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

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

Abstract

In this paper, we develop an efficient homogeneous and self-dual interior-point method (IPM) for the linear programs arising in economic model predictive control (MPC) of linear systems. To exploit the structure in the optimization problems, our algorithm utilizes a Riccati iteration procedure which is adapted to the linear system of equations solved in homogeneous and self-dual IPMs, and specifically tailored to economic MPC. Fast convergence is further achieved by means of a recent warm-start strategy for homogeneous and self-dual IPMs that has not previously been applied to MPC. We implement our algorithm in MATLAB and C, and its performance is analyzed using a conceptual power management case study. Closed loop simulations show that 1) our algorithm is significantly faster than several state-of-the-art IPMs based on sparse linear algebra, and 2) warm-start reduces the number of iterations by 35-40%.

Index Terms

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

I. Introduction

During the last 30-40 years, model predictive control (MPC) for constrained dynamic systems has become the most successful advanced control technology in the process industries [1]–[4]. MPC is attractive because of its ability to handle constraints, time delays, and multivariate systems in a straightforward and transparent way.

The basic idea of MPC is to optimize the predicted behavior of a dynamic model over a finite horizon by solving an optimization problem. The optimization problem is updated and resolved at each sampling instant when new information is available. Conventionally, the optimization problem solved in MPC is formulated as a convex program that penalizes deviations between the controlled output and a set-point. While this approach ensures that the set-point is reached in a reasonable amount of time, it does not guarantee that the transition between set-points is

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, giaf, andsk, rhal, jbjo}@dtu.dk)

performed in an economically efficient way. To overcome this problem, economic MPC has emerged as a promising technology which, as an alternative to classical set-point based MPC, provides a systematic method for optimizing economic performance [5]–[11]. Stability of economic MPC has been addressed in [5]–[9], [12].

One research area where economic MPC is receiving a growing amount of attention is within control of smart energy systems. Such systems typically consist of a large number of power generators and consumers with diverse operational features and capabilities. These units operate with on available power and operating conditions, and system constraints arise due to the power demand and flexibility requirements. Applications of economic MPC in smart energy systems include cost-efficient control of refrigeration systems [13], building climate control [14], [15], and charging batteries in electric vehicles [16].

In linear economic MPC, the constrained optimal control problem can be posed as a linear program. As the optimization problem is solved online, the performance and reliability of the optimization algorithm solving the linear program is important. In this paper, we develop a homogeneous and self-dual variant of Mehrotra's predictor-corrector method [17], [18] for economic MPC that combines the following performance improvement components:

- **Riccati Iteration Procedure**: The optimization problem solved in linear economic MPC is highly structured. We exploit this structure to speed-up the most time consuming numerical operations using a Riccati iteration procedure.
- **Warm-Start**: In MPC applications, the optimization problems solved at successive time steps are closely related. To take advantage of the solution from the previous sampling instant, we implement a recently developed warm-start strategy for homogeneous and self-dual IPMs. This method does not introduce any additional significant computations and has been reported to reduce the number of iterations by 30-75% based on the `NETLIB` collection of test problems [19].

Although Riccati-based IPMs have been developed in [20]–[23] for set-point based MPC with an ℓ_2-penalty, and in [24] with ℓ_1-penalty, these results are not directly applicable to the homogeneous and self-dual model, which has become widely adopted by state-of-the-art IPMs for linear programming [25], [26]. Moreover, a Riccati iteration procedure has not previously been efficiently implemented for economic MPC. We also remark that no results on warm-start MPC applications using the strategy of [19] has been reported.

A. Paper Organization

This paper is organized as follows. In Section II, we formulate the control law associated with economic MPC as the solution to a highly structured linear program. A homogeneous and self-dual IPM for the linear program is derived in Section III. Section IV and Section V demonstrate how to implement the IPM efficiently using a Riccati iteration procedure and linear algebra routines specifically tailored to economic MPC. Warm-start is discussed in Section VI. In Section VII, a MATLAB and C implementation of our algorithm denoted `LPempc` is compared to several state-of-the-art IPMs in a simple power management case study[1]. Concluding remarks are given in Section

[1] Software is available via http://www2.imm.dtu.dk/~jbjo/publications

VIII. Details on our Riccati iteration procedure for economic MPC are enclosed in Appendix VIII-A.

II. Optimal Control Problem

We consider linear state space systems in the form

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + Bu_k + E\boldsymbol{d}_k, \qquad \boldsymbol{d}_k \sim N(0, R_d), \tag{1a}$$

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

$$\boldsymbol{z}_k = C_z\boldsymbol{x}_k, \tag{1c}$$

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

Economic MPC based on the separation and certainty equivalence principle defines a control law for the system (1) that optimizes the control actions with respect to an economic objective function, input limits, input-rate limits and soft output limits. Evaluation of this control law requires the solution to the linear program

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

s.t.

$$\hat{x}_{k+j+1|k} = A\hat{x}_{k+j|k} + Bu_{k+j}, \qquad j \in \mathcal{N}_0, \tag{2b}$$

$$\hat{z}_{k+j|k} = C_z\hat{x}_{k+j|k}, \qquad j \in \mathcal{N}_1, \tag{2c}$$

$$\underline{u}_{k+j} \leq u_{k+j} \leq \overline{u}_{k+j}, \qquad j \in \mathcal{N}_0, \tag{2d}$$

$$\Delta\underline{u}_{k+j} \leq \Delta u_{k+j} \leq \Delta\overline{u}_{k+j}, \qquad j \in \mathcal{N}_0, \tag{2e}$$

$$\underline{z}_{k+j} - \rho_{k+j} \leq \hat{z}_{k+j|k} \leq \overline{z}_{k+j} + \rho_{k+j}, \qquad j \in \mathcal{N}_1, \tag{2f}$$

$$\rho_{k+j} \geq 0, \qquad j \in \mathcal{N}_1, \tag{2g}$$

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 estimate, $\hat{x}_{k|k}$, the input limits, $(\underline{u}_{k+j}, \overline{u}_{k+j})$, the input-rate limits, $(\Delta\underline{u}_{k+j}, \Delta\overline{u}_{k+j})$, the output limits, $(\underline{z}_{k+j}, \overline{z}_{k+j})$, the input prices, p_{k+j}, and the price for violating the output constraints, q_{k+j}. As an example, in power systems management p_k may be the cost of fuel and q_k may be the cost of not meeting the power demand.

.

The filtered estimate, $\hat{x}_{k|k} := E[\boldsymbol{x}_k|Y_k]$, is the conditional expectation of $\boldsymbol{x}_k$ given the observations $Y_k := \begin{bmatrix} y_0^T & y_1^T & y_2^T & \dots & y_k^T \end{bmatrix}^T$. We obtain this value using the Kalman filter.

The input-rate is defined terms of the backward difference operator

$$\Delta u_{k+j} := u_{k+j} - u_{k+j-1}, \qquad j \in \mathcal{N}_0.$$

By augmenting the state-space system such that

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

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

we can express (2e) as

$$\Delta\underline{u}_{k+j} \leq u_{k+j} - Dx_k \leq \Delta\overline{u}_{k+j}, \qquad\qquad j \in \mathcal{N}_0,$$

in which $D := \begin{bmatrix} 0 & I \end{bmatrix}$. 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.

