

A Homogeneous and Self-Dual Interior-Point Linear Programming Algorithm for Economic Model Predictive Control

Leo Emil Sokoler, Gianluca Frison, Anders Skajaa, Rasmus Halvgaard, John Bagterp Jørgensen

Abstract—We develop an efficient homogeneous and self-dual interior-point method (IPM) for the linear programs arising in economic model predictive control (MPC) of constrained linear systems with linear objective functions. The algorithm is based on a Riccati iteration procedure, which is adapted to the linear system of equations solved in homogeneous and self-dual IPMs. Fast convergence is further achieved by means of a recent warm-start strategy for homogeneous and self-dual IPMs. We implement the algorithm in MATLAB and C. Its performance is tested using a conceptual power management case study. Closed loop simulations show that 1) the proposed algorithm is significantly faster than several state-of-the-art IPMs based on sparse linear algebra, and 2) warm-start reduces the average number of iterations by 35-40%.

Index Terms—Optimization algorithms, Linear programming algorithms, Predictive control for linear systems, Riccati iterations, Energy systems

I. INTRODUCTION

In economic MPC of linear systems with a linear objective function and linear constraints, the constrained optimal control problem can be posed as a linear program (LP). As the optimization problem is solved online, the performance and reliability of the optimization algorithm solving the LP is important. In this paper, we develop a homogeneous and self-dual variant of Mehrotra’s predictor-corrector method [1], [2] for economic MPC that combines the following performance improving components:

- **Riccati Iteration Procedure:** We speed-up the most time consuming numerical operations using a Riccati iteration procedure.
- **Warm-Start:** We implement a warm-start strategy for the homogeneous and self-dual IPMs. [3] reports that this method reduces the number of iterations by 30-75% based on the NETLIB collection of test problems.

Riccati-based IPMs have been developed for set-point based MPC with an ℓ_2 -penalty [4]–[6] and with an ℓ_1 -penalty [7]. A Riccati iteration procedure has not previously been combined with the homogeneous and self-dual model. The homogeneous and self-dual model has become widely adopted by state-of-the-art IPMs for linear and conic programming. This paper is organized as follows. Section II formulates the control law associated with economic MPC as the solution to a highly

structured LP. Section III introduces the homogeneous and self-dual model. This section also provides a Riccati iteration procedure for solving the structured linear system that occurs in every iteration of an IPM for solving the LP. Section IV discusses warm-start. Section V compares a MATLAB and C implementation of the proposed algorithm denoted LP_{empc} to several state-of-the-art IPMs using a simple power management case study¹. Concluding remarks are given in Section VI.

II. OPTIMAL CONTROL PROBLEM

We consider linear state space systems in the form

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + E\mathbf{d}_k, \quad \mathbf{d}_k \sim N(0, R_d), \quad (1a)$$

$$\mathbf{y}_k = C_y\mathbf{x}_k + \mathbf{e}_k, \quad \mathbf{e}_k \sim N(0, R_e), \quad (1b)$$

$$\mathbf{z}_k = C_z\mathbf{x}_k, \quad (1c)$$

where $\mathbf{x}_0 \sim N(\hat{\mathbf{x}}_0, P_0)$. (A, B, C_y, C_z, E) are the state space matrices, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the measurement vector, $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is the output vector, \mathbf{d}_k is the process noise vector and \mathbf{e}_k is the measurement noise vector. We use bold letters to denote stochastic variables.

In this paper, the Economic MPC optimal control problem is defined to optimize the control actions of (1) with respect to a linear economic objective function, input limits, input-rate limits and soft output limits. Evaluation of this control law requires the solution to the LP

$$\min_{\mathbf{u}, \hat{\mathbf{x}}, \hat{\mathbf{z}}, \rho} \sum_{j \in \mathcal{N}_0} p_{k+j}^T \mathbf{u}_{k+j} + q_{k+j+1}^T \rho_{k+j+1}, \quad (2a)$$

s.t.

$$\hat{\mathbf{x}}_{k+j+1|k} = A\hat{\mathbf{x}}_{k+j|k} + B\mathbf{u}_{k+j}, \quad j \in \mathcal{N}_0, \quad (2b)$$

$$\hat{\mathbf{z}}_{k+j|k} = C_z\hat{\mathbf{x}}_{k+j|k}, \quad j \in \mathcal{N}_1, \quad (2c)$$

$$\underline{\mathbf{u}}_{k+j} \leq \mathbf{u}_{k+j} \leq \bar{\mathbf{u}}_{k+j}, \quad j \in \mathcal{N}_0, \quad (2d)$$

$$\Delta \underline{\mathbf{u}}_{k+j} \leq \mathbf{u}_{k+j} - \mathbf{u}_{k+j-1} \leq \Delta \bar{\mathbf{u}}_{k+j}, \quad j \in \mathcal{N}_0, \quad (2e)$$

$$\underline{\mathbf{z}}_{k+j} - \rho_{k+j} \leq \hat{\mathbf{z}}_{k+j|k} \leq \bar{\mathbf{z}}_{k+j} + \rho_{k+j}, \quad j \in \mathcal{N}_1, \quad (2f)$$

$$\rho_{k+j} \geq 0, \quad j \in \mathcal{N}_1, \quad (2g)$$

where $\mathcal{N}_i := \{0 + i, 1 + i, \dots, N - 1 + i\}$, with N being the length of the prediction and control horizon. The problem data are the state-space matrices, (A, B, C_z) , the filtered

The authors are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark. L. E. Sokoler and A. Skajaa are also affiliated with DONG Energy, DK-2820 Gentofte, Denmark (e-mail: {leso, giap, andsk, rhal, jbjjo}@dtu.dk)

