathbf{diag}(d)^2 P & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix}. \tag{9}$$

This system of equations is called the KKT system and the matrix in the left-hand side is called the KKT matrix. The term $\kappa P^T \mathbf{diag}(d)^2 P$ is the Hessian of the barrier $\kappa\phi(z)$ and has block diagonal structure in case of the mixed linear inequality constraints (3)-(4). Let us define

$$\begin{aligned} M_k &= \mathbf{diag}(1/(t_k - T_k^x x_k - T_k^w w_k)) T_k^x, \quad 0 \le k \le N-1 \\ L_k &= \mathbf{diag}(1/(t_k - T_k^x x_k - T_k^w w_k)) T_k^w, \quad 0 \le k \le N-1 \\ M_N &= \mathbf{diag}(1/(t_N - T_N^x x_N)) T_N^x, \end{aligned}$$

then we obtain

$$P^T \mathbf{diag}(d)^2 P = \begin{bmatrix} M_0^T M_0 & M_0^T L_0 & & \\ L_0^T M_0 & L_0^T L_0 & & \\ & & \ddots & \\ & & & M_N^T M_N \end{bmatrix}.$$

In the next section a method is presented to compute the Newton step Δz and $\Delta\nu$ efficiently. Having obtained this Newton step, we perform a feasibility search and a backtracking line search on the norm of the residual r (see, e.g., [21, §9.2]). Finally, the primal and dual variables are updated: $z := z + s\Delta z$ and $\nu := \nu + s\Delta\nu$. This procedure is repeated until the norm of the residual is below an acceptable threshold.

IV. COMPUTING THE NEWTON STEP

A. Factorization

By rearranging the KKT system (9), the block diagonal structure of the KKT matrix is revealed. Let us write the (rearranged) KKT system as $M\xi = r$, with

$$M = \begin{bmatrix} \Phi_0 & \Gamma_0^T & 0 & & & & \\ \Gamma_0 & 0 & \Upsilon^T & & & & \\ 0 & \Upsilon & \Phi_1 & & & & \\ & & & \ddots & & & \\ & & & & \Phi_{N-1} & \Gamma_{N-1}^T & 0 \\ & & & & \Gamma_{N-1} & 0 & \Upsilon^T \\ & & & & 0 & \Upsilon & \Phi_N \end{bmatrix},$$

$$\xi = \begin{bmatrix} \Delta x_0 \\ \Delta w_0 \\ \Delta\nu_1 \\ \vdots \\ \Delta\nu_N \\ \Delta x_N \end{bmatrix}, \quad r = \begin{bmatrix} r_d^{x_0} \\ r_d^{w_0} \\ r_p^{\nu_1} \\ \vdots \\ r_p^{\nu_N} \\ r_d^{x_N} \end{bmatrix},$$

where $r_d^{x_0}$ denotes the dual residual associated with x_0, and $r_p^{\nu_1}$ denotes the primal residual associated with ν_1 and where we defined

$$\Phi_0 = \begin{bmatrix} S_0^{-1} S_0^{-T} + C_0^T C_0 + \kappa M_0^T M_0 & C_0^T H_0 + \kappa M_0^T L_0 \\ H_0^T C_0 + \kappa L_0^T M_0 & I_m + H_0^T H_0 + \kappa L_0^T L_0 \end{bmatrix},$$

$$\Phi_k = \begin{bmatrix} C_k^T C_k + \kappa M_k^T M_k & C_k^T H_k + \kappa M_k^T L_k \\ H_k^T C_k + \kappa L_k^T M_k & I_m + H_k^T H_k + \kappa L_k^T L_k \end{bmatrix}, \quad 1 \le k \le N-1$$

$$\Phi_N = \begin{bmatrix} C_N^T C_N + \kappa M_N^T M_N \end{bmatrix},$$

$$\Gamma_k = \begin{bmatrix} A_k & G_k \end{bmatrix}, \quad 0 \le k \le N-1$$

$$\Upsilon = \begin{bmatrix} -I_n \\ 0 \end{bmatrix}.$$

Lemma 4.1: The KKT matrix is symmetric indefinite and can be factorized by an indefinite Cholesky decomposition $M = LDL^T$ with $D = \mathbf{blkdiag}(I_{n+m}, -I_n, \ldots, I_{n+m}, -I_n, I_n)$ and