A. Linear Program Formulation

By aggregating the problem data into the structures g, F, b, H and c, (2) can be put into the form

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

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

$$t := \begin{bmatrix} u_0^T & \hat{x}_1^T & \rho_1^T & u_1^T & \hat{x}_2^T & \rho_2^T \end{bmatrix}^T,$$

$$g := \begin{bmatrix} p_0^T & 0 & q_1^T & p_1^T & 0 & q_2^T \end{bmatrix}^T,$$

and

$$\left[F \,\middle|\, b \right] := \left[\begin{array}{cccccc|c} B & -I & 0 & 0 & 0 & 0 & -A\hat{x}_0 \\ 0 & A & 0 & B & -I & 0 & 0 \end{array} \right],$$

$$\left[H \,\middle|\, c \right] := \left[\begin{array}{cccccc|c} I & 0 & 0 & 0 & 0 & 0 & \overline{u}_0 \\ 0 & 0 & 0 & I & 0 & 0 & \overline{u}_1 \\ -I & 0 & 0 & 0 & 0 & 0 & -\underline{u}_0 \\ 0 & 0 & 0 & -I & 0 & 0 & -\underline{u}_1 \\ I & 0 & 0 & 0 & 0 & 0 & \Delta\tilde{u}_0 \\ 0 & -D & 0 & I & 0 & 0 & \Delta\overline{u}_1 \\ -I & 0 & 0 & 0 & 0 & 0 & -\Delta\utilde{u}_0 \\ 0 & D & 0 & -I & 0 & 0 & -\Delta\underline{u}_1 \\ 0 & C_z & -I & 0 & 0 & 0 & \overline{z}_1 \\ 0 & 0 & 0 & 0 & C_z & -I & \overline{z}_2 \\ 0 & -C_z & -I & 0 & 0 & 0 & -\underline{z}_1 \\ 0 & 0 & 0 & 0 & -C_z & -I & -\underline{z}_2 \\ 0 & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I & 0 \end{array} \right],$$

where

$$\Delta\tilde{u}_0 := \Delta\overline{u}_0 + D\hat{x}_0,$$

$$\Delta\underset{\sim}{u}_0 := \Delta\underline{u}_0 + D\hat{x}_0.$$

Hence, (2) can be posed as a highly structured linear program 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.

.

III. Homogeneous and Self-Dual Model

The dual of the linear program (3) is

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

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).

In homogeneous and self-dual IPMs, the solution to (3)-(4) is obtained by solving a related homogeneous and self-dual linear program [27]–[29]. 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 [19].

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)-(4), may be stated as the linear feasibility problem

$$\tilde{t}, \tilde{v}, \tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa}, \tag{5a}$$

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

$$b\tilde{\tau} - F\tilde{t} = 0, \tag{5c}$$

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

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

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

Proposition 1 shows that the solution to (3)-(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 $j = 1, 2, \ldots, m_I$ and $\tilde{\tau}^*\tilde{\kappa}^* = 0$. For such a solution, one of the following conditions hold

- **I)** $\tilde{\tau}^* > 0$ and $\tilde{\kappa}^* = 0$:

  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)-(4).
- **II)** $\tilde{\tau}^* = 0$ and $\tilde{\kappa}^* > 0$:

  The problem (3) is infeasible or unbounded; either $-b^T\tilde{v}^* - c^T\tilde{w}^* > 0$ (implies primal infeasibility), or $g^T\tilde{t}^* < 0$ (implies dual infeasibility).

Proof: See [29]. ■

A. Interior Point Method

We now present a homogeneous and self-dual IPM for solving (5). For compact notation, we denote the optimization variables by $\phi := (\tilde{t}, \tilde{v}, \tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa})$, and introduce superscript k to indicate a particular iteration number. The necessary and sufficient optimality conditions for (5) are $(\tilde{w}, \tilde{s}, \tilde{\kappa}, \tilde{\tau}) \geq 0$ and

$$V(\phi) := \begin{bmatrix} V_1(\phi) \\ V_2(\phi) \\ V_3(\phi) \\ V_4(\phi) \\ V_5(\phi) \\ V_6(\phi) \end{bmatrix} = \begin{bmatrix} F^T\tilde{v} + H^T\tilde{w} + g\tilde{\tau} \\ b\tilde{\tau} - F\tilde{t} \\ c\tilde{\tau} - H\tilde{t} - \tilde{s} \\ -g^T\tilde{t} - b^T\tilde{v} - c^T\tilde{w} + \tilde{\kappa} \\ \tilde{W}\tilde{S}\mathbf{1}_{m_I} \\ \tilde{\tau}\tilde{\kappa} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

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

To find a point satisfying the optimality conditions, we use a variant of Mehrotra's predictor-corrector method [17], [18]. The method tracks the central path $\mathcal{C}$, which connects an initial point ϕ^0 satisfying $(\tilde{w}^0, \tilde{s}^0, \tilde{\kappa}^0, \tilde{\tau}^0) \geq 0$ to a strict complementary solution of (5), denoted ϕ^*. Formally, we can write the central path as

$$\mathcal{C} := \left\{ \phi | V(\phi) = \gamma r^0, (\tilde{w}, \tilde{s}, \tilde{\kappa}, \tilde{\tau}) \geq 0, \gamma \in [0,1] \right\}.$$

In this definition

$$r^0 := \left[V_1(\phi^0)^T \; V_2(\phi^0)^T \; V_3(\phi^0)^T \; V_4(\phi^0)^T \; \mu^0 \mathbf{1}_{m_I}^T \; \mu^0 \right]^T,$$

where $\mu^0 := ((\tilde{w}^0)^T\tilde{s}^0 + \tilde{\tau}^0\tilde{\kappa}^0)/(m_I + 1)$ is a measure of the duality gap, and $V_i(\phi)$ is the i'th set of components of $V(\phi)$, defined as in (6).

The basic idea in IPMs is to generate a sequence of iterates along the central path $\{\phi^0, \phi^1, \ldots, \phi^k, \ldots, \phi^N\}$, such that $\phi^N \to \phi^*$ as $N \to \infty$. In Mehrotra's predictor-corrector method, the iterates are computed by repeating a two-step procedure. In the first part of the procedure an affine-scaling direction is determined and an affine step is computed. Secondly a center-corrector direction is found using information from the affine step. The resulting optimizing search direction is then the sum of the affine-scaling direction and the center-corrector direction.

Affine Step: The affine-scaling direction, $\Delta\phi^k_{\text{aff}}$, corresponds to the Newton direction for (6), and is obtained by solving the linear system of equations

$$J_V(\phi^k)\Delta\phi^k_{\text{aff}} = -V(\phi^k). \tag{7}$$

The Jacobian of V evaluated at ϕ^k is

$$J_V(\phi^k) = \begin{bmatrix} 0 & F^T & H^T & 0 & g & 0 \\ -F & 0 & 0 & 0 & b & 0 \\ -H & 0 & 0 & -I & c & 0 \\ -g^T & -b^T & -c^T & 0 & 0 & 1 \\ 0 & 0 & \tilde{S}^k & \tilde{W}^k & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{\kappa}^k & \tilde{\tau}^k \end{bmatrix}. \tag{8}$$

Given the solution to (7), we find the maximum step length in the affine-scaling direction for the primal and dual variables, such that (5f) remains satisfied

$$\alpha_{\text{aff}}^k := \max \left\{ a_{\text{aff}} \in [0,1] | \begin{bmatrix} \tilde{w}^k \\ \tilde{\tau}^k \end{bmatrix} + a_{\text{aff}} \begin{bmatrix} \Delta\tilde{w}_{\text{aff}}^k \\ \Delta\tilde{\tau}_{\text{aff}}^k \end{bmatrix} \geq 0 \right\},$$

$$\beta_{\text{aff}}^k := \max \left\{ b_{\text{aff}} \in [0,1] | \begin{bmatrix} \tilde{s}^k \\ \tilde{\kappa}^k \end{bmatrix} + b_{\text{aff}} \begin{bmatrix} \Delta\tilde{s}_{\text{aff}}^k \\ \Delta\tilde{\kappa}_{\text{aff}}^k \end{bmatrix} \geq 0 \right\}.$$

Accordingly, affine variables are computed as

$$\tilde{w}_{\text{aff}}^k := \tilde{w}^k + \alpha_{\text{aff}}^k \Delta\tilde{w}_{\text{aff}}^k, \qquad \tilde{s}_{\text{aff}}^k := \tilde{s}^k + \beta_{\text{aff}}^k \Delta\tilde{s}_{\text{aff}}^k,$$

$$\tilde{\tau}_{\text{aff}}^k := \tilde{\tau}^k + \alpha_{\text{aff}}^k \Delta\tilde{\tau}_{\text{aff}}^k, \qquad \tilde{\kappa}_{\text{aff}}^k := \tilde{\kappa}^k + \beta_{\text{aff}}^k \Delta\tilde{\kappa}_{\text{aff}}^k.$$