¹Software and implementation details are available via <http://www2.imm.dtu.dk/~jbjjo/publications>

estimate, $\hat{x}_{k|k}$, the input limits, $(\underline{u}_{k+j}, \bar{u}_{k+j})$, the input-rate limits, $(\Delta \underline{u}_{k+j}, \Delta \bar{u}_{k+j})$, the output limits, $(\underline{z}_{k+j}, \bar{z}_{k+j})$, the input prices, p_{k+j} , and the price for violating the output constraints, q_{k+j} . Note that for compact notation, the optimization variables in (2) are written as u, \hat{x}, \hat{z} and ρ , where $u = [u_k^T \ u_{k+1}^T \ \dots \ u_{k+N-1}^T]^T$, and similarly for \hat{x}, \hat{z} , and ρ . The filtered estimate, $\hat{x}_{k|k} := E[x_k|Y_k]$, is the conditional expectation of x_k given the observations $Y_k := [y_0^T \ y_1^T \ y_2^T \ \dots \ y_k^T]^T$. We obtain this value using the Kalman filter. By augmenting the state-space system such that

$$A := \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{x}_k := \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix}, \quad B := \begin{bmatrix} B \\ I \end{bmatrix},$$

$$E := \begin{bmatrix} E \\ 0 \end{bmatrix}, \quad C_z := [C_z \ 0], \quad C_y := [C_y \ 0],$$

we can express (2e) as

$$\Delta \underline{u}_{k+j} \leq u_{k+j} - D x_k \leq \Delta \bar{u}_{k+j}, \quad j \in \mathcal{N}_0,$$

in which $D := [0 \ I]$. This formulation simplifies later computations considerably. To keep the notation simple we assume that $k = 0$ and write $\hat{x}_j := \hat{x}_{0+j|0}$ for conditional expressions. The problem data is collected in the structures g, F, b, H and c , and (2) is put into the form

$$\min_{t,s} \{g^T t | Ft = b, Ht + s = c, s \geq 0\}. \quad (3)$$

As an example, consider the case for $N = 2$

$$t := [u_0^T \ \hat{x}_1^T \ \rho_1^T \ u_1^T \ \hat{x}_2^T \ \rho_2^T]^T,$$

$$g := [p_0^T \ 0 \ q_1^T \ p_1^T \ 0 \ q_2^T]^T,$$

and

$$\left[\begin{array}{cccccc|c} F & | & b \end{array} \right] := \left[\begin{array}{cccccc|c} B & -I & 0 & 0 & 0 & 0 & -A\hat{x}_0 \\ 0 & A & 0 & B & -I & 0 & 0 \\ \hline I & 0 & 0 & 0 & 0 & 0 & \bar{u}_0 \\ 0 & 0 & 0 & I & 0 & 0 & \bar{u}_1 \\ -I & 0 & 0 & 0 & 0 & 0 & -\underline{u}_0 \\ 0 & 0 & 0 & -I & 0 & 0 & -\underline{u}_1 \\ I & 0 & 0 & 0 & 0 & 0 & \Delta \bar{u}_0 \\ 0 & -D & 0 & I & 0 & 0 & \Delta \bar{u}_1 \\ \hline -I & 0 & 0 & 0 & 0 & 0 & -\Delta u_0 \\ 0 & D & 0 & -I & 0 & 0 & -\Delta u_1 \\ 0 & C_z & -I & 0 & 0 & 0 & \bar{z}_1 \\ 0 & 0 & 0 & 0 & C_z & -I & \bar{z}_2 \\ 0 & -C_z & -I & 0 & 0 & 0 & -\underline{z}_1 \\ 0 & 0 & 0 & 0 & -C_z & -I & -\underline{z}_2 \\ 0 & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I & 0 \end{array} \right],$$

where $\Delta \tilde{u}_0 := \Delta \bar{u}_0 + D\hat{x}_0$ and $\Delta u_0 := \Delta \underline{u}_0 + D\hat{x}_0$. The problem (2) can thus be posed as a highly structured LP with $n := N(n_u + n_x + n_z)$ variables, $m_E := Nn_x$ equality constraints, and $m_I := N(4n_u + 3n_z)$ inequality constraints. Note that we have eliminated \hat{z}_j from the optimization problem using the linear relation (2c).

III. HOMOGENEOUS AND SELF-DUAL INTERIOR-POINT METHOD

The dual of the LP (3) is

$$\max_{v,w} \{-b^T v - c^T w | -F^T v - H^T w = g, w \geq 0\}, \quad (4)$$

where $v \in \mathbb{R}^{m_E}$ and $w \in \mathbb{R}^{m_I}$ are dual variables corresponding to the Lagrange multipliers for the equality constraints and the inequality constraint of (3), respectively. We assume that F has full row rank. This is always the case for the problem (3).

Homogeneous and self-dual IPMs solve (3) and (4) indirectly. The idea is to construct a related LP with properties that are useful for IPMs. Aside from an inherent ability to detect infeasibility, recent advances show that IPMs based on the homogeneous and self-dual model can be warm-started efficiently [3]. We refer to [8]–[10] for proofs and details.

Introduce a new set of optimization variables $(\tilde{t}, \tilde{v}, \tilde{w}, \tilde{s})$, and the additional scalar variables $(\tilde{\tau}, \tilde{\kappa})$. Then the self-dual and homogeneous problem associated with (3) and (4) may be stated as the linear feasibility problem

$$\text{find } \tilde{t}, \tilde{v}, \tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa}, \quad (5a)$$

$$\text{s.t. } F^T \tilde{v} + H^T \tilde{w} + g\tilde{\tau} = 0, \quad (5b)$$

$$b\tilde{\tau} - F\tilde{t} = 0, \quad (5c)$$

$$c\tilde{\tau} - H\tilde{t} - \tilde{s} = 0, \quad (5d)$$

$$-g^T \tilde{t} - b^T \tilde{v} - c^T \tilde{w} + \tilde{\kappa} = 0, \quad (5e)$$

$$(\tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa}) \geq 0, \quad (5f)$$

Proposition 1 shows that the solution to (3) and (4) can be obtained by solving (5).