$$L = \begin{bmatrix} S_{0+}^{-1} & & & & & & \\ \Gamma_0 S_{0+}^T & S_1^T & & & & & \\ 0 & -\Upsilon S_1^{-1} & S_{1+}^{-1} & & & & \\ & & & \ddots & & & \\ & & & & S_{N-1+}^{-1} & & \\ & & & & \Gamma_{N-1} S_{N-1+}^T & S_N^T & \\ & & & & 0 & -\Upsilon S_N^{-1} & S_{N+}^{-1} \end{bmatrix}$$

which can be recursively computed by Algorithm 1.

Algorithm 1: [Riccati recursion]

1) *Initialization:* $\hat{R}_0 = S_0$
2) *For* $k = 0, \ldots, N-1$*:*

   (a) *Measurement update step*

   Let $D_k = \begin{bmatrix} C_k & H_k \\ \sqrt{\kappa}M_k & \sqrt{\kappa}L_k \end{bmatrix}$

   Compute QR-factorization:

$$\begin{bmatrix} D_k \begin{bmatrix} \hat{R}_k^T & \\ & I_m \end{bmatrix} \\ I_{n+m} \end{bmatrix} = \begin{bmatrix} Q_k & \tilde{Q}_k \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}$$

   Set $S_{k+} = R_k^{-T} \begin{bmatrix} \hat{R}_k & \\ & I_m \end{bmatrix}$

   (b) *Model forwarding step*

   Let $\Gamma_k = \begin{bmatrix} A_k & G_k \end{bmatrix}$

   Compute QR-factorization:

$$S_{k+}\Gamma_k^T = \begin{bmatrix} \hat{Q}_{k+1} & \bar{Q}_{k+1} \end{bmatrix} \begin{bmatrix} \hat{R}_{k+1} \\ 0 \end{bmatrix}$$

   Set $S_{k+1} = \hat{R}_{k+1}$

   endfor.

3) *Final time step:*

   (a) *Measurement update step*

   Let $D_N = \begin{bmatrix} C_N \\ \sqrt{\kappa}M_N \end{bmatrix}$

   Compute QR-factorization:

$$\begin{bmatrix} D_N \hat{R}_N^T \\ I_n \end{bmatrix} = \begin{bmatrix} Q_N & \tilde{Q}_N \end{bmatrix} \begin{bmatrix} R_N \\ 0 \end{bmatrix}$$

   Set $S_{N+} = R_N^{-T} \hat{R}_N$

A proof of Lemma 4.1 is now given.

Proof: To find out how S_{0+}, S_1, S_{1+} etc. can be computed, we multiply out LDL^T and equate with M. Then,

$$\begin{aligned} S_{0+}^{-1}S_{0+}^{-T} &= \Phi_0 && (10)\\ S_1^T S_1 &= \Gamma_0 S_{0+}^T S_{0+} \Gamma_0^T && (11)\\ &\vdots \\ S_{k+}^{-1}S_{k+}^{-T} &= \Upsilon S_k^{-1} S_k^{-T} \Upsilon^T + \Phi_k && (12)\\ S_{k+1}^T S_{k+1} &= \Gamma_k S_{k+}^T S_{k+} \Gamma_k^T && (13)\\ &\vdots \\ S_{N+}^{-1}S_{N+}^{-T} &= \Upsilon S_N^{-1} S_N^{-T} \Upsilon^T + \Phi_N && (14) \end{aligned}$$

First we prove the measurement update step, i.e. Equations (10), (12) and (14).