The affine variables provide a measure of the relative reduction in the duality gap, in the affine-scaling direction. This information is used to update the centering parameter γ. For this purpose, we use the following heuristic proposed in [18]

$$\gamma^k := \left[\frac{\mu_{\text{aff}}^k}{\mu^k}\right]^3 = \left[\frac{((\tilde{w}_{\text{aff}}^k)^T \tilde{s}_{\text{aff}}^k + \tilde{\tau}_{\text{aff}}^k \tilde{\kappa}_{\text{aff}}^k)}{((\tilde{w}^k)^T \tilde{s}^k + \tilde{\tau}^k \tilde{\kappa}^k)}\right]^3. \tag{9}$$

Search Direction: The optimizing search direction, $\Delta\phi^k$, is the sum of the affine-scaling direction and a center-corrector direction. Using the approach described in e.g. [18] and [30], we determine this direction by solving (7) with a modified right hand side

$$J_V(\phi^k)\Delta\phi^k = -\bar{V}(\phi^k). \tag{10}$$

and

$$\bar{V}_5(\phi^k) := V_5(\phi^k) + \Delta\tilde{W}_{\text{aff}}^k \Delta\tilde{S}_{\text{aff}}^k \mathbf{1}_{m_I} - \gamma^k \mu^k \mathbf{1}_{m_I}^T,$$

$$\bar{V}_6(\phi^k) := V_6(\phi^k) + \Delta\tilde{\tau}_{\text{aff}}^k \Delta\tilde{\kappa}_{\text{aff}}^k - \gamma^k \mu^k.$$

The diagonal matrices $\Delta\tilde{W}_{\text{aff}}^k$ and $\Delta\tilde{S}_{\text{aff}}^k$ are defined in a similar way to and . Terms involving these matrices are, as well as $\Delta\tilde{\tau}_{\text{aff}}^k$ and $\Delta\tilde{\kappa}_{\text{aff}}^k$, included to compensate for linearization errors [18]. We also notice that by employing the heuristic (9), the search direction is forced towards the central path if $\mu_{\text{aff}}^k \approx \mu^k$, meaning that only a small step in the non-negative orthant $(\tilde{w}, \tilde{s}, \tilde{\kappa}, \tilde{\tau}) \geq 0$ is available in the affine-scaling direction.

1) Stopping Criteria: To classify a solution as optimal, we adopt the criteria proposed in [30]

$$\varrho_E^k \leq \epsilon_E, \qquad \varrho_I^k \leq \epsilon_I, \qquad \varrho_D^k \leq \epsilon_D, \qquad \varrho_O^k \leq \epsilon_O. \tag{11}$$

Moreover, the problem is considered to be infeasible if $\tilde{\tau}^k \leq \epsilon_\tau \max(1, \tilde{\kappa}^k)$, and

$$\varrho_E^k \leq \epsilon_E, \qquad \varrho_I^k \leq \epsilon_I, \qquad \varrho_D^k \leq \epsilon_D, \qquad \varrho_G^k \leq \epsilon_G. \tag{12}$$

ϵ_τ, ϵ_E, ϵ_I, ϵ_D, ϵ_O and ϵ_G are small user-defined tolerances and

$$\begin{aligned}
\varrho_D &:= ||V_1(\phi)||_\infty / \max(1, || \begin{bmatrix} H^T & F^T & g \end{bmatrix} ||_\infty), \\
\varrho_E &:= ||V_2(\phi)||_\infty / \max(1, || \begin{bmatrix} F & b \end{bmatrix} ||_\infty), \\
\varrho_I &:= ||V_3(\phi)||_\infty / \max(1, || \begin{bmatrix} H & I & c \end{bmatrix} ||_\infty), \\
\varrho_G &:= |L - \tilde{\kappa}| / \max(1, || \begin{bmatrix} g^T & b^T & c^T & 1 \end{bmatrix} ||_\infty), \\
\varrho_O &:= |L| / (\tilde{\tau} + | - b^T \tilde{v} - c^T \tilde{w}|).
\end{aligned}$$

where $L := g^T \tilde{t} - (-b^T \tilde{v} - c^T \tilde{w})$ is the duality gap.

2) Algorithm: Algorithm 1 summarizes the homogeneous and self-dual IPM described in this paper. We use a damping parameter, ν, to keep the iterates well inside the interior of the non-negative orthant, $(\tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa}) \geq 0$.

IV. Riccati Iteration Procedure

The main computational efforts in Algorithm 1 are the solution of the linear systems (7) and (10). For an arbitrary right hand side, we can write these operations as

$$F^T \Delta\tilde{v} + H^T \Delta\tilde{w} + g \Delta\tilde{\tau} = r_1, \tag{13a}$$

$$b \Delta\tilde{\tau} - F \Delta\tilde{t} = r_2, \tag{13b}$$

$$c \Delta\tilde{\tau} - H \Delta\tilde{t} - \Delta\tilde{s} = r_3, \tag{13c}$$

$$g^T \Delta\tilde{t} + b^T \Delta\tilde{v} + c^T \Delta\tilde{w} - \Delta\tilde{\kappa} = r_4, \tag{13d}$$

$$\tilde{W} \Delta\tilde{s} + \tilde{S} \Delta\tilde{w} = r_5, \tag{13e}$$

$$\tilde{\kappa} \Delta\tilde{\tau} + \tilde{\tau} \Delta\tilde{\kappa} = r_6. \tag{13f}$$

We remark that the system (13) is different from the system solved in standard IPMs, due to the introduction of the homogeneous and self-dual model. Consequently, existing Riccati iteration procedures for MPC cannot be applied directly. As shown in Proposition 2 however, the solution to (13) can be obtained by solving a reduced linear system and a number of computationally inexpensive operations.

Algorithm 1 Homogeneous and self-dual IPM for (5)

```
Require: DATA             $(g, F, b, H, c)$
         INITIAL POINT    $(\tilde{t}, \tilde{v}, \tilde{w}, \tilde{s}, \tilde{\tau}, \tilde{\kappa})$
         PARAMETERS       $\nu \in [0.95; 0.999]$

// initialize
$\mu \leftarrow (\tilde{w}^T \tilde{s} + \tilde{\tau}\tilde{\kappa})/(m_I + 1)$
while not CONVERGED do
    // affine-scaling direction
    $\Delta\phi_{\text{aff}} \leftarrow -J_V(\phi)^{-1} V(\phi)$
    $\alpha_{\text{aff}} \leftarrow \max\{a_{\text{aff}} \in [0,1] | (\tilde{w}, \tilde{\tau}) + a_{\text{aff}}(\Delta\tilde{w}_{\text{aff}}, \Delta\tilde{\tau}_{\text{aff}}) \geq 0\}$
    $\beta_{\text{aff}} \leftarrow \max\{b_{\text{aff}} \in [0,1] | (\tilde{s}, \tilde{\kappa}) + b_{\text{aff}}(\Delta\tilde{s}_{\text{aff}}, \Delta\tilde{\kappa}_{\text{aff}}) \geq 0\}$
    $\tilde{s}_{\text{aff}} \leftarrow \tilde{s} + \beta_{\text{aff}}\Delta\tilde{s}_{\text{aff}}, \quad \tilde{\kappa}_{\text{aff}} \leftarrow \tilde{\kappa} + \tilde{\beta}_{\text{aff}}\Delta\tilde{\kappa}_{\text{aff}}$
    $\tilde{w}_{\text{aff}} \leftarrow \tilde{w} + \alpha_{\text{aff}}\Delta\tilde{w}_{\text{aff}}, \quad \tilde{\tau}_{\text{aff}} \leftarrow \tilde{\tau} + \tilde{\alpha}_{\text{aff}}\Delta\tilde{\tau}_{\text{aff}}$
    $\mu_{\text{aff}} \leftarrow (\tilde{w}_{\text{aff}}^T \tilde{s}_{\text{aff}} + \tilde{\tau}_{\text{aff}}\tilde{\kappa}_{\text{aff}})/(m_I + 1)$
    $\gamma \leftarrow (\mu_{\text{aff}}/\mu)^3$
    // optimizing search direction
    $\Delta\phi \leftarrow -J_V(\phi)^{-1} \bar{V}(\phi)$
    $\alpha \leftarrow \max\{a \in [0,1] | (\tilde{w}, \tilde{\tau}) + a(\Delta\tilde{w}, \Delta\tilde{\tau}) \geq 0\}$
    $\beta \leftarrow \max\{b \in [0,1] | (\tilde{s}, \tilde{\kappa}) + b(\Delta\tilde{s}, \Delta\tilde{\kappa}) \geq 0\}$
    $\tilde{t} \leftarrow \tilde{t} + \nu\beta\Delta\tilde{t}, \quad \tilde{s} \leftarrow \tilde{s} + \nu\beta\Delta\tilde{s}, \quad \tilde{\kappa} \leftarrow \tilde{\kappa} + \nu\beta\Delta\tilde{\kappa}$
    $\tilde{v} \leftarrow \tilde{v} + \nu\alpha\Delta\tilde{v}, \quad \tilde{w} \leftarrow \tilde{w} + \nu\alpha\Delta\tilde{w}, \quad \tilde{\tau} \leftarrow \tilde{\tau} + \nu\alpha\Delta\tilde{\tau}$
    $\mu \leftarrow (\tilde{w}^T \tilde{s} + \tilde{\tau}\tilde{\kappa})/(m_I + 1)$
end while
```