Proposition 1: The linear feasibility problem (5) always has a strict complimentary solution $(\tilde{t}^*, \tilde{v}^*, \tilde{w}^*, \tilde{s}^*, \tilde{\tau}^*, \tilde{\kappa}^*)$ satisfying $\tilde{s}_j^* \tilde{w}_j^* = 0$ for $j = 1, 2, \dots, m_I$ and $\tilde{\tau}^* \tilde{\kappa}^* = 0$. For such a solution, one of the following conditions hold

- 1) $\tilde{\tau}^* > 0, \tilde{\kappa}^* = 0$: The scaled solution $(t^*, v^*, w^*, s^*) = (\tilde{t}^*, \tilde{v}^*, \tilde{w}^*, \tilde{s}^*)/\tilde{\tau}^*$ is a primal-dual optimal solution to (3) and (4).
- 2) $\tilde{\tau}^* = 0, \tilde{\kappa}^* > 0$: The problem (3) is infeasible or unbounded; either $-b^T \tilde{v}^* - c^T \tilde{w}^* > 0$ (primal infeasible), or $g^T \tilde{t}^* < 0$ (dual infeasible).

Proof: See [10], [11]. ■

A. Riccati Iteration Procedure for Economic MPC

Let k denote a particular iteration number. The necessary and sufficient optimality conditions for (5) are $(\tilde{w}, \tilde{s}, \tilde{\kappa}, \tilde{\tau}) \geq 0$ and

$$\begin{bmatrix} F^T \tilde{v} + H^T \tilde{w} + g\tilde{\tau} \\ b\tilde{\tau} - F\tilde{t} \\ c\tilde{\tau} - H\tilde{t} - \tilde{s} \\ -g^T \tilde{t} - b^T \tilde{v} - c^T \tilde{w} + \tilde{\kappa} \\ \tilde{W} \tilde{S} \mathbf{1}_{m_I} \\ \tilde{\tau} \tilde{\kappa} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (6)$$

\tilde{W} is a diagonal matrix with the elements of \tilde{w} in its diagonal, and similarly for \tilde{S} . Moreover, $\mathbf{1}_{m_I}$ is the column vector of all ones of size m_I . The main computational bottleneck in

finding a point that satisfies the optimality conditions using a homogeneous and self-dual IPM is solving a linear system of equations (in each iteration of the IPM) [1], [2]. For an arbitrary right hand side, we can solve this linear system of equations as

$$F^T \Delta \tilde{v} + H^T \Delta \tilde{w} + g \Delta \tilde{\tau} = r_1, \quad (7a)$$

$$b \Delta \tilde{\tau} - F \Delta \tilde{t} = r_2, \quad (7b)$$

$$c \Delta \tilde{\tau} - H \Delta \tilde{t} - \Delta \tilde{s} = r_3, \quad (7c)$$

$$g^T \Delta \tilde{t} + b^T \Delta \tilde{v} + c^T \Delta \tilde{w} - \Delta \tilde{\kappa} = r_4, \quad (7d)$$

$$\tilde{W} \Delta \tilde{s} + \tilde{S} \Delta \tilde{w} = r_5, \quad (7e)$$

$$\tilde{\kappa} \Delta \tilde{\tau} + \tilde{\tau} \Delta \tilde{\kappa} = r_6. \quad (7f)$$

The system (7) is different from the system solved in standard IPMs. The Riccati iteration procedures proposed in [4]–[6] can therefore not be applied directly to solve the system (7). Proposition 2 shows that a solution to (7) can be computed by solving a reduced linear system of equations (8) and a number of computationally inexpensive operations. Proposition 3 shows that the reduced system can be solved using a Riccati iteration procedure that scales as $O(N(n_u + n_x + n_z)^3)$. The procedure eliminates variables from the reduced linear system (9), such that it can be expressed in a form (11) that often occurs in optimal control problems. Riccati iterations solve the resulting system efficiently by exploiting its structure [4]–[6]. The complexity of solving the system (7) directly using sparse linear algebra routines is linear to quadratic in N , while a general purpose solver using dense linear algebra scales cubically [12]. If the number of states, n_x , is large compared to the number of inputs, n_u , condensing methods are more efficient than Riccati-based solvers [5]. Note that it is convenient to have an algorithm that scales linearly in the prediction horizon, N , as stability of MPC schemes often may be achieved by selecting a sufficiently large value of N [13], [14].

Reference [15] provides details on a predictor-corrector IPM [1], [2], `LPempc`, that utilizes a C implementation (with calls to BLAS and LAPACK) of the numerical procedure presented in Proposition 2 and Proposition 3. To speed-up the numerical computations and reduce the storage requirements, `LPempc` handles operations involving the structured matrices F and H as specialized linear algebra routines.

Proposition 2: The solution to (7) can be obtained by solving

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & \tilde{W}^{-1} \tilde{S} \end{bmatrix} \begin{bmatrix} f_1 & h_1 \\ f_2 & h_2 \\ f_3 & h_3 \end{bmatrix} = \begin{bmatrix} r_1 & -g \\ r_2 & -b \\ r_3 & -c \end{bmatrix}, \quad (8)$$

and subsequent computation of

$$\Delta \tilde{\tau} = \frac{r_6 - \tilde{\tau}(g^T f_1 + b^T f_2 + c^T f_3)}{\tilde{\kappa} + \tilde{\tau}(g^T h_1 + b^T h_2 + c^T h_3)},$$

$$\Delta \tilde{t} = f_1 + h_1 \Delta \tilde{\tau},$$

$$\Delta \tilde{v} = f_2 + h_2 \Delta \tilde{\tau},$$

$$\Delta \tilde{w} = f_3 + h_3 \Delta \tilde{\tau},$$

$$\Delta \tilde{\kappa} = g^T \Delta \tilde{t} + b^T \Delta \tilde{v} + c^T \Delta \tilde{w} - r_4,$$

$$\Delta \tilde{s} = \tilde{W}^{-1}(r_C - \tilde{S} \Delta \tilde{w}),$$

where $r_3 := r_3 + \tilde{W}^{-1} r_5$ and $r_6 := r_6 + \tilde{\tau} r_4$.