Let us define $\hat{R}_k = S_k$ and $D_k = \begin{bmatrix} C_k & H_k \\ \sqrt{\kappa}M_k & \sqrt{\kappa}L_k \end{bmatrix}$. Note that

$$\begin{aligned} \Phi_0 &= \begin{bmatrix} S_0^{-1}S_0^{-T} & \\ & I_m \end{bmatrix} + D_0^T D_0, \\ \Phi_k &= \begin{bmatrix} 0 & \\ & I_m \end{bmatrix} + D_k^T D_k, \quad 1 \le k \le N-1 \\ \Phi_N &= D_N^T D_N, \end{aligned} \qquad (15)$$

Let us compute the QR-factorization

$$\begin{bmatrix} D_k \begin{bmatrix} \hat{R}_k^T & \\ & I_m \end{bmatrix} \\ I_{n+m} \end{bmatrix} = \begin{bmatrix} Q_k & \tilde{Q}_k \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix} \qquad (16)$$

where Q_k and $\tilde{Q}_k$ are orthogonal matrices and R_k is upper triangular. Then we can set

$$S_{k+} = R_k^{-T} \begin{bmatrix} \hat{R}_k & \\ & I_m \end{bmatrix} \qquad (17)$$

To see that this is a valid choice for S_{k+}, we invert both sides, assuming without loss of generality that S_{k+} is invertible, and right-multiply with Q_k^T:

$$\begin{aligned} S_{k+}^{-1}Q_k^T &= \begin{bmatrix} \hat{R}_k^{-1} & \\ & I_m \end{bmatrix} R_k^T Q_k^T \\ &= \begin{bmatrix} \hat{R}_k^{-1} & \\ & I_m \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \hat{R}_k & \\ & I_m \end{bmatrix} D_k^T \\ I_{n+m} \end{bmatrix} \\ &= \begin{bmatrix} D_k^T \\ \begin{bmatrix} \hat{R}_k^{-1} & \\ & I_m \end{bmatrix} \end{bmatrix} \end{aligned}$$

where Eq. (16) was used in the second line. Now, since Q_k is orthogonal, we can write

$$\begin{aligned} S_{k+}^{-1}Q_k^T Q_k S_{k+}^{-T} &= S_{k+}^{-1}S_{k+}^{-T} \\ &= \begin{bmatrix} D_k^T \\ \begin{bmatrix} \hat{R}_k^{-1} & \\ & I_m \end{bmatrix} \end{bmatrix} \begin{bmatrix} D_k & \begin{bmatrix} \hat{R}_k^{-T} & \\ & I_m \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} \hat{R}_k^{-1}\hat{R}_k^{-T} & 0 \\ 0 & I_m \end{bmatrix} + D_k^T D_k \end{aligned}$$

which, after substitution of $\hat{R}_k$ and using (15), concludes the proof for Equations (10) and (12). For Eq. (14) the reasoning is similar.

Next, we prove the model forwarding step. From Eq. (13) it can be seen that with any orthogonal matrix $\hat{Q}_{k+1}$

$$S_{k+1} = \hat{Q}_{k+1}^T S_{k+} \Gamma_k^T, \quad 0 \le k \le N-1 \qquad (18)$$

We compute a QR-factorization

$$S_{k+}\Gamma_k^T = \begin{bmatrix} \hat{Q}_{k+1} & \bar{Q}_{k+1} \end{bmatrix} \begin{bmatrix} \hat{R}_{k+1} \\ 0 \end{bmatrix} \qquad (19)$$

where $\hat{Q}_{k+1}$ and $\bar{Q}_{k+1}$ are orthogonal matrices and $\hat{R}_{k+1}$ is upper triangular. ■

By applying this factorization to an arbitrary right hand side the KKT optimal vector is obtained after a forward and a backward vector solve. However, the residual vector $r_d^{x_0}$ contains a term $S_0^{-1}S_0^{-T}(x_0 - \bar{x})$. Hence, if we would solve $L\xi' = r$ and subsequently solve $DL^T\xi = \xi'$, then we would come across a term $S_0^{-T}(x_0 - \bar{x})$ after the forward vector solve, and invertibility of S_0 would be required. Fortunately, there is an elegant solution to this. A block LU decomposition of the KKT matrix would lead to the classical Kalman filter equations. By applying it to the residual vector and after some matrix

manipulations the invertibility assumption can be removed. Next, the recursion can be reformulated to use the square-root factors computed by Algorithm 1. The forward and backward vector recursions are described in Algorithm 2 and Algorithm 3. Proofs are omitted due to space limitations.