Proposition 2: The solution to (13) can be obtained by solving

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & \tilde{W}^{-1}\tilde{S} \end{bmatrix} \begin{bmatrix} f_1 & h_1 \\ f_2 & h_2 \\ f_3 & h_3 \end{bmatrix} = \begin{bmatrix} r_1 & -g \\ r_2 & -b \\ r_3 & -c \end{bmatrix}, \tag{14}$$

and subsequent computation of

$$\begin{aligned}
\Delta\tilde{\tau} &= \frac{r_6 - \tilde{\tau}(g^T f_1 + b^T f_2 + c^T f_3)}{\tilde{\kappa} + \tilde{\tau}(g^T h_1 + b^T h_2 + c^T h_3)}, \\
\Delta\tilde{t} &= f_1 + h_1 \Delta\tilde{\tau}, \\
\Delta\tilde{v} &= f_2 + h_2 \Delta\tilde{\tau}, \\
\Delta\tilde{w} &= f_3 + h_3 \Delta\tilde{\tau}, \\
\Delta\tilde{\kappa} &= g^T \Delta\tilde{t} + b^T \Delta\tilde{v} + c^T \Delta\tilde{w} - r_4, \\
\Delta\tilde{s} &= \tilde{W}^{-1}(r_C - \tilde{S}\Delta\tilde{w}),
\end{aligned}$$

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

Proof: See [30]. ■

Appendix VIII-A provides an efficient Riccati iteration procedure for the solution of (14). The proposed method has order of complexity $O(N(n_u + n_x + n_z)^3)$ per iteration. In comparison, the complexity of solving the system directly using sparse linear algebra routines is linear to quadratic in N, while a general purpose solver using dense linear algebra scales cubically [31]. We remark that 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 [21].

V. Special Operators

To speed-up the numerical computations and reduce the storage requirements of our algorithm, operations involving the structured matrices F and H are implemented as specialized linear algebra routines [32].

Denote the Lagrange multipliers associated with the inequality constraints (2d)- (2g) as η, λ, υ, ω, γ, ζ and ξ where

$$\eta := \begin{bmatrix} \eta_0^T & \eta_1^T & \dots & \eta_{N-1}^T \end{bmatrix}^T,$$

and similarly for λ, υ, γ, ζ and ξ. The multipliers (η, λ) are associated with the input limits (2d), (υ, ω) are associated with the input-rate limits (2e), (γ, ζ) are associated for the output limits (2f), and ξ is associated with the non-negative constraints (2g).

Using the notation above, we can write the optimization variables, $(\tilde{t}, \tilde{v}, \tilde{w})$, as

$$\begin{aligned}
\tilde{t} &= \begin{bmatrix} u_0^T & \hat{x}_1^T & \rho_1^T & u_1^T & \hat{x}_2^T & \rho_2^T & \dots & u_{N-1}^T & \hat{x}_N^T & \rho_N^T \end{bmatrix}^T, \\
\tilde{v} &= \begin{bmatrix} \tilde{v}_1^T & \tilde{v}_2^T & \dots & \tilde{v}_N^T \end{bmatrix}^T, \\
\tilde{w} &= \begin{bmatrix} \eta^T & \lambda^T & \upsilon^T & \omega^T & \gamma^T & \zeta^T & \xi^T \end{bmatrix}^T.
\end{aligned}$$

The rows of $F^T\tilde{v}$ and $H\tilde{w}$ are computed as

$$(F^T\tilde{v})_i = \begin{cases} B^T\tilde{v}_j, & i \in \mathbb{I}_1, \\ A^T\tilde{v}_{j+1} - \tilde{v}_j, & i \in \mathbb{I}_2, \\ 0, & i \in \mathbb{I}_4, \\ -\tilde{v}_{j-2}, & i \in \mathbb{I}_4, \end{cases}$$

where $\mathbb{I}_1 := 1, 4, \ldots, 3N-2$, $\mathbb{I}_2 := 2, 5, \ldots, 3N-4$, $\mathbb{I}_3 := 3, 6, \ldots, 3N$, $\mathbb{I}_4 := 3N-1$ and $j := \lfloor i/3 \rfloor + 1$. Moreover

$$(H^T\tilde{w})_i = \begin{cases} \eta_{j-1} - \lambda_{j-1} + \upsilon_{j-1} - \omega_{j-1}, & i \in \mathbb{I}_1, \\ D^T(\omega_j - \upsilon_j) + C_z^T(\gamma_j - \zeta_j), & i \in \mathbb{I}_2, \\ -\gamma_{j-1} - \zeta_{j-1} - \xi_{j-1}, & i \in \mathbb{I}_3, \\ C_z^T(\gamma_{j-2} - \zeta_{j-2}), & i \in \mathbb{I}_4, \end{cases}$$

where $j := \lfloor i/3 \rfloor + 1$. For $F\tilde{t}$ and $H\tilde{t}$ we have

$$(F\tilde{t})_i = \begin{cases} Bu_{i-1} - \hat{x}_i, & i = 1, \\ A\hat{x}_{i-1} + Bu_{i-1} - \hat{x}_i, & i = 2, 3, \ldots, N, \end{cases}$$

and

$$(Ht)_i = \begin{cases} u_j, & i = 1, \ldots, N, \\ -u_j, & i = N+1, \ldots, 2N, \\ u_0, & i = 2N+1, \\ -D\hat{x}_j + u_j, & i = 2N+2, \ldots, 3N, \\ -u_0, & i = 3N+1, \\ D\hat{x}_j - u_j, & i = 3N+2, \ldots, 4N, \\ C_z\hat{x}_{j+1} - \rho_{j+1}, & i = 4N+1, \ldots, 5N, \\ -C_z\hat{x}_{j+1} - \rho_{j+1}, & i = 5N+1, \ldots, 6N, \\ -\rho_{j+1}, & i = 6N+1, \ldots, 7N. \end{cases}$$

in which $j := (i-1) \mod N$.

VI.

We apply the warm-starting strategy from [19] to pick an initial point for Algorithm 1. The main idea is to combine a guess of the solution (candidate point) with a standard cold starting point $\phi^0 = (0, 0, \mathbf{1}_{m_I}, \mathbf{1}_{m_I}, 1, 1)$. An important feature of the homogenous and self-dual model is that this cold-starting point is perfectly centralized with respect to the central path [19].

The initial point 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 $(\bar{t},\bar{v},\bar{w},\bar{s})$ is the candidate point and $\lambda \in [0,1]$ is a tuning parameter. Notice that in case $\lambda = 0$, the initial point becomes the standard cold-starting point. Conversely, $\lambda = 1$ means that the initial solution becomes the candidate point. Since this point typically lies close to the boundary of the non-negative orthant, $(\tilde{w},\tilde{s},\tilde{\kappa},\tilde{\tau}) \geq 0$, $\lambda = 1$ can lead to ill-conditioned linear systems and/or blocking of the search direction [33].