Proof: See [11]. \blacksquare

Proposition 3: System (8) can be solved in $O(N(n_u + n_x + n_z)^3)$ operations using a Riccati iteration procedure.

Proof: For a single arbitrary right hand side, we may write the system (8) as

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & \tilde{W}^{-1} \tilde{S} \end{bmatrix} \begin{bmatrix} \Delta \tilde{t} \\ \Delta \tilde{v} \\ \Delta \tilde{w} \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ r_I \end{bmatrix}. \quad (9)$$

Denote the Lagrange multipliers associated with the inequality constraints (2d) and (2g) as $\eta, \delta, v, \omega, \gamma, \zeta$ and ξ where

$$\eta := [\eta_0^T \ \eta_1^T \ \dots \ \eta_{N-1}^T]^T,$$

and similarly for δ, v, γ, ζ and ξ . The multipliers (η, δ) are associated with the input limits (2d), (v, ω) are associated with the input-rate limits (2e), (γ, ζ) are associated with the output limits (2f), and ξ is associated with the non-negative constraints (2g). The system variables are written in the form

$$\Delta \tilde{t} = [\Delta u_0^T \ \Delta \hat{x}_1^T \ \Delta \rho_1^T \ \dots \ \Delta u_{N-1}^T \ \Delta \hat{x}_N^T \ \Delta \rho_N^T]^T,$$

$$\Delta \tilde{v} = [\Delta \tilde{v}_0^T \ \Delta \tilde{v}_1^T \ \dots \ \Delta \tilde{v}_{N-1}^T]^T,$$

$$\Delta \tilde{w} = [\Delta \eta^T \ \Delta \delta^T \ \Delta v^T \ \Delta \omega^T \ \Delta \gamma^T \ \Delta \zeta^T \ \Delta \xi^T]^T.$$

Accordingly, we partition the right hand side such that

$$r_D = [r_{u,0}^T \ r_{x,1}^T \ r_{w,1}^T \ \dots \ r_{u,N-1}^T \ r_{x,N}^T \ r_{w,N}^T]^T,$$

$$r_E = [R_{v,0}^T \ R_{v,1}^T \ \dots \ R_{v,N-1}^T]^T,$$

$$r_I = [r_\eta^T \ r_\delta^T \ r_v^T \ r_\omega^T \ r_\gamma^T \ r_\zeta^T \ r_\xi^T]^T,$$

and write the diagonal matrix $\tilde{W}^{-1} \tilde{S}$ in terms of diagonal submatrices

$$\tilde{W}^{-1} \tilde{S} = \mathbf{diag}(\Sigma_\eta^T, \Sigma_\delta^T, \Sigma_v^T, \Sigma_\omega^T, \Sigma_\gamma^T, \Sigma_\zeta^T, \Sigma_\xi^T).$$

The linear system of equations (9) can now be stated in the form

$$\Delta \eta_i - \Delta \delta_i + \Delta v_i - \Delta \omega_i + B^T \Delta \tilde{v}_i = r_{u,i}, \quad i \in \mathcal{N}_0,$$

$$-\Delta u_i + \Sigma_{\eta,i} \Delta \eta_i = r_{\eta,i}, \quad i \in \mathcal{N}_0,$$

$$\Delta u_i + \Sigma_{\delta,i} \Delta \delta_i = r_{\delta,i}, \quad i \in \mathcal{N}_0,$$

$$-\Delta u_i + D \Delta \hat{x}_i + \Sigma_{v,i} \Delta v_i = r_{v,i}, \quad i \in \tilde{\mathcal{N}}_0,$$

$$\Delta u_i - D \Delta \hat{x}_i + \Sigma_{\omega,i} \Delta \omega_i = r_{\omega,i}, \quad i \in \tilde{\mathcal{N}}_0,$$

$$\Delta \hat{x}_{i+1} - A \Delta \hat{x}_i - B \Delta u_i = R_{v,i}, \quad i \in \tilde{\mathcal{N}}_0,$$

$$\Delta \rho_i - C_z \Delta \hat{x}_i + \Sigma_{\gamma,i} \Delta \gamma_i = r_{\gamma,i}, \quad i \in \mathcal{N}_1,$$

$$\Delta \rho_i + C_z \Delta \hat{x}_i + \Sigma_{\zeta,i} \Delta \zeta_i = r_{\zeta,i}, \quad i \in \mathcal{N}_1,$$

$$\Delta \rho_i + \Sigma_{\xi,i} \Delta \xi_i = r_{\xi,i}, \quad i \in \mathcal{N}_1,$$

$$-\Delta \gamma_i - \Delta \zeta_i - \Delta \xi_i = r_{w,i}, \quad i \in \mathcal{N}_1,$$

$$-\Delta \tilde{v}_i + C_z^T (\Delta \gamma_{i+1} - \Delta \zeta_{i+1}) + A^T \Delta \tilde{v}_i + D^T (\omega_i - \Delta v_i) = r_{x,i}, \quad i \in \tilde{\mathcal{N}}_0,$$

with $\tilde{\mathcal{N}}_0 := \mathcal{N}_0 \setminus \{0\}$ and the special cases

$$-\Delta u_0 + \Sigma_{v,0} \Delta v_0 = r_{v,0},$$

$$\Delta u_0 + \Sigma_{\omega,0} \Delta \omega_0 = r_{\omega,0},$$

$$\Delta \hat{x}_1 - B \Delta u_0 = R_{v,0},$$

$$-\Delta \tilde{v}_{N-1} + C_z^T (\Delta \gamma_N - \Delta \zeta_N) = r_{x,N}.$$