Algorithm 2: [Forward vector recursion]

1) *Initialization:*

$$\begin{aligned} d_0 &= \begin{bmatrix} \bar{x} - x_0 \\ -w_0 \end{bmatrix} \\ c_0 &= S_{0+}^T S_{0+} \Gamma_0^T \nu_1 \end{aligned}$$

2) *For* $k = 0, \ldots, N-1$*:*

$$\begin{aligned} g_k &= \begin{bmatrix} h_k + C_k x_k + H_k w_k \\ \sqrt{\kappa} e_r \end{bmatrix} \\ \begin{bmatrix} \Delta x_k^{'} \\ \Delta w_k^{'} \end{bmatrix} &= d_k - S_{k+}^T Q_k^T \begin{bmatrix} D_k d_k + g_k \\ 0 \end{bmatrix} - c_k \\ d_{k+1} &= \begin{bmatrix} \Gamma_k \begin{bmatrix} \Delta x_k^{'} + x_k \\ \Delta w_k^{'} + w_k \end{bmatrix} + f_k - x_{k+1} \\ -w_{k+1} \end{bmatrix} \end{aligned}$$

If $(k = N-1)$ then

$$c_N = -S_{N+}^T S_{N+} \nu_N$$

else

$$c_{k+1} = S_{k+1+}^T S_{k+1+} (\Gamma_{k+1}^T \nu_{k+2} - \begin{bmatrix} \nu_{k+1} \\ 0 \end{bmatrix})$$

endif.
endfor.

3) *Final time step:*

$$\begin{aligned} g_N &= \begin{bmatrix} h_N + C_N x_N \\ \sqrt{\kappa} e_s \end{bmatrix} \\ \Delta x_N^{'} &= d_N - S_{N+}^T Q_N^T \begin{bmatrix} D_N d_N + g_N \\ 0 \end{bmatrix} - c_N \end{aligned}$$

Algorithm 3: [Backward vector recursion]

1) *Initialization:* $\Delta x_N = \Delta x_N^{'}$
2) *For* $k = N-1, \ldots, 0$*:*
If $(k = N-1)$ then

$$\Delta \nu_N = r_d^{x_N} - \Upsilon^T \Phi_N \Delta x_N$$

else

$$\begin{aligned} \Delta \nu_{k+1} &= r_d^{x_{k+1}} \\ &\quad - \Upsilon^T (\Phi_{k+1} \begin{bmatrix} \Delta x_{k+1} \\ \Delta w_{k+1} \end{bmatrix} + \Gamma_{k+1}^T \Delta \nu_{k+2}) \end{aligned}$$

endif.

$$\begin{bmatrix} \Delta x_k \\ \Delta w_k \end{bmatrix} = \begin{bmatrix} \Delta x_k^{'} \\ \Delta w_k^{'} \end{bmatrix} - S_{k+}^T \hat{Q}_{k+1} \hat{R}_{k+1} \Delta \nu_{k+1}$$

endfor.

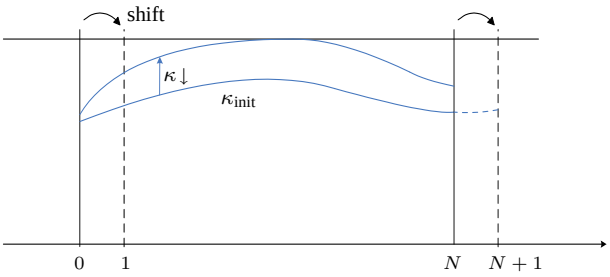

Fig. 1. Illustration of the hot starting procedure. Solve a sequence of problems with decreasing κ until reasonable accuracy, shift the solution of the problem with $\kappa = \kappa_{\text{init}}$ to the next time step.