In MPC applications, the optimal control problem is solved in a closed-loop fashion. Therefore, a good choice of the candidate point at time k can be constructed using the solution from the previous time step. As an example consider the solution of (3)-(4) at time step $k = 0$, for $N = 3$

$$t^* := \begin{bmatrix} u_0^{*T} & \hat{x}_1^{*T} & \rho_1^{*T} & u_1^{*T} & \hat{x}_2^{*T} & \rho_2^{*T} & u_2^{*T} & \hat{x}_3^{*T} & \rho_3^{*T} \end{bmatrix}^T .$$

In this case we use the following candidate point at time step $k = 1$

$$\bar{t} := \begin{bmatrix} u_1^{*T} & \hat{x}_2^{*T} & \rho_2^{*T} & u_2^{*T} & \hat{x}_3^{*T} & \rho_3^{*T} & u_2^{*T} & \hat{x}_3^{*T} & \rho_3^{*T} \end{bmatrix}^T .$$

Similarly, we left-shift the slack variables, s, and the dual variables v and w.

VII. 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.

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 [35]. For $i = 1, 2, \ldots, m$

$$Y_i(s) = \frac{1}{(\tau_i s + 1)^3}\left(U_i(s) + D_i(s)\right) + E_i(s). \tag{15}$$

$D_i(s)$ 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 simply the sum $Z(s) = \sum_{i=1}^m \frac{1}{(\tau_i s+1)^3}\left(U_i(s) + D_i(s)\right)$. We convert the 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, $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ is the measured power production, and $\boldsymbol{z}_k \in \mathbb{R}^{n_z}$ is the total power production. Note that $n_u = n_y = m$ and $n_z = 1$. It is assumed that $\boldsymbol{d}_k \sim N(0,\sigma I)$, and $\boldsymbol{e}_k \sim N(0,\sigma I)$.

TABLE I

CASE STUDY PARAMETERS

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

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 and control horizon to $N = 80$ time steps. 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 shows a number of box-plots used to tune 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, $(\tilde{w}, \tilde{s}, \tilde{\kappa}, \tilde{\tau}) \geq 0$. 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 depicts the computation time of 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 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 CPLEX. Notice that when the number of units is large, the optimization problem

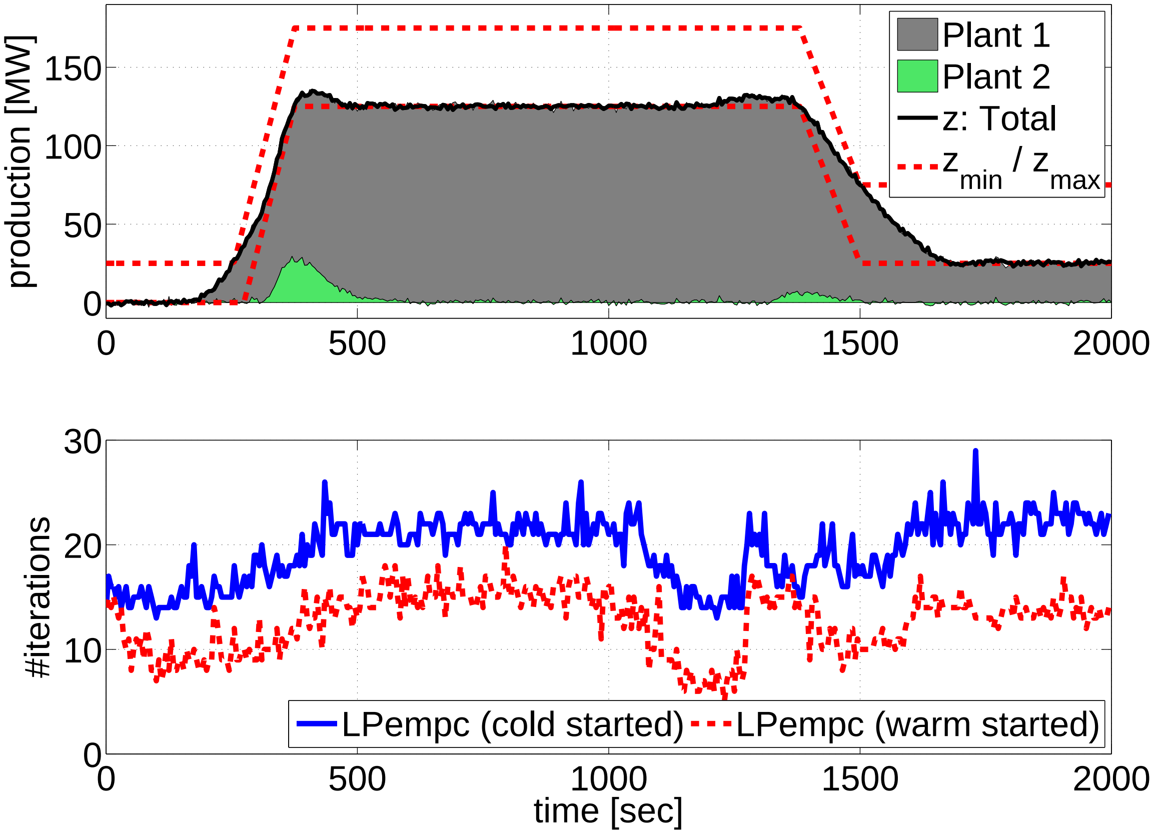


Fig. 1. Closed-loop simulation of a power system controlled by economic MPC. Warm-start ($\lambda = 0.99$) yields a significant reduction in the number of iterations.

(2) may be solved using Dantzig-Wolfe decomposition [37]. The Dantzig-Wolfe decomposition algorithm solves a number of subproblems that have the structure (2). These subproblems can be solved efficiently using `LPempc`.

VIII. Conclusions

In this paper, we have developed a computationally efficient IPM for linear economic MPC. The novelty of our algorithm is the combination of a 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 30%-35%, and that a MATLAB and C implementation of our algorithm, `LPempc`, is significantly faster than several state-of-the-art IPMs. 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