The Lagrange multipliers associated with the inequality constraints $\Delta\eta$, $\Delta\delta$, Δv , $\Delta\omega$, $\Delta\gamma$, $\Delta\zeta$ and $\Delta\xi$ are eliminated from the equations above. This is computationally inexpensive as the matrices to be inverted in the process are all diagonal. The reduced set of equations are

$$B^T \Delta \tilde{v}_0 + U_0 \Delta u_0 = R_{u,0}, \quad (10a)$$

$$B^T \Delta \tilde{v}_i + U_i \Delta u_i + G_i \Delta \hat{x}_i = R_{u,i}, \quad i \in \tilde{N}_0, \quad (10b)$$

$$-\Delta \hat{x}_1 + B \Delta u_0 = R_{v,0}, \quad (10c)$$

$$-\Delta \hat{x}_{i+1} + A \Delta \hat{x}_i + B \Delta u_i = R_{v,i}, \quad i \in \tilde{N}_0, \quad (10d)$$

$$W_i \Delta \rho_i + M_i^T \Delta \hat{x}_i = R_{w,i}, \quad i \in \tilde{N}_1, \quad (10e)$$

$$-\Delta \tilde{v}_{i-1} + M_i \Delta \rho_i + \bar{X}_i \Delta \hat{x}_i + G_i^T \Delta u_i + A^T \Delta \tilde{v}_i = \bar{R}_{x,i}, \quad i \in \tilde{N}_0, \quad (10f)$$

$$-\Delta \tilde{v}_{N-1} + M_N \Delta \rho_N + \bar{X}_N \Delta \hat{x}_N = \bar{R}_{x,N}, \quad (10g)$$

where we have defined

$$U_i := \Sigma_{\eta,i}^{-1} + \Sigma_{\delta,i}^{-1} + \Sigma_{v,i}^{-1} + \Sigma_{w,i}^{-1}, \quad i \in \tilde{N}_0,$$

$$G_i := -(\Sigma_{\omega,i}^{-1} + \Sigma_{v,i}^{-1})D, \quad i \in \tilde{N}_0,$$

$$W_i := \Sigma_{\zeta,i}^{-1} + \Sigma_{\xi,i}^{-1} + \Sigma_{\gamma,i}^{-1}, \quad i \in \tilde{N}_1,$$

$$M_i := C_z^T (\Sigma_{\zeta,i}^{-1} - \Sigma_{\gamma,i}^{-1}), \quad i \in \tilde{N}_1,$$

$$\bar{X}_i := C_z^T (\Sigma_{\zeta,i}^{-1} + \Sigma_{v,i}^{-1})C_z + D^T (\Sigma_{\gamma,i}^{-1} + \Sigma_{\omega,i}^{-1})D, \quad i \in \tilde{N}_0,$$

$$\bar{X}_N := C_z^T (\Sigma_{\zeta,N}^{-1} + \Sigma_{v,N}^{-1})C_z,$$

and

$$R_{u,i} := r_{u,i} + \bar{r}_{\delta,i} + \bar{r}_{\omega,i} - \bar{r}_{\eta,i} - \bar{r}_{v,i}, \quad i \in \tilde{N}_0,$$

$$R_{v,i} := -R_{v,i}, \quad i \in \tilde{N}_0,$$

$$R_{w,i} := r_{w,i-1} + \bar{r}_{\zeta,i-1} + \bar{r}_{\xi,i} + \bar{r}_{\gamma,i}, \quad i \in \tilde{N}_1,$$

$$\bar{R}_{x,i} := r_{x,i} + C_z^T (\bar{r}_{\zeta,i} - \bar{r}_{\gamma,i}) + D^T (\bar{r}_{v,i} - \bar{r}_{w,i}), \quad i \in \tilde{N}_0,$$

$$\bar{R}_{x,N} := r_{x,N} + C_z^T (\bar{r}_{\zeta,N} - \bar{r}_{\gamma,N}).$$

Finally, $\bar{r}_{\delta,i} := \Sigma_{\delta,i}^{-1} r_{\delta,i}$. In a similar way we have defined $\bar{r}_{\omega,i}$, $\bar{r}_{\eta,i}$, $\bar{r}_{v,i}$, $\bar{r}_{\zeta,i}$, $\bar{r}_{\xi,i}$ and $\bar{r}_{\gamma,i}$. Solving (10e) for $\Delta \rho_i$ and substituting into the remaining equations of (10) yields

$$B^T \Delta \tilde{v}_0 + U_0 \Delta u_0 = R_{u,0}$$

$$B^T \Delta \tilde{v}_i + U_i \Delta u_i + G_i \Delta \hat{x}_i = R_{u,i}, \quad i \in \tilde{N}_0$$

$$-\Delta \hat{x}_1 + B \Delta u_0 = R_{v,0}$$

$$-\Delta \hat{x}_{i+1} + A \Delta \hat{x}_i + B \Delta u_i = R_{v,i}, \quad i \in \tilde{N}_0$$

$$-\Delta \tilde{v}_{i-1} + X_i \Delta \hat{x}_i + G_i^T \Delta u_i + A^T \Delta \tilde{v}_i = R_{x,i}, \quad i \in \tilde{N}_0$$

$$-\Delta \tilde{v}_{N-1} + X_N \Delta \hat{x}_N = R_{x,N}$$

where $X_i := \bar{X}_i - M_i W_i^{-1} M_i^T$ and $R_{x,i} := \bar{R}_{x,i} - M_i W_i^{-1} R_{w,i}$. As an example let $N = 3$. In this case, the equations above may be arranged as