B. Hot starting

In MHE, similar optimal control problems are solved successively. Therefore, it is a reasonable assumption that the solution of an MHE problem can be shifted one time step forward to yield a good starting point for the next MHE problem. Unfortunately, as pointed out in [1], in interior-point methods it is better to use a starting point away from the boundary (a strictly feasible point). More specifically we recall that in interior-point methods a sequence of problems of the form (7) is solved for decreasing κ, however if we shift from one MHE problem to the next, we start a new optimization problem, i.e. we start with $\kappa = \kappa_{\text{init}}$. Hence it turns out that the solution to the previous MHE problem (i.e. with small κ) is not a good initialization. The procedure we suggest here, illustrated in Figure 1, is to solve a sequence of problems for decreasing κ until reasonable accuracy is obtained, but shift the solution of the problem with $\kappa = \kappa_{\text{init}}$ to the next time step as a good starting point.

V. EXAMPLES

The algorithm was implemented in C with calls to BLAS and Lapack libraries [23]: *dgeqrf*, the QR factorization algorithm from LAPACK using Householder reflections, was used; BLAS routines *dgemm* and *dgemv* were used for matrix-matrix and matrix-vector multiplications respectively. Our platform is a 2Ghz AMD Athlon running Linux and the computation times were obtained with the Unix command gettimeofday.

A. Constrained linear system

Consider the following linear system [19]

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.99 & 0.2 \\ -0.1 & 0.3 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k, \quad k = 0, 1, \ldots \\ y_k &= \begin{bmatrix} 1 & -3 \end{bmatrix} x_k + v_k, \qquad\qquad k = 0, 1, \ldots \end{aligned}$$

We assume v_k to be zero-mean normally distributed random noise with variance 0.01, and $w_k = |z_k|$ with z_k zero-mean normally distributed random noise with unit variance. We also assume x_0 to be normally distributed with zero mean and unit covariance.

We formulate the constrained estimation problem with $Q = 1$, $R = 0.01$, $P_0 = I_2$, and $\bar{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

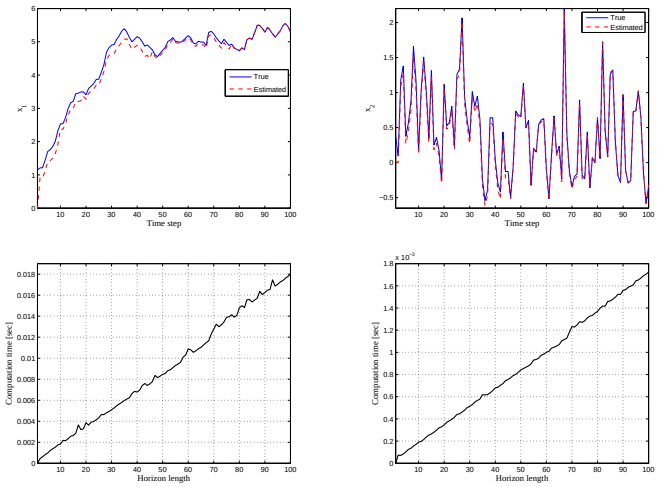

Fig. 2. Numerical example for estimation with growing horizon and a fixed number of iterations (ten). **Top left:** True and estimated first states. **Top right:** True and estimated second states. **Bottom left:** Computation times. **Bottom left:** Computation times for single iteration (for comparison).

The results are shown in Figure 2. The computation times are in the order of milliseconds. As an example, a constrained MHE problem of horizon 40 can be solved in around 5msec for this two-state problem.

VI. CONCLUSIONS AND FUTURE WORK

In this paper a structure exploiting interior-point method for moving horizon estimation is presented. The crucial step in the interior-point method, the computation of the Newton direction, can be done efficiently and robustly with a Riccati recursion using square root factors. Covariances of all within-horizon states are naturally modified with the logarithmic barrier term and are obtained as a by-product of the factorization. A numerical example demonstrates that computation times in the range of milliseconds are feasible. Future research is directed towards embedding the methods proposed in this paper within an SQP framework for nonlinear moving horizon estimation.