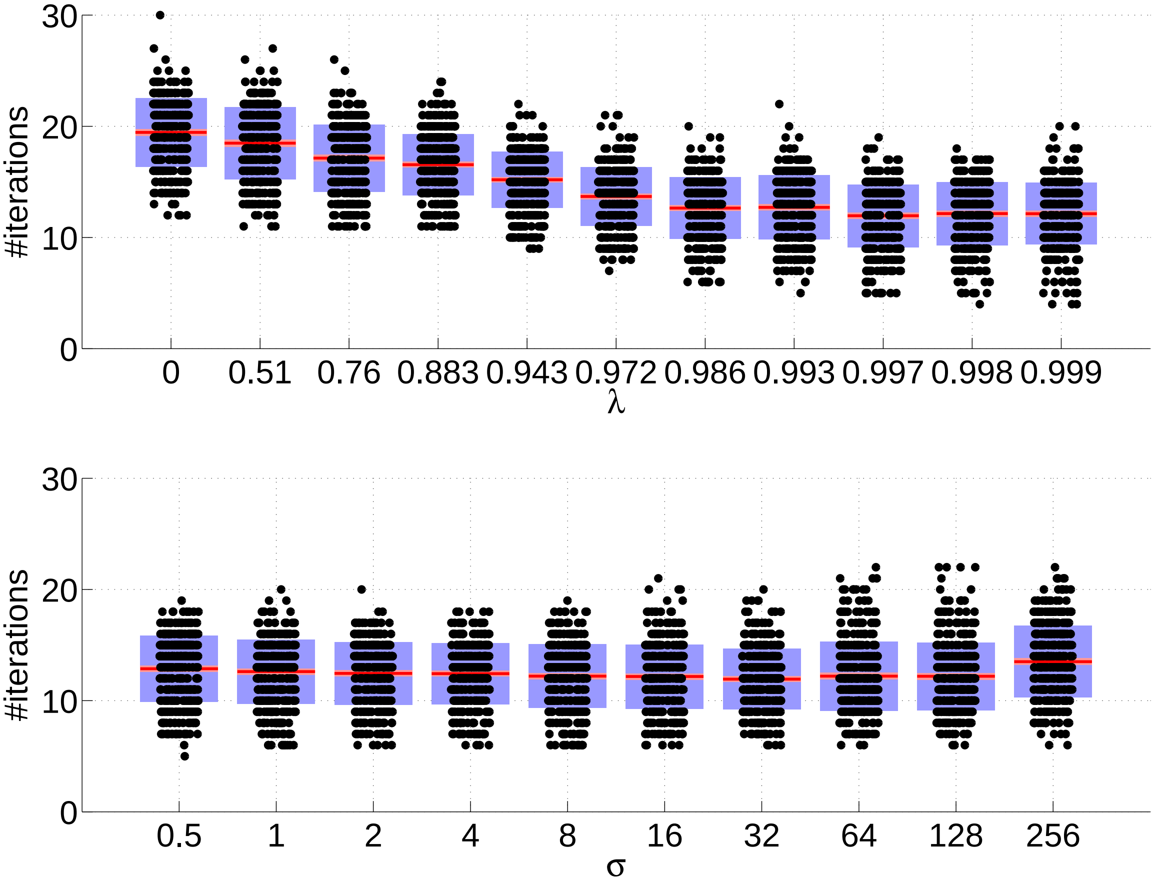


Fig. 2. Number of iterations as a function of the tuning parameter λ, and the noise parameter σ. Each box-plot has been generated based on an entire closed-loop simulation. In the top plot we have fixed $\sigma = 1$, and in the bottom plot $\lambda = 0.99$.

results also show that the difference in computing time becomes larger with the problem size, as `LPempc` scales in a favourable way.

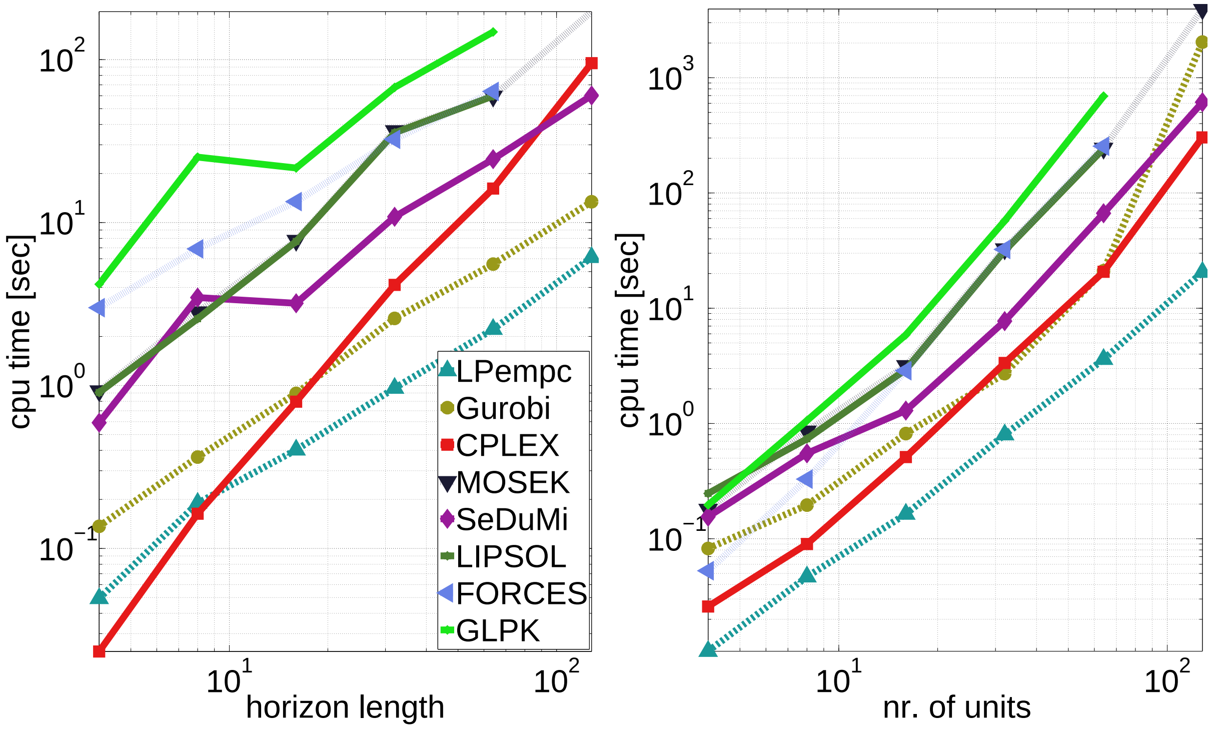
A. Riccati Iteration Procedure for Economic MPC

Consider the system (14)

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & \tilde{W}^{-1}\tilde{S} \end{bmatrix} \begin{bmatrix} f_1 & h_1 \\ f_2 & h_2 \\ f_3 & h_3 \end{bmatrix} = \begin{bmatrix} r_1 & -g \\ r_2 & -b \\ \hat{r}_3 & -c \end{bmatrix}.$$

For a single arbitrary right hand side, we may write this system as

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & \tilde{W}^{-1}\tilde{S} \end{bmatrix} \begin{bmatrix} \Delta\tilde{t} \\ \Delta\tilde{v} \\ \Delta\tilde{w} \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ r_I \end{bmatrix}. \tag{16}$$



(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.

Using the same notation as in Section V, we write the solution to (16) in the form

$$\Delta\tilde{t} = \begin{bmatrix}\Delta u_0^T & \Delta\hat{x}_1^T & \Delta\rho_1^T & \dots & \Delta u_{N-1}^T & \Delta\hat{x}_N^T & \Delta\rho_N^T\end{bmatrix}^T,$$

$$\Delta\tilde{v} = \begin{bmatrix}\Delta\tilde{v}_0^T & \Delta\tilde{v}_1^T & \dots & \Delta\tilde{v}_{N-1}^T\end{bmatrix}^T,$$

$$\Delta\tilde{w} = \begin{bmatrix}\Delta\eta^T & \Delta\lambda^T & \Delta\upsilon^T & \Delta\omega^T & \Delta\gamma^T & \Delta\zeta^T & \Delta\xi^T\end{bmatrix}^T.$$

Accordingly, we partition the right hand side such that

$$r_D = \begin{bmatrix}r_{u,0}^T & r_{x,1}^T & r_{w,1}^T & \dots & r_{u,N-1}^T & r_{x,N}^T & r_{w,N}^T\end{bmatrix}^T,$$

$$r_E = \begin{bmatrix}R_{v,0}^T & R_{v,1}^T & \dots & R_{v,N-1}^T\end{bmatrix}^T,$$

$$r_I = \begin{bmatrix}r_\eta^T & r_\lambda^T & r_\upsilon^T & r_\omega^T & r_\gamma^T & r_\zeta^T & r_\xi^T\end{bmatrix}^T,$$

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

$$\tilde{W}^{-1}\tilde{S} = \textbf{diag}\left(\Sigma_\eta^T, \Sigma_\lambda^T, \Sigma_\upsilon^T, \Sigma_\omega^T, \Sigma_\gamma^T, \Sigma_\zeta^T, \Sigma_\xi^T\right).$$