$$\begin{bmatrix} U_0 B^T & & & & & \\ B & -I & & & & \\ -I & X_1 G_1^T & A^T & & & \\ & G_1 U_1 & B^T & & & \\ & A & B & -I & & \\ & & & -I & X_2 G_2^T & A^T \\ & & & & G_2 U_2 & B^T \\ & & & & A & B & -I \\ & & & & & & -I & X_3 \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta \tilde{v}_0 \\ \Delta \hat{x}_1 \\ \Delta u_1 \\ \Delta \tilde{v}_1 \\ \Delta \hat{x}_2 \\ \Delta u_2 \\ \Delta \tilde{v}_2 \\ \Delta \hat{x}_3 \end{bmatrix} = \begin{bmatrix} R_{u,0} \\ R_{v,0} \\ R_{x,1} \\ R_{u,1} \\ R_{v,1} \\ R_{x,2} \\ R_{u,2} \\ R_{v,2} \\ R_{x,3} \end{bmatrix} \quad (11)$$

This system can be solved by a Riccati iteration procedure in $O(N(n_u + n_x + n_z)^3)$ operations [4], [6], [7]. ■

IV. WARM-START

We apply the warm-start strategy of [3] to pick an initial point for the homogeneous and self-dual IPM. The method combines a guess of the solution (candidate point), $(\bar{t}, \bar{v}, \bar{w}, \bar{s})$, with the standard starting point $(0, 0, \mathbf{1}_{m_I}, \mathbf{1}_{m_I}, 1, 1)$. An important feature of the homogeneous and self-dual model is that the standard starting point is perfectly centralized with respect to the central path [3]. This makes warm-start work well for IPMs based on the homogeneous and self-dual model. The initial point in [3] is defined as

$$\begin{aligned} \tilde{w}^0 &= \lambda \bar{w} + (1 - \lambda) \mathbf{1}_{m_I}, & \tilde{s}^0 &= \lambda \bar{s} + (1 - \lambda) \mathbf{1}_{m_I}, \\ \tilde{t}^0 &= \lambda \bar{t}, & \tilde{v}^0 &= \lambda \bar{v}, \\ \tilde{\tau}^0 &= 1, & \tilde{\kappa}^0 &= (\tilde{w}^0)^T \tilde{s}^0 / m_I, \end{aligned}$$

where $\lambda \in [0, 1]$ is a tuning parameter. When $\lambda = 0$ the initial point becomes the standard starting point, and for $\lambda = 1$ the initial point becomes the candidate point.

In MPC applications, the optimal control problem is solved in a receding horizon manner. A good choice of the candidate point at time k can therefore be constructed using the solution from the previous time step. As an example consider the solution of (3) and (4) at time step $k = 0$, for $N = 3$

$$t^* := [u_0^{*T} \quad \hat{x}_1^{*T} \quad \rho_1^{*T} \quad u_1^{*T} \quad \hat{x}_2^{*T} \quad \rho_2^{*T} \quad u_2^{*T} \quad \hat{x}_3^{*T} \quad \rho_3^{*T}]^T.$$

The following candidate point is then used at time step $k = 1$

$$\bar{t} := [u_1^{*T} \quad \hat{x}_2^{*T} \quad \rho_2^{*T} \quad u_2^{*T} \quad \hat{x}_3^{*T} \quad \rho_3^{*T} \quad u_2^{*T} \quad \hat{x}_3^{*T} \quad \rho_3^{*T}]^T.$$

Similarly, we left-shift the slack variables, s , and the dual variables, v and w , to construct \bar{s} , \bar{v} and \bar{w} .

V. POWER MANAGEMENT CASE STUDY

In this section, we compare LPempc against IPMs from the following software packages: Gurobi, SeDuMi, MOSEK, LIPSOL, GLPK. These state-of-the-art IPMs are mainly written in low-level language such as FORTRAN and C, and rely on sparse linear algebra that are specifically tailored to the solution of large-scale sparse linear and conic programs. We also include the simplex method provided by CPLEX in our comparisons, as well as FORCES that is an IPM based on automatic code generation [16]. All solvers are called from MATLAB using MEX interfaces. We have performed our simulations using an Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz with 4 GB RAM running a 64-bit Ubuntu 12.04.1 LTS operating system.

The test system is a system of m generic power generating units in the form introduced in [17]. For $i = 1, 2, \dots, m$

$$Y_i(s) = \frac{1}{(\tau_i s + 1)^3} (U_i(s) + D_i(s)) + E_i(s). \quad (12)$$

$D_i(s)$ is the process noise, $E_i(s)$ is the measurement noise, $U_i(s)$ is the power set-point and $Y_i(s)$ is the power production. The total production from the m power generating units is the sum $Z(s) = \sum_{i=1}^m \frac{1}{(\tau_i s + 1)^3} (U_i(s) + D_i(s))$. We convert the

TABLE I
 CASE STUDY PARAMETERS

	τ_i	p_k	\underline{u}_k	\bar{u}_k	$\Delta \underline{u}_k$	$\Delta \bar{u}_k$
Power Plant 1	90	100	0	200	-20	20
Power Plant 2	30	200	0	150	-40	40

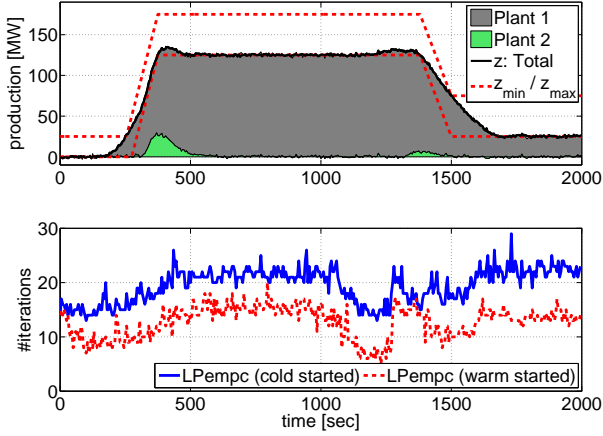


Fig. 1. Closed-loop simulation of a power system controlled by economic MPC. Warm-start ($\lambda = 0.99$) yields a significant reduction in the number of iterations.

transfer function model into the state space form (1) using a sampling time of $T_s = 5$ seconds. In the resulting model structure, $u_k \in \mathbb{R}^{n_u}$ is the n_u power set-points, $y_k \in \mathbb{R}^{n_y}$ is the measured power production, and $z_k \in \mathbb{R}^{n_z}$ is the total power production. Note that $n_u = n_y = m$ and $n_z = 1$. It is assumed that $d_k \sim N(0, \sigma I)$ and $e_k \sim N(0, \sigma I)$.