VII. ACKNOWLEDGMENTS

Niels Haverbeke is a research assistant at Katholieke Universiteit Leuven, Belgium, supported by IWT-Vlaanderen. Moritz Diehl is an associate professor and Bart De Moor is a full professor at Katholieke Universiteit Leuven. Research supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, G.0320.08, G.0558.08, G.0557.08, G.0588.09 research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, Belgian Federal Science Policy Office: IUAP P6/04; EU: ERNSI; FP7-HD-MPC. Contract Research: AMINAL. Other:Helmholtz: viCERP.

REFERENCES

[1] C. Rao, S. Wright, and J. Rawlings, "Application of Interior-Point Methods to Model Predictive Control," *Journal of Optimization Theory and Applications*, vol. 99, pp. 723–757, 1998.

[2] J. Jorgensen, J. Rawlings, and S. Jorgensen, "Numerical methods for large-scale moving horizon estimation and control," in *Proceedings of Int. Symposium on Dynamics and Control Process Systems (DYCOPS)*, 2004.

[3] M. Tenny and J. Rawlings, "Efficient Moving Horizon Estimation and Nonlinear Model Predictive Control," in *Proceedings of the American Control Conference*, 2002.

[4] L. Vandenberghe, S. Boyd, and M. Nouralishahi, "Robust Linear Programming and Optimal Control," in *Proceedings of IFAC World Congress*, 2002, pp. 271 – 276.

[5] L. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear Predictive Control*, ser. Progress in Systems Theory, F. Allgöwer and A. Zheng, Eds., vol. 26. Basel Boston Berlin: Birkhäuser, 2000, pp. 219–244.

[6] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," in *Proceedings of NMPC 08 Workshop Pavia*, 2008.

[7] A. Hansson and S. Boyd, "Robust optimal control of linear discrete time systems using primal-dual interior-point methods," in *Proceedings of the American Control Conference*, Jan. 1998.

[8] A. Hansson, "A primal-dual interior-point method for robust optimal control of linear discrete-time systems," *IEEE Transactions on Automatic Control*, Jan. 2000.

[9] P. Goulart, E. Kerrigan, and D. Ralph, "Efficient robust optimization for robust control with constraints," *Mathematical Programming*, vol. 114, pp. 115–147, 2008.

[10] R. Bartlett, L. Biegler, J. Backstrom, and V. Gopal, "Quadratic programming algorithms for large-scale model predictive control," *Journal of Process Control*, vol. 12, pp. 775–795, 2002.

[11] T. Glad and H. Jonson, "A method for state and control constrained linear quadratic control problems," in *Proceedings of IFAC World Congress*, 1984, pp. 1583 – 1587.

[12] M. Steinbach, "A structured interior point sqp method for nonlinear optimal control problems," in *Computational Optimal Control, volume 115 of Int. Series Numer. Math*. Verlag, 1994, pp. 213–222.

[13] Y. Wang and S. Boyd, "Fast Model Predictive Control Using Online Optimization," in *Proceedings of IFAC World Congress*, 2008, pp. 6974 – 6997.

[14] B. D. O. Anderson and J. B. Moore, *Optimal Filtering (Dover Books on Engineering)*. Dover Publications, January 2005.

[15] S. Gillijns, "Kalman filtering techniques for system inversion and data assimilation," Ph.D. dissertation, Katholieke Universiteit Leuven, 2007.

[16] J. Potter and R. Stern, "Statistical filtering of space navigation measurements," in *Proceedings of AIAA Guidance and Control Conference*, 1963.

[17] M. Verhaegen and P. Van Dooren, "Numerical Aspects of Different Kalman Filter Implementations," *IEEE Transactions on Automatic Control*, vol. 31, pp. 907 – 917, 1986.

[18] C. Rao, J. Rawlings, and J. Lee, "Constrained linear state estimation - a moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, February 2001.

[19] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, pp. 246–258, 2003.

[20] D. Robertson, J. Lee, and J. Rawlings, "A moving horizon-based approach for least-squares estimation," *AIChE*, vol. 42, pp. 2209–2224, 1996.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[23] E. Anderson, A. McKenney, D. Sorensen, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. D. Croz, S. Hammarling, *et al.*, *LAPACK Users' guide*. SIAM, 1999.