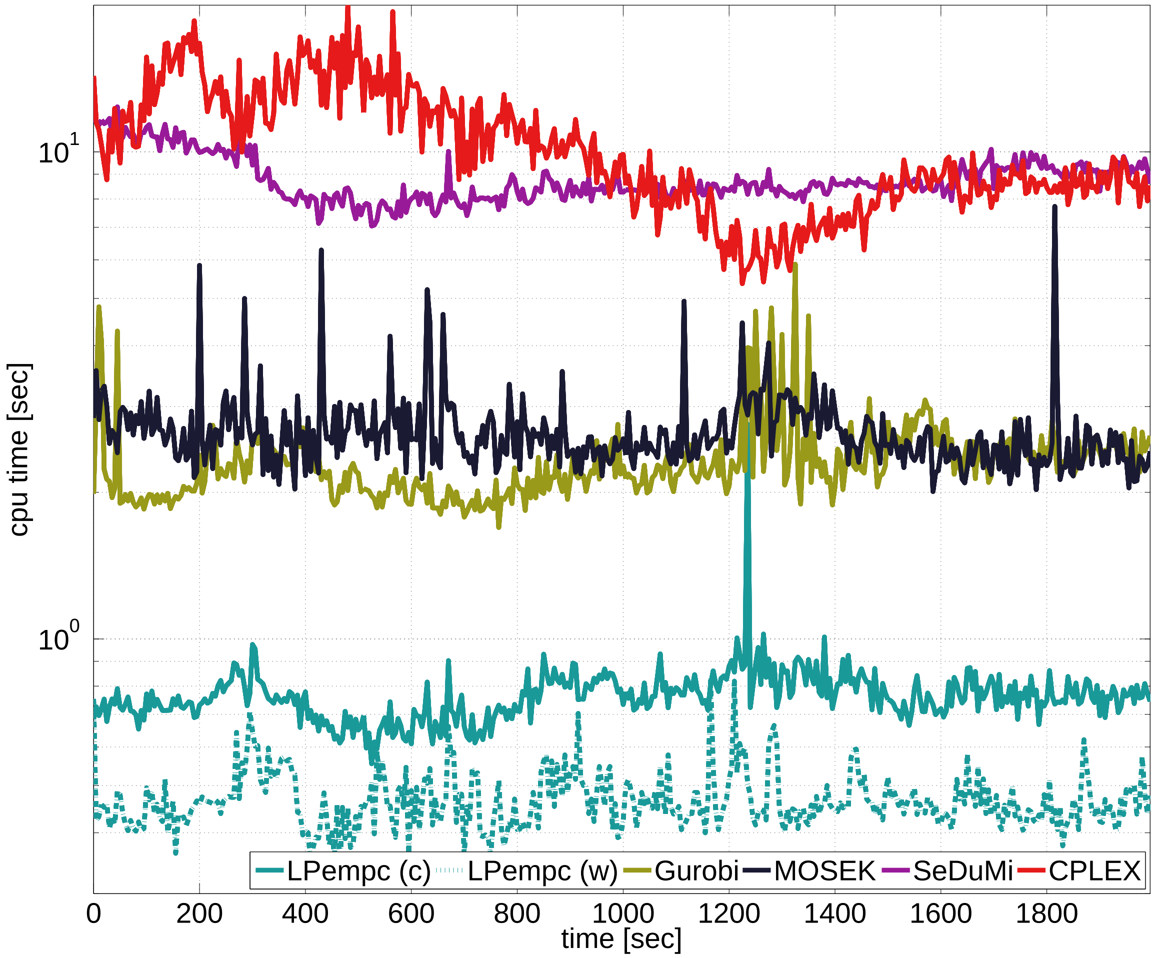


Fig. 4. CPU-time for solving (2) with 15 power generating units and a prediction horizon of 200 time steps. We observe that warm-starting reduces the average number of iterations by approximately 40%.

The linear system of equations (16) can now be stated in the form

$$
\begin{aligned}
\Delta\eta_i - \Delta\lambda_i + \Delta\upsilon_i - \Delta\omega_i + B^T\Delta\tilde{v}_i &= r_{u,i}, && i \in \mathcal{N}_0,\\
-\Delta u_i + \Sigma_{\eta,i}\Delta\eta_i &= r_{\eta,i}, && i \in \mathcal{N}_0,\\
\Delta u_i + \Sigma_{\lambda,i}\Delta\lambda_i &= r_{\lambda,i}, && i \in \mathcal{N}_0,\\
-\Delta u_i + D\Delta\hat{x}_i + \Sigma_{\upsilon,i}\Delta\upsilon_i &= r_{\upsilon,i}, && i \in \tilde{\mathcal{N}}_0,\\
\Delta u_i - D\Delta\hat{x}_i + \Sigma_{\omega,i}\Delta\omega_i &= r_{\omega,i}, && i \in \tilde{\mathcal{N}}_0,\\
\Delta\hat{x}_{i+1} - A\Delta\hat{x}_i - B\Delta u_i &= R_{\upsilon,i}, && i \in \tilde{\mathcal{N}}_0,\\
\Delta\rho_i - C_z\Delta\hat{x}_i + \Sigma_{\gamma,i}\Delta\gamma_i &= r_{\gamma,i}, && i \in \mathcal{N}_1,\\
\Delta\rho_i + C_z\Delta\hat{x}_i + \Sigma_{\zeta,i}\Delta\zeta_i &= r_{\zeta,i}, && i \in \mathcal{N}_1,\\
\Delta\rho_i + \Sigma_{\xi,i}\Delta\xi_i &= r_{\xi,i}, && i \in \mathcal{N}_1,\\
-\Delta\gamma_i - \Delta\zeta_i - \Delta\xi_i &= r_{w,i}, && i \in \mathcal{N}_1,\\
-\Delta\tilde{v}_i + C_z^T(\Delta\gamma_{i+1} - \Delta\zeta_{i+1}) + A^T\Delta\tilde{v}_i & && \\
+D^T(\omega_i - \Delta\upsilon_i) &= r_{x,i}, && i \in \tilde{\mathcal{N}}_0,
\end{aligned}
$$

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

$$\begin{aligned}
-\Delta u_0 + \Sigma_{\upsilon,0}\Delta\upsilon_0 &= r_{\upsilon,0},\\
\Delta u_0 + \Sigma_{\omega,0}\Delta\omega_0 &= r_{\omega,0},\\
\Delta\hat{x}_1 - B\Delta u_0 &= R_{v,0},\\
-\Delta\tilde{v}_{N-1} + C_z^T(\Delta\gamma_N - \Delta\zeta_N) &= r_{x,N}.
\end{aligned}$$

By eliminating the Lagrange multipliers for the inequality constrains $\Delta\eta$, $\Delta\lambda$, $\Delta\upsilon$, $\Delta\omega$, $\Delta\gamma$, $\Delta\zeta$ and $\Delta\xi$ we get the reduced set of equations

$$\begin{aligned}
B^T\Delta\tilde{v}_0 + U_0\Delta u_0 &= R_{u,0}, && && \text{(17a)}\\
B^T\Delta\tilde{v}_i + U_i\Delta u_i + G_i\Delta\hat{x}_i &= R_{u,i}, && i \in \tilde{\mathcal{N}}_0, && \text{(17b)}\\
-\Delta\hat{x}_1 + B\Delta u_0 &= R_{v,0}, && && \text{(17c)}\\
-\Delta\hat{x}_{i+1} + A\Delta\hat{x}_i + B\Delta u_i &= R_{v,i}, && i \in \tilde{\mathcal{N}}_0, && \text{(17d)}\\
W_i\Delta\rho_i + M_i^T\Delta\hat{x}_i &= R_{w,i}, && i \in \mathcal{N}_1, && \text{(17e)}\\
-\Delta\tilde{v}_{i-1} + M_i\Delta\rho_i + \bar{X}_i\Delta\hat{x}_i & && && \\
+G_i^T\Delta u_i + A^T\Delta\tilde{v}_i &= \bar{R}_{x,i}, && i \in \tilde{\mathcal{N}}_0, && \text{(17f)}\\
-\Delta\tilde{v}_{N-1} + M_N\Delta\rho_N + \bar{X}_N\Delta\hat{x}_N &= \bar{R}_{x,N}, && && \text{(17g)}
\end{aligned}$$

where we have defined

$$\begin{aligned}
U_i &:= \Sigma_{\eta,i}^{-1} + \Sigma_{\lambda,i}^{-1} + \Sigma_{\omega,i}^{-1} + \Sigma_{\upsilon,i}^{-1}, && i \in \mathcal{N}_0,\\
G_i &:= -(\Sigma_{\omega,i}^{-1} + \Sigma_{\upsilon,i}^{-1})D, && i \in \tilde{\mathcal{N}}_0,\\
W_i &:= \Sigma_{\zeta,i}^{-1} + \Sigma_{\xi,i}^{-1} + \Sigma_{\gamma,i}^{-1}, && i \in \mathcal{N}_1,\\
M_i &:= C_z^T(\Sigma_{\zeta,i}^{-1} - \Sigma_{\gamma,i}^{-1}), && i \in \mathcal{N}_1,\\
\bar{X}_i &:= C_z^T(\Sigma_{\zeta,i}^{-1} + \Sigma_{\upsilon,i}^{-1})C_z + D^T(\Sigma_{\gamma,i}^{-1} + \Sigma_{\omega,i}^{-1})D, && i \in \tilde{\mathcal{N}}_0,\\
\bar{X}_N &:= C_z^T(\Sigma_{\zeta,N}^{-1} + \Sigma_{\upsilon,N}^{-1})C_z.
\end{aligned}$$