A. Simulations

An example with two power generating units is considered; a cheap/slow unit, and an expensive/fast unit. This conflict between response time and operating costs represents a common situation in the power industry where large thermal power plants often produce a majority of the electricity, while the use of units with faster dynamics such as diesel generators and gas turbines are limited to critical peak periods. The controller objective is to coordinate the most cost-efficient power production, respecting capacity constraints and a time-varying electricity demand. It is assumed that full information about the initial state is given $x_0 \sim (0, 0)$, and that the penalty of violating the output constraints is $q_k = 10^4$ for all time steps. Table I lists the system and controller parameters. We set the prediction horizon to $N = 80$ time steps. It has been verified by simulation that the controller is stable for this value of N .

Fig. 1 shows a closed-loop simulation with $\sigma = 1$. The figure illustrates the power production of each power generating unit. The cheap unit produces 97% of the energy, while the expensive unit is activated only to compensate for the power imbalances otherwise caused by the slow unit. Fig. 1 also shows that warm-start yields a significant reduction in the number of iterations. On average the number of iterations is reduced by approximately 37%.

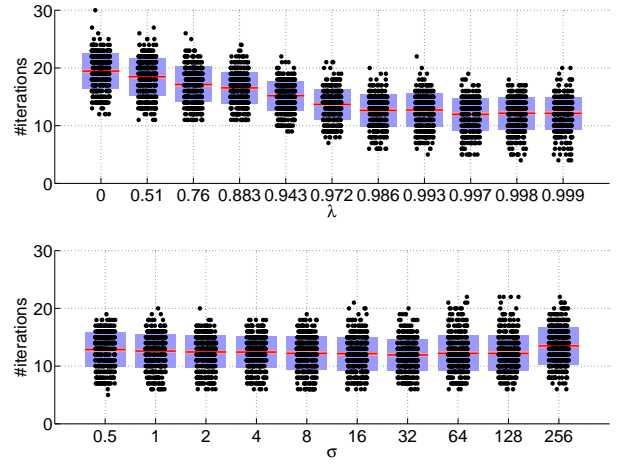
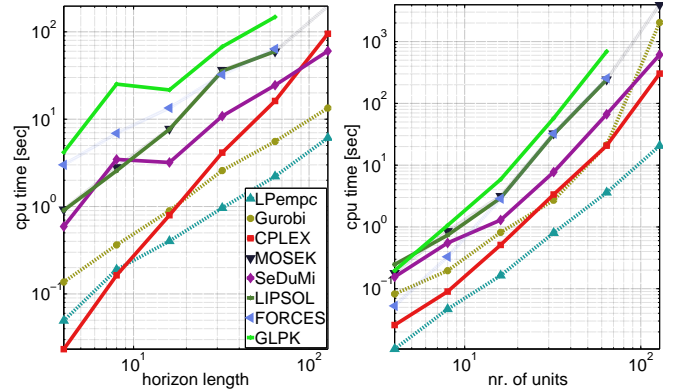


Fig. 2. Number of iterations as a function of the tuning parameter λ and the noise parameter σ . Each box-plot has been generated based on an entire closed-loop simulation. In the top plot we have fixed $\sigma = 1$, and in the bottom plot $\lambda = 0.99$.



(a) Increasing N and fixed $m = 32$. (b) Increasing m and fixed $N = 32$.
 Fig. 3. CPU-time for the different solvers as a function of the horizon N length and the number of power generators m .

Fig. 2 shows a number of box-plots used to select the warm-start parameter λ . The case $\lambda = 0$ corresponds to a cold-start. For all values of λ , warm-start reduces the average number of iterations. We have chosen $\lambda = 0.99$ for our controller. This value of λ yields an initial point which is both close to the candidate point and lies well inside the interior of the non-negative orthant, $(\bar{w}, \bar{s}, \bar{\kappa}, \bar{\tau}) \geq 0$. Fig. 2 shows that for $\lambda = 0.99$, the number of iterations is reduced even when the variance of the process and measurement noise is increased significantly.

Fig. 3 plots the computation time for solving the LP (2) as a function of the number of power generating units, m , and the length of the horizon, N . The figure shows that LPmpc is faster than all other solvers with a significant margin. In addition to the algorithms included in Fig. 3, the problem (2) was solved using the code generation based IPM CVXGEN [18]. For problems larger than $m = 4$ and $N = 12$, code generation in CVXGEN fails due to the problem size. Therefore, we have not included results for CVXGEN in our benchmark. In general, code generation based solvers such as CVXGEN and

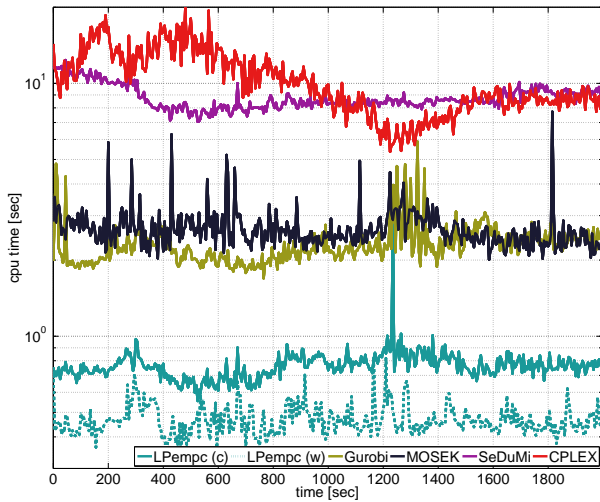


Fig. 4. CPU-time for solving (2) with 15 power generating units and a prediction horizon of 200 time steps. We observe that warm-starting reduces the average number of iterations by approximately 40%.

