A Dantzig-Wolfe Decomposition Algorithm for Linear Economic Model Predictive Control of Dynamically Decoupled Subsystems

L.E. Sokoler\textsuperscript{a,b}, L. Standardi\textsuperscript{a}, K. Edlund\textsuperscript{b}, N.K. Poulsen\textsuperscript{a}, H. Madsen\textsuperscript{a}, J.B. Jørgensen\textsuperscript{*a}

\textsuperscript{a}Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
\textsuperscript{b}DONG Energy, DK-2820 Gentofte, Denmark

Abstract

This paper presents a warm-started Dantzig-Wolfe decomposition algorithm tailored to economic model predictive control of dynamically decoupled subsystems. We formulate the constrained optimal control problem solved at each sampling instant as a linear program with state space constraints, input limits, input rate limits, and soft output limits. The objective function of the linear program is related directly to the cost of operating the subsystems, and the cost of violating the soft output constraints. Simulations for large-scale economic power dispatch problems show that the proposed algorithm is favorable over both state-of-the-art linear programming solvers, and a structure exploiting implementation of the alternating direction method of multipliers. It is also demonstrated that the control strategy presented in this paper can be tuned using a weighted $\ell_1$-regularization term. In the presence of process and measurement noise, such a regularization term is critical for achieving the desired closed-loop performance.

Key words: Optimization, Dantzig-Wolfe decomposition, Regularization, Linear programming, Distributed model predictive control, Energy management

1. Introduction

Conventionally, the optimal control problem (OCP) solved in model predictive control (MPC) is formulated as a convex program that penalizes deviations between the controlled output and a setpoint [1–4]. While this approach ensures that the setpoint is reached in a reasonable amount of time, it does not guarantee that the transition between setpoints is performed in an economically efficient way. To overcome this problem, MPC has been extended to solve OCPs with more general cost functions, providing a systematic method for optimizing economic performance [5–11]. Stability and other properties of such economic MPC (EMPC) schemes have been addressed in [5–9, 12].

The main contribution of this paper is a Dantzig-Wolfe decomposition algorithm for control of dynamically decoupled subsystems that solves the OCP in an efficient and reliable way. As the control law is computed in real-time, such an algorithm allows EMPC to be employed even for applications with thousands of subsystems. In particular, we consider an $\ell_1$-regularized linear type of OCP with input constraints, input rate constraints and soft output constraints. Each subsystem is governed by a discrete state space model. The coupling of the subsystems occurs through a set of aggregated variables.

The Dantzig-Wolfe decomposition algorithm, presented in this paper, exploits that dynamically decoupled subsystems give rise to a block-angular structure in the OCP constraint matrix. This allows the OCP to be decomposed into a master problem and a number of subproblems [13–15]. The master problem includes a set of linking constraints which couples the subsystems, whereas the subproblems are concerned only with the individual subsystems. Using an iterative approach illustrated in Fig. 1, the decomposed problem can be solved via a delayed column generation procedure. Such techniques have previously been applied to conventional norm-based MPC in [16–18].

The block-angular constraint matrix structure appears for dy-
namically decoupled subsystems with linking constraints [19]. Dynamic multi-plant models as well as dynamic multi-product models are examples of such models [20]. Dynamic multi-plant models occur e.g. in the production planning for multiple refineries [21]. For process systems, dynamically decoupled systems with linking constraints occur when independent units are connected to shared process equipment such as pipes. A Boiler-turbine system producing high pressure (HP), middle pressure (MP) and low pressure (LP) steam as well as electricity is a common example of a system that can be modeled as dynamically decoupled subsystems (the boilers) that have linking constraints (the demand for various steam qualities and electrical power) [16, 22]. In upstream offshore oil production, the compressors and pumps of a number of production wells share the pipeline, separators and compressors to bring the oil onshore [23, 24]. This is also an example of a system that can be modeled as dynamically decoupled subsystems with linking constraints. Smart Grid systems in which a number of independently energy producers and consumers are controlled to balance power represent yet another instance of dynamically decoupled subsystems management case study. We show that as the aggregated variable limits, \( \Delta x_{j,k} \), the states by \( \sum_{j,k} \Delta u_{j,k} \), and that the performance of this strategy increases with the weights on the \( \ell_1 \)-regularization term. Inclusion of the regularization term is critical for the controller performance in the face of stochastic process and measurement noise as well as model-plant mismatch.

\[ x_{j,k+1} = A_j x_{j,k} + B_j u_{j,k}, \quad j \in M, \quad (1a) \]
\[ y_{j,k} = C_j x_{j,k}, \quad j \in M, \quad (1b) \]

where \( M = \{1, 2, \ldots, M\} \). The state space matrices are denoted by \( (A_j, B_j, C_j) \), the states by \( x_{j,k} \in \mathbb{R}^{n_j(l)} \), the inputs by \( u_{j,k} \in \mathbb{R}^{n_{j}(l)} \), and the outputs by \( y_{j,k} \in \mathbb{R}^{n_{j}(l)} \). Moreover, we define the aggregated variables

\[ y_{r,k} = \sum_{j \in M} T_j' y_{j,k} = \sum_{j \in M} T_j' C_j x_{j,k}, \quad (2) \]

in which \( T_j \in \mathbb{R}^{n_{r}(l) \times n_{j}(l)} \) are subsystem multipliers.

The OCP defining the EMPc control law for the subsystems (1), is in this paper defined as

\[ \min_{u_{j,k}, y_{r,k}, \psi} \psi = \psi_{eco} + \psi_{reg}, \quad (3a) \]

with

\[ \psi_{eco} = \sum_{k \in N_0} \left( q_{r,k} y_{r,k+1} + \sum_{j \in M} p_{j,k} u_{j,k} + r_{j,k+1} y_{j,k+1} \right), \quad (3b) \]
\[ \psi_{reg} = \sum_{k \in N_0, j \in M} w_{j,k} \| \Delta u_{j,k} \|_1, \quad (3c) \]

and subject to the constraints

\[ x_{j,k+1} = A_j x_{j,k} + B_j u_{j,k}, \quad k \in N_0, j \in M, \quad (3d) \]
\[ y_{j,k} = C_j x_{j,k}, \quad k \in N_1, j \in M, \quad (3e) \]
\[ y_{r,k} = \sum_{j \in M} T_j' C_j x_{j,k}, \quad k \in N_1, \quad (3f) \]
\[ u_{j,k} \leq u_{j,k} \leq \bar{u}_{j,k}, \quad k \in N_0, j \in M, \quad (3g) \]
\[ \Delta u_{j,k} = \Delta u_{j,k} \leq \bar{u}_{j,k}, \quad k \in N_0, j \in M, \quad (3h) \]
\[ y_{j,k} = \gamma_{j,k} \leq y_{j,k} + \gamma_{j,k}, \quad k \in N_1, j \in M, \quad (3i) \]
\[ 0 \leq \gamma_{j,k} \leq \bar{\gamma}_{j,k}, \quad k \in N_1, j \in M, \quad (3j) \]
\[ \gamma_{r,k} = \rho_{k} \leq \gamma_{r,k} \leq \bar{\gamma}_{r,k}, \quad k \in N_1, \quad (3k) \]
\[ 0 \leq \rho_{k} \leq \bar{\rho}_{k}, \quad k \in N_1. \quad (3l) \]

The input rate is defined as \( \Delta u_{i,k} = u_{i,k} - u_{i,k-1} \) and \( N_i = \{0 + i, 1 + i, \ldots, N - 1 + i\} \) with \( N \) being the length of the prediction horizon.

The input data to (3) are the input limits, \( (\underline{u}_{j,k}, \bar{u}_{j,k}) \), the input rate limits, \( (\Delta u_{j,k}, \bar{u}_{j,k}) \), the subsystem output limits, \( (\gamma_{j,k}, \bar{\gamma}_{j,k}) \), the aggregated variable limits, \( (\gamma_{r,k}, \bar{\gamma}_{r,k}) \), the input prices, \( p_{j,k} \), the price for violating the subsystem output limits, \( r_{j,k} \), and the performance indicators for the proposed algorithms. These performance indicators are computed using a conceptual energy systems management case study. Concluding remarks are given in Section 6.

2. Problem Definition

We consider \( M \) dynamically decoupled discrete state space models in the form
price for violating the aggregated variable limits \( q_k \). The slack variables \( y_{jk} \) and \( \rho_k \) account for the violation of the soft output constraints. We impose upper limits, \( (\overline{y}_{jk}, \overline{\rho}_{jk}) \), on these variables, as this simplifies later computations considerably.

The objective function (3a) consists of an economic term (3b) and a regularization term (3c). The economic term (3b) represents the cost of operating the subsystems and the cost of violating the soft output constraints. The regularization term (3c) is included to obtain a well-behaved solution. In our paper, the regularization term is formulated as a weighted \( \ell_1 \)-penalty on the input rate. Using an \( \ell_1 \)-penalty ensures that the resulting OCP is a linear program that can be solved using Dantzig-Wolfe decomposition.

**Remark 1.** An alternative way of expressing the OCP objective function, (3a), is as a trade-off between the economic term and the regularization term, such that

\[
\psi = \alpha \psi_{ec} + (1 - \alpha) \psi_{reg}, \quad \alpha \in [0, 1],
\]

where \( \alpha \) is a user-defined parameter. [12] discusses the trade-off between the economic term and an \( \ell_1 \)-regularization term.

The regularization term (3c) is a special case of

\[
\psi_{reg} = \sum_{k \in \mathcal{N}_0} \left( \sum_{j \in \mathcal{M}} w_{jk}^\ell ||x_{jk+1} - \bar{x}_{jk+1}||_1 + \sum_{j \in \mathcal{M}} w_{jk}^\rho ||u_{jk} - \bar{u}_{jk}||_1 + w_{jk}^\gamma ||\Delta u_{jk}||_1 \right),
\]

in which \( \{x_{jk+1}, \bar{u}_{jk}\}_{k \in \mathcal{N}_0, j \in \mathcal{M}} \) are target values that may be computed by a target calculator or a real-time optimization layer. An objective function consisting only of (5) corresponds to conventional \( \ell_1 \) norm-based MPC. [18] solves such problems using Dantzig-Wolfe decomposition.

**Remark 2.** The objective function (4) can be used as an approximation to a mean-variance-based objective function, where the economic term (3b) is a certainty-equivalent approximation to the mean of the cost function, and the regularization term (5) approximates the variance. [31] reports a mean-variance approach to EMPC for production optimization in an oil field. The key advantage in using the deterministic formulation (3b) and (5), is that the computational load is significantly reduced compared to a mean-variance approach based on Monte Carlo simulations. Other measures of risk than the mean-variance formulation that can be used to regularize the solution are Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) [32].

### 2.1. Compact Formulation

By eliminating the states using equation (1a), we can write the output equation, (1b), as

\[
y_{jk} = C_j^\gamma x_{j0} + \sum_{i \in \mathcal{N}_0} H_{jk-i} u_{ji}, \quad j \in \mathcal{M}, k \in \mathcal{N}_1,
\]

where the impulse response coefficients are given by

\[
H_{jk} = C_j A_j^{k-1} B_j, \quad j \in \mathcal{M}, k \in \mathcal{N}_1.
\]

Consequently

\[
y_{Tk} = \sum_{j \in \mathcal{M}} \left( T_j C_j A_j^T x_{j0} + \sum_{i \in \mathcal{N}_0} T_j H_{jk-i} u_{ji} \right), \quad k \in \mathcal{N}_1.
\]

Define the vectors

\[
y_j = \begin{bmatrix} y_{j1}^T & y_{j2}^T & \cdots & y_{jN}^T \end{bmatrix}^T, \quad j \in \mathcal{M}, \tag{6a}
\]

\[
u_j = \begin{bmatrix} u_{j0}^T & u_{j1}^T & \cdots & u_{jN-1}^T \end{bmatrix}^T, \quad j \in \mathcal{M}, \tag{6b}
\]

and the matrices

\[
\Gamma_j = \begin{bmatrix} H_{j1} & 0 & \cdots & 0 \\
H_{j2} & H_{j1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
H_{jN} & H_{jN-1} & \cdots & H_{j1} \end{bmatrix}, \quad \Phi_j = \begin{bmatrix} C_j A_j^1 \\
C_j A_j^2 \\
\vdots \\
C_j A_j^{N-1} \end{bmatrix},
\]

for \( j \in \mathcal{M} \). Then, for each of the subsystems (6a)

\[
y_j = \Gamma_j \nu_j + \Phi_j y_{j0}, \quad j \in \mathcal{M}. \tag{7}
\]

By introducing \( \tilde{\Gamma}_j \) and \( \tilde{\Phi}_j \) accordingly, it follows that \( y_T = \sum_{j \in \mathcal{M}} \tilde{\Gamma}_j \nu_j + \tilde{\Phi}_j y_{j0} \).

The notation is simplified further with

\[
u_j = \begin{bmatrix} u_{j0}^T & u_{j1}^T & \cdots & u_{jN-1}^T \end{bmatrix}^T, \quad j \in \mathcal{M},
\]

\[
\tilde{\nu}_j = \begin{bmatrix} \bar{u}_{j0}^T & \bar{u}_{j1}^T & \cdots & \bar{u}_{jN-1}^T \end{bmatrix}^T, \quad j \in \mathcal{M},
\]

and similarly we define \( \Delta \nu_j, \Delta \tilde{\nu}_j, \overline{y}_j, \overline{\gamma}_j, \overline{\nu}_j, \bar{\psi}, \bar{\theta}_j, \bar{\rho}, \bar{\eta}_j, \bar{\eta}_j, q, p_j, r_j, \overline{\psi}_j \). Using these definitions, the OCP, (3), may be written in the form

\[
\min_{\nu_j, \gamma_j, \eta_j} \psi = q^T \rho + \sum_{j \in \mathcal{M}} (\nu_j + (\gamma_j + \bar{\psi}_j \eta_j), \tag{8a}
\]

subject to a set of decoupled constraints

\[
u_j \leq \overline{\nu}_j, \quad j \in \mathcal{M}, \tag{8b}
\]

\[
\Delta \nu_j - \Delta \overline{\nu}_j \leq \Delta \nu_j \leq \Delta \overline{\nu}_j - \Delta \overline{\nu}_j, \quad j \in \mathcal{M}, \tag{8c}
\]

\[
\overline{y}_j - \gamma_j \leq \Gamma_j \nu_j + \Phi_j y_{j0} \leq \overline{y}_j + \gamma_j, \quad j \in \mathcal{M}, \tag{8d}
\]

\[
\overline{\nu}_j - \eta_j \leq \Delta \nu_j \leq \overline{\nu}_j + \eta_j, \quad j \in \mathcal{M}, \tag{8e}
\]

\[
0 \leq \eta_j \leq \tilde{\eta}_j, \quad j \in \mathcal{M}, \tag{8f}
\]

\[
0 \leq \gamma_j \leq \overline{\gamma}_j, \quad j \in \mathcal{M}, \tag{8g}
\]

and a set of linking constraints

\[
\overline{y}_j - \rho \leq \sum_{j \in \mathcal{M}} \tilde{\Gamma}_j \nu_j + \tilde{\Phi}_j y_{j0} \leq \overline{y}_j + \rho, \tag{8g}
\]

\[
0 \leq \rho \leq \bar{\rho}, \tag{8i}
\]
Figure 2: The block-angular structure of the constraint matrix in (8). The efficiency of the Dantzig-Wolfe decomposition method depends on this structure.

where \( \Lambda \) and \( I_0 \) are defined as

\[
\Lambda_j = \begin{bmatrix}
I & I \\
-I & I \\
\vdots & \vdots \\
-I & I \\
\end{bmatrix}, \quad I_0 = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
\end{bmatrix}.
\]

We remark that (8e)-(8f) imply that \( \eta_{jk} \geq |\Delta u_{jk}| \). Note also, that the structure of the constraint matrix in (8), can be stated in the block-angular form illustrated in Fig. 2.

In particular, (8) is written as

\[
\begin{aligned}
\min_z \psi &= \sum_{j \in \mathcal{M}} c_j^T z_j, \\
\text{s.t. } G_j z_j &\geq g_j, \quad j \in \mathcal{M}, \\
\sum_{j \in \mathcal{M}} H_j z_j &\geq h,
\end{aligned}
\tag{9a}
\]

\[
\begin{aligned}
G_j &= \begin{bmatrix}
\tilde{G}_j \\
-G_j
\end{bmatrix}, \\
g_j &= \begin{bmatrix}
\tilde{g}_j \\
-\tilde{g}_j
\end{bmatrix}, \\
H_j &= \begin{bmatrix}
\tilde{H}_j \\
-\tilde{H}_j
\end{bmatrix}, \\
h &= \begin{bmatrix}
h \\
-h
\end{bmatrix},
\end{aligned}
\tag{9b}
\]

in which

\[
\begin{bmatrix}
\bar{G}_j \\
\bar{g}_j
\end{bmatrix} = \begin{bmatrix}
I & 0 & 0 & u_j \\
\Lambda & 0 & 0 & \Delta u_j \\
\Gamma_j & I & 0 & \tilde{y}_j \\
0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix}
\tilde{y}_j \\
\tilde{y}_j
\end{bmatrix},
\]

\[
\begin{bmatrix}
\bar{H}_j \\
\bar{h}
\end{bmatrix} = \begin{bmatrix}
\tilde{H}_j \\
0 \end{bmatrix} \begin{bmatrix}
0 & y_T & -\infty & \tilde{y}_T
\end{bmatrix}.
\]

for \( j \in \mathcal{M} \), with

\[
\begin{aligned}
y_T &= y_T - \sum_{j \in \mathcal{M}} \Phi_j x_{j,0}, \\
\tilde{y}_T &= \tilde{y}_T - \sum_{j \in \mathcal{M}} \Phi_j x_{j,0}, \\
\gamma_j &= y_j - \Phi_j x_{j,0}, \\
\tilde{y}_j &= \tilde{y}_j - \Phi_j x_{j,0}, \\
\Delta u_j &= \Delta u_j + I_0 u_{j,-1}, \\
\tilde{\Delta} u_j &= \tilde{\Delta} u_j + I_0 u_{j,-1}.
\end{aligned}
\]

In the special case \( j = M + 1 \)

\[
\begin{bmatrix}
\bar{G}_{M+1} \\
\bar{g}_{M+1} \\
\bar{G}_{M+1}
\end{bmatrix} = \begin{bmatrix}
I \\
0 \\
\bar{P}
\end{bmatrix},
\]

\[
\begin{bmatrix}
\bar{H}_{M+1} \\
\bar{h} \\
\bar{H}_{M+1}
\end{bmatrix} = \begin{bmatrix}
I & 0 & \bar{P}
\end{bmatrix}.
\]

Remark 3. We only use (9) to have a convenient notation. In the actual solution of all the linear and quadratic programs reported in this paper, the bound constraints are exploited.

3. Dantzig-Wolfe Decomposition

Dantzig-Wolfe decomposition utilizes the fact that a convex set can be characterized by its extreme points and its extreme rays [13–15]. In particular, for each \( j \in \mathcal{M} \), the set of points satisfying the decoupled constraints (9b) may be written as

\[
\mathcal{G}_j = \{ z_j | G_j z_j \geq g_j \},
\]

\[
= \{ z_j | \sum_{j \in \mathcal{P}} \lambda_j^i z_j \geq \sum_{j \in \mathcal{P}} \lambda_j^i \geq 0 \quad \forall i \in \mathcal{P} \},
\]

where \( z_j^i \) are the extreme points of \( \mathcal{G}_j \), and \( \lambda_j^i \) are convex combination multipliers. Note that since each of the sets \( \mathcal{G}_j \) are bounded, extreme rays are not needed in their representation.

By replacing the decision variables in (9) by convex combination multipliers, we obtain the master problem formulation

\[
\begin{aligned}
\min_{\lambda} \psi &= \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{P}} c_j^T \lambda_j^i, \\
\text{s.t. } \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{P}} H_j^i \lambda_j^i &\geq h, \\
\sum_{i \in \mathcal{P}} \lambda_j^i &\geq 1, \quad j \in \mathcal{M}, \\
\lambda_j^i &\geq 0, \quad j \in \mathcal{M}, \quad i \in \mathcal{P}.
\end{aligned}
\tag{10a-d}
\]
where we have defined
\[ H_j^i = H_{j^i_p}, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (11a) \]
\[ c_j^i = c_{j^i_p}^i, \quad j \in \tilde{M}, \quad i \in \mathcal{P}. \quad (11b) \]

Given a solution, \( \lambda^* \), to the master problem (10), a solution to the original problem, (9) can be obtained as
\[ z_j^* = \sum_{i \in \mathcal{P}} (\lambda_j^i)^T c_j^i, \quad j \in \tilde{M}. \]

The number of extreme points, \( |\mathcal{P}| \), can increase exponentially with the size of the original problem. In such cases, it is computationally inefficient to solve the master problem directly. In the following section, we overcome this issue by employing a column generation procedure that replaces \( \mathcal{P} \) by a subset \( \tilde{\mathcal{P}} \).

### 3.1. Column Generation Procedure

The dual linear program of (10) may be stated as
\[ \max_{\alpha, \beta} \phi = \alpha^T h + \sum_{j \in \tilde{M}} \beta_j, \quad (12a) \]
\[ \text{s.t.} \quad (H_j^i)^T \alpha + \beta_j \leq c_j^i, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (12b) \]
\[ \alpha \geq 0, \quad (12c) \]

in which \( \alpha \in \mathbb{R}^\tilde{M} \) and \( \beta \in \mathbb{R}^{\tilde{M}+1} \) are the Lagrange multipliers associated with the linking constraints, (10b), and the convexity constraints, (10c), respectively.

The necessary and sufficient optimality conditions for (10) and (12) are
\[ \sum_{j \in \tilde{M}} \sum_{i \in \mathcal{P}} H_j^i \lambda_j^i \geq h, \quad (13a) \]
\[ \sum_{i \in \mathcal{P}} \lambda_j^i = 1, \quad j \in \tilde{M}, \quad (13b) \]
\[ \lambda_j^i \geq 0, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (13c) \]
\[ c_j^i - (H_j^i)^T \alpha - \beta_j \geq 0, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (13d) \]
\[ \alpha \geq 0, \quad (13e) \]
\[ \lambda_j^i (c_j^i - (H_j^i)^T \alpha - \beta_j) = 0, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (13f) \]

In Proposition 1 we derive conditions for which a solution satisfying this set of optimality conditions can be obtained by solving the master problem (10), over a subset of the original variables.

**Proposition 1.** Let \( \tilde{\mathcal{P}} \subseteq \mathcal{P} \) for all \( j \in \tilde{M} \), and define \( (\lambda^*, \alpha^*, \beta^*) \) as a primal-dual solution to (10) and (12) restricted to the subset \( \tilde{\mathcal{P}} \). Then the solution
\[ \alpha^* = \alpha, \quad (14a) \]
\[ \beta_j^* = \beta_j, \quad j \in \tilde{M}, \quad (14b) \]
\[ (\lambda^* j)^i = \begin{cases} \lambda_j^i & \text{if } i \in \tilde{\mathcal{P}} \\ 0 & \text{if } i \in \mathcal{P} \setminus \tilde{\mathcal{P}} \end{cases}, \quad j \in \tilde{M}, \quad i \in \mathcal{P}, \quad (14c) \]
satisfies the conditions, (13), if the optimal objective value of the subproblem
\[ \min_{z_j^i} \varphi_j = (c_j - H_j^i \alpha^*)^T z_j^i - \beta_j^* \quad (14a) \]
\[ \text{s.t.} \quad G_j z_j^i \geq g_j, \quad (14b) \]
is non-negative for each \( j \in \tilde{M} \).

**Proof.** The solution \( (\lambda^*, \alpha^*, \beta^*) \) satisfies (13a) since
\[ \sum_{j \in \tilde{M}} \sum_{i \in \mathcal{P}} H_j^i (\lambda^* j)^i = \sum_{j \in \tilde{M}} \sum_{i \in \mathcal{P}} H_j^i \lambda_j^i \geq h, \]
which follows from the definition of \( (\lambda, \alpha, \beta) \). Similarly, it can be verified that the conditions (13b), (13c), (13e) and (13f) are fulfilled.

**Remark 4.** The problem (14) is an OCP with linear constraints and a linear objective function. [10, 11, 33] and [34] provide an efficient Riccati-based homogeneous and self-dual interior-point linear programming algorithm for such problems. Using the optimal interior point solution found by this algorithm, crossover methods can be applied to obtain an optimal extreme point for the column generation procedure [15].

Algorithm 1 summarizes a column generation procedure based on Theorem 10.

### 3.2. Warm-Start

A sequence of closely related OCPs are solved in a moving horizon implementation of EMPC. Therefore, in Algorithm 1 the feasible initial guess of the solution, \( z_j^i \), at the current sampling instant is constructed from the solution at the previous sampling instant.

Given the solution to (14)
\[ u_j^i = [u_j^{T,0} \ldots u_j^{T,N-1}]^T, \quad j \in \tilde{M}, \]
\[ y_j^i = [y_j^{T,0} \ldots y_j^{T,N-1}]^T, \quad j \in \tilde{M}, \]
\[ \eta_j^i = [\eta_j^{T,0} \ldots \eta_j^{T,N-1}]^T, \quad j \in \tilde{M}, \]
\[ \rho_j^i = [\rho_j^{T,0} \ldots \rho_j^{T,N-1}]^T, \]
we construct an initial point for the following sampling instant as
\[ z_j^0 = [u_j^{T,0}^T \ldots \eta_j^{T,0}^T \rho_j^{T,0}^T]^T, \quad j \in \tilde{M}. \]
Algorithm 1 Column generation procedure for solution of (10).

Require: \( (i_{\text{max}}, \varepsilon), \{l_j^i\}_{j \in \mathcal{M}} \)

\( i = 0, \) converged = false

while not converged and \( i < i_{\text{max}} \) do

\[ \mathcal{P} = \{0, 1, \ldots, i\} \]

compute problem data

for each \( j \in \mathcal{M} \) do

\[ H_j = H_j \new \]

\[ c_j = c_j \new \]

end for

solve restricted master problem

\((\varphi^*, \chi^*, \alpha^*, \beta^*) \leftarrow \) solve (10) with \( \mathcal{P} = \mathcal{P} \)

solve subproblems

for each \( j \in \mathcal{M} \) do

\( (\varphi_j^*, \chi_j^*) \leftarrow \) solve (14)

end for

check if converged

if \( |\varphi_j| \geq \varepsilon \forall j \in \mathcal{M} \) then

converged = true

end if

update extreme points

for each \( j \in \mathcal{M} \) do

\[ z_j^{i+1} = \varphi_j^* \]

\( i = i + 1 \)

end for

end if
while

\[ u_0^j = \begin{bmatrix} u_1^T & \cdots & u_{j-1}^T & (a_{j, 0})^T \end{bmatrix}^T, \quad j \in \mathcal{M}, \]

\[ y_0^j = \begin{bmatrix} \gamma_1^T & \cdots & \gamma_{j-1}^T & (y_{j, 0})^T \end{bmatrix}^T, \quad j \in \mathcal{M}, \]

\[ \eta_0^j = \begin{bmatrix} \eta_1^T & \cdots & \eta_{j-1}^T & (\eta_{j, 0})^T \end{bmatrix}^T, \quad j \in \mathcal{M}. \]

Finally

\[ z_{M+1}^0 = \rho_0 = \begin{bmatrix} \rho_1^T & \cdots & \rho_{j, N+1}^T \end{bmatrix}^T. \]

The original solution values are thus shifted forward in time, and \( y_0^j, y_{j+1}^0, \eta_1^0, \) and \( \rho_{N+1}^0 \) are appended to the resulting initial point.

In our implementation

\[ u_0^j = u_{j-1}^*, \quad j \in \mathcal{M}, \]

\[ \eta_{j, N}^0 = 0, \quad j \in \mathcal{M}. \]

We use the state space equations (1) and (2) to compute \( y_{j+1}^0 \) and \( y_{j+1}^0 \) associated with this input sequence. We then construct the initial slack values as

\[ y_{j+1}^0 = \max(y_{j+1}^0 - y_{j+1}^0, 0) + \max(0 - y_{j+1}^0, 0), \]

for each \( j \in \mathcal{M} \), and

\[ \rho_{N+1}^0 = \max(y_{N+1}^0 - y_{N+1}^0, 0) + \max(0 - y_{N+1}^0, 0). \]

As the solution to the OCP often only differs slightly between successive sampling instants, the initial point generated as above provides a warm-start for Algorithm 1.

3.3. Cold-Start

In the case that no previous solution is available for generating a warm start, a feasible initial guess of the solution, \( \{a_{j, 0}\}_{j \in \mathcal{M}} \), in Algorithm 1 can be constructed by adjusting the slack variables, \( \gamma_j^0 \) and \( \rho_j^0 \). Let \( \{a_{j, 0}\}_{j \in \mathcal{M}} \) be feasible with respect to the input constraints and the input-rate constraints. Such a point is easily obtained in practice. As an example consider \( u_0^j = u_j \) for each \( j \in \mathcal{M} \). Then, in a similar way as for the warm-start strategy, we compute

\[ \gamma_{j, k}^0 = \max(y_{j, k} - y_{j, k+1}, 0) + \max(0 - y_{j, k} - y_{j, k+1}, 0), \]

\[ \rho_{j, k}^0 = \max(y_{j, k} - y_{j, k+1}, 0) + \max(y_{j, k} - y_{j, k+1}, 0), \]

where \( k \in \mathcal{N}_t, \ j \in \mathcal{M} \). The values, \( \gamma_j^0 \) and \( \rho_j^0 \), are the sub-system outputs and the aggregated variables associated with the inputs, \( \{a_{j, 0}\}_{j \in \mathcal{M}} \), computed via (1) and (2). Finally, \( \eta_j^0 = \Delta u_j^0 \) for each \( j \in \mathcal{M} \).

4. The Alternating Direction Method of Multipliers

ADMM has been demonstrated as a powerful algorithm for solving large-scale structured convex optimization problems [27], including a range of OCPs arising in MPC applications [28–30]. In this section, we present a distributed ADMM scheme for solving the OCP (9). We refer to [27] for details and proofs.

To solve (9) via ADMM, we first introduce the auxiliary variables \( v_j \), and write the OCP as

\[ \min_{\psi} \psi = \sum_{j \in \mathcal{M}} c_j^T v_j, \]

s.t. \( G_j z_j \geq g_j, \quad j \in \mathcal{M}, \]

\( H_j v_j = v_j, \quad j \in \mathcal{M}, \]

\[ \sum_{j \in \mathcal{M}} v_j \geq h, \]

Using indicator functions, this problem can be stated in the standard ADMM form [27]

\[ \min_{\psi} \psi = \sum_{j \in \mathcal{M}} \left(c_j^T z_j + I_{Z_j}(z_j) + I_{\mathcal{V}}(v)\right), \quad (16a) \]

s.t. \( H_j z_j = v_j, \quad j \in \mathcal{M} \) (16b)

where \( Z_j = \{z_j | G_j z_j \geq g_j\}, \mathcal{V} = \{v | \sum_{j \in \mathcal{M}} v_j \geq h\} \), and \( I_{\mathcal{A}} \) is the indicator function of a set \( \mathcal{A} \) defined as

\[ I_{\mathcal{A}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{A}, \\ \infty & \text{otherwise.} \end{cases} \]
For the problem (16), the ADMM recursions described in [27] can be written as:

\[ z_{j+1}^{i} = \arg \min_{z_{j}} c_{j}^{T} z_{j} + \frac{\rho}{2} \| H_{j} z_{j} - v_{j}^{i} + u_{j}^{i} \|^{2}, \quad j \in \mathcal{M}, \tag{17a} \]

\[ v_{j+1}^{i} = \arg \min_{v_{j}} \frac{\rho}{2} \sum_{j \in \mathcal{M}} \| H_{j} z_{j}^{i+1} - v_{j} + u_{j}^{i} \|^{2}, \quad j \in \mathcal{M}, \tag{17b} \]

\[ u_{j+1}^{i} = u_{j}^{i} + H_{j} z_{j}^{i+1} - v_{j}^{i+1}, \quad j \in \mathcal{M}, \tag{17c} \]

where \( u^{i} \) is a scaled dual variable.

The update of \( z_{j} \), (17a), thus consists of solving the constrained quadratic program

\[ \min_{z_{j}} \frac{\rho}{