Furthermore

$$\begin{aligned}
R_{u,i} &:= r_{u,i} + \bar{r}_{\lambda,i} + \bar{r}_{\omega,i} - \bar{r}_{\eta,i} - \bar{r}_{\upsilon,i}, && i \in \mathcal{N}_0,\\
R_{v,i} &:= -R_{v,i}, && i \in \mathcal{N}_0,\\
R_{w,i} &:= r_{w,i-1} + \bar{r}_{\zeta,i-1} + \bar{r}_{\xi,i} + \bar{r}_{\gamma,i}, && i \in \mathcal{N}_1,\\
\bar{R}_{x,i} &:= r_{x,i} + C_z^T(\bar{r}_{\zeta,i} - \bar{r}_{\gamma,i}) + D^T(\bar{r}_{\upsilon,i} - \bar{r}_{\omega,i}), && i \in \tilde{\mathcal{N}}_0,\\
\bar{R}_{x,N} &:= r_{x,N} + C_z^T(\bar{r}_{\zeta,N} - \bar{r}_{\gamma,N}).
\end{aligned}$$

For compact notation, we have introduced the notation $\bar{r}_{\lambda,i} := \Sigma_{\lambda,i}^{-1} r_{\lambda,i}$, and in a similar way we define $\bar{r}_{\omega,i}$, $\bar{r}_{\eta,i}$, $\bar{r}_{\upsilon,i}$, $\bar{r}_{\zeta,i}$, $\bar{r}_{\xi,i}$ and $\bar{r}_{\gamma,i}$. Solving (17e) for $\Delta\rho_i$ gives

$$\Delta\rho_i = W_i^{-1}(R_{w,i} - M_i^T \Delta\hat{x}_i), \qquad i \in \mathcal{N}_1. \tag{18}$$

Substituting back into (17) yields the equations

$$\begin{aligned}
B^T\Delta\tilde{v}_0 + U_0\Delta u_0 &= R_{u,0} \\
B^T\Delta\tilde{v}_i + U_i\Delta u_i + G_i\Delta\hat{x}_i &= R_{u,i}, && i \in \tilde{\mathcal{N}}_0 \\
-\Delta\hat{x}_1 + B\Delta u_0 &= R_{v,0} \\
-\Delta\hat{x}_{i+1} + A\Delta\hat{x}_i + B\Delta u_i &= R_{v,i}, && i \in \tilde{\mathcal{N}}_0 \\
-\Delta\tilde{v}_{i-1} + X_i\Delta\hat{x}_i + G_i^T\Delta u_i + A^T\Delta\tilde{v}_i &= R_{x,i}, && i \in \tilde{\mathcal{N}}_0 \\
-\Delta\tilde{v}_{N-1} + X_N\Delta\hat{x}_N &= R_{x,N}
\end{aligned}$$

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

$$\left[\begin{array}{cc|ccc|ccc|c}
U_0 & B^T & & & & & & & \\
B & & -I & & & & & & \\
\hline
 & -I & X_1 & G_1^T & A^T & & & & \\
 & & G_1 & U_1 & B^T & & & & \\
 & & A & B & & -I & & & \\
\hline
 & & & & -I & X_2 & G_2^T & A^T & \\
 & & & & & G_2 & U_2 & B^T & \\
 & & & & & A & B & & -I \\
\hline
 & & & & & & & -I & X_3
\end{array}\right]
\left[\begin{array}{c}
\Delta u_0 \\ \Delta\tilde{v}_0 \\ \hline \Delta\hat{x}_1 \\ \Delta u_1 \\ \Delta\tilde{v}_1 \\ \hline \Delta\hat{x}_2 \\ \Delta u_2 \\ \Delta\tilde{v}_2 \\ \hline \Delta\hat{x}_3
\end{array}\right]
=
\left[\begin{array}{c}
R_{u,0} \\ R_{v,0} \\ \hline R_{x,1} \\ R_{u,1} \\ R_{v,1} \\ \hline R_{x,2} \\ R_{u,2} \\ R_{v,2} \\ \hline R_{x,3}
\end{array}\right]$$

This system can be solved efficiently by a Riccati iteration procedure [20], [22]–[24].

References

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

[2] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[3] J. M. Maciejowski, *Predictive Control: With Constraints*, ser. Pearson Education. Prentice Hall, 2002.

[4] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems*, vol. 20, no. 3, pp. 38–52, 2000.

[5] 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.

[6] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, no. 3, pp. 725–734, 2013.

[7] J. B. Rawlings, D. Angeli, and C. N. Bates, "Fundamentals of economic model predictive control," in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, 2012, pp. 3851–3861.

[8] D. Angeli, R. Amrit, and J. B. Rawlings, "On Average Performance and Stability of Economic Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1615–1626, 2012.

[9] J. B. Rawlings, D. Bonne, J. B. Jørgensen, A. N. Venkat, and S. B. Jørgensen, "Unreachable Setpoints in Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.

[10] L. E. Sokoler, G. Frison, K. Edlund, A. Skajaa, and J. B. Jørgensen, "A Riccati Based Homogeneous and Self-Dual Interior-Point Method for Linear Economic Model Predictive Control," in *2013 IEEE Multi-conference on Systems and Control*, 2013, pp. 592–598.

[11] L. E. Sokoler, A. Skajaa, G. Frison, R. Halvgaard, and J. B. Jørgensen, "A Warm-Started Homogeneous and Self-Dual Interior-Point Method for Linear Economic Model Predictive Control," in *2013 IEEE 52th Annual Conference on Decision and Control (CDC)*, 2013, pp. 3677–3683.

[12] R. Amrit, J. B. Rawlings, and L. T. Biegler, "Optimizing process economics online using model predictive control," *Computers & Chemical Engineering*, vol. 58, no. 0, pp. 334–343, 2013.

[13] T. G. Hovgaard, L. F. S. Larsen, and J. B. Jørgensen, "Flexible and cost efficient power consumption using economic MPC a supermarket refrigeration benchmark," in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 848–854.

[14] J. Ma, S. J. Qin, B. Li, and T. Salsbury, "Economic model predictive control for building energy systems," in *2011 IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2011, pp. 1–6.

[15] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jørgensen, "Economic Model Predictive Control for building climate control in a Smart Grid," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2012, pp. 1–6.

[16] R. Halvgaard, N. K. Poulsen, H. Madsen, J. B. Jørgensen, F. Marra, and D. E. M. Bondy, "Electric vehicle charge planning using Economic Model Predictive Control," in *2012 IEEE International Electric Vehicle Conference (IEVC)*, 2012, pp. 1–6.

[17] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[18] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] J. B. Jørgensen, "Moving Horizon Estimation and Control," Ph.D. Thesis, Department of Chemical Engineering, Technical University of Denmark, 2005.

[24] 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.

[25] H. Frenk, *High Performance Optimization*, ser. Applied Optimization. Springer, 2000.

[26] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] K. Edlund, L. E. Sokoler, and J. B. Jørgensen, "A primal-dual interior-point linear programming algorithm for MPC," in *48th IEEE Conference on Decision and Control, held jointly with 28th Chinese Control Conference*, 2009, pp. 351–356.

[33] J. Gondzio and A. Grothey, "A New Unblocking Technique to Warmstart Interior Point Methods Based on Sensitivity Analysis," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1184–1210, 2008.

[34] A. Domahidi, A. Zgraggen, 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.

[35] 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.

[36] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[37] L. E. Sokoler, L. Standardi, K. Edlund, N. K. Poulsen, H. Madsen, and J. B. Jørgensen, "A Dantzig-Wolfe decomposition algorithm for linear economic model predictive control of dynamically decoupled subsystems," *Journal of Process Control*, vol. 24, no. 8, pp. 1225–1236, 2014.