FORCES are most competitive for small-dimensional problems [16]. Fig. 4 shows CPU-timings for a closed-loop simulation with 15 power generating units and a prediction horizon of $N = 200$ time steps. Only the most competitive solvers are included. In this simulation LPempc is up to an order of magnitude faster than CPLEX, Gurobi, SeDuMi and MOSEK, depending on the problem data. On average, LPempc is approximately 5 times faster than Gurobi, 6 times faster than MOSEK, 19 times faster than SeDuMi, and 22 times faster than CPLEX.

VI. CONCLUSIONS

In this paper, we have developed a computationally efficient IPM for economic MPC of linear systems with a linear objective function and linear constraints. The algorithm combines the homogeneous and self-dual model, and a Riccati iteration procedure specifically tailored to MPC. This is a significant contribution since existing Riccati iteration procedures for MPC are not directly applicable to the homogeneous and self-dual model that has become widely adopted by state-of-the-art IPMs for linear programming. We have also implemented and tested a recent warm-start strategy for homogeneous and self-dual IPMs that has not previously been used in MPC applications. Our simulations show that this strategy reduces the average number of iterations by 35-40%, and that a MATLAB and C implementation of the proposed algorithm, LPempc, is significantly faster than several state-of-the-art IPMs, as well as automatic code generation based IPMs for MPC. In a conceptual power management case study, LPempc is up to an order of magnitude faster than CPLEX, Gurobi, SeDuMi and MOSEK. This is important, since the computing time of solving the optimal control problem is critical in MPC applications. The simulation results also show that the difference in computing time becomes larger with the problem size as LPempc scales in a favorable way.

ACKNOWLEDGEMENT

This work was funded in part by 1) the Danish Ministry of Higher Education and Science in the industrial PhD project "Stochastic MPC with Applications in Smart Energy Systems" (11-117435); 2) the Southern Denmark Growth Forum and the European Regional Development Fund in the project "Smart & Cool"; and 3) The Danish Council for Strategic Research in the project "CITIES - Centre for IT-Intelligent Energy Systems in Cities" (1305-00027B).

REFERENCES

- [1] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, 1992.
- [2] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.
- [3] A. Skajaa, E. D. Andersen, and Y. Ye, "Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems," *Mathematical Programming Computation*, vol. 5, no. 1, pp. 1–25, 2013.
- [4] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of Interior-Point Methods to Model Predictive Control," *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 723–757, 1998.
- [5] J. B. Jørgensen, G. Frison, N. F. Gade-Nielsen, and B. Dammann, "Numerical Methods for Solution of the Extended Linear Quadratic Control Problem," in *4th IFAC Nonlinear Model Predictive Control Conference*, 2012, pp. 187–193.
- [6] J. B. Jørgensen, J. B. Rawlings, and S. B. Jørgensen, "Numerical Methods for Large-Scale Moving Horizon Estimation and Control," in *International Symposium on Dynamics and Control Process Systems (DYCOPS)*, vol. 7, 2004.
- [7] L. Vandenberghe, S. Boyd, and M. Nouralishahi, "Robust linear programming and optimal control," in *Proceedings of the 15th IFAC World Congress*, vol. 15, 2002, pp. 271–276.
- [8] E. D. Andersen, J. Gondzio, C. Meszaros, and X. Xu, "Implementation of Interior Point Methods for Large Scale Linear Programming," Ecole des Hautes Etudes Commerciales, Universite de Geneve, Papers 96.3, 1996.
- [9] X. Xu, P.-F. Hung, and Y. Ye, "A simplified homogeneous and self-dual linear programming algorithm and its implementation," *Annals of Operations Research*, vol. 62, no. 1, pp. 151–171, 1996.
- [10] Y. Ye, M. J. Todd, and S. Mizuno, "An $O(\sqrt{n}L)$ -Iteration Homogeneous and Self-Dual Linear Programming Algorithm," *Mathematics of Operations Research*, vol. 19, no. 1, pp. 53–67, 1994.
- [11] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.
- [12] V. M. Zavala and L. T. Biegler, "Nonlinear Model Predictive Control," in *Control, Nonlinear Programming Strategies for State Estimation and Model Predictive*, ser. Lecture Notes in Control and Information Sciences, L. Magni, D. M. Raimondo, and F. Allgöwer, Eds. Springer Berlin Heidelberg, 2009, vol. 384, pp. 419–432.
- [13] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov Function for Economic Optimizing Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2011.
- [14] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, no. 3, pp. 725–734, 2013.
- [15] L. E. Sokoler, G. Frison, A. Skajaa, R. Halvgaard, and J. B. Jørgensen, "A Homogeneous and Self-Dual Interior-Point Linear Programming Algorithm for Economic Model Predictive Control," Department of Applied Mathematics and Computer Science, Technical University of Denmark, Tech. Rep., 2014. [Online]. Available: http://www2.imm.dtu.dk/~jbjo/DTU_MPC_2013_25.html
- [16] A. Domahidi, A. Zraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control," in *51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 668–674.
- [17] K. Edlund, T. Mølbak, and J. D. Bendtsen, "Simple models for model-based portfolio load balancing controller synthesis," in *6th IFAC Symposium on Power Plants and Power Systems Control*, 2009, pp. 173–178.
- [18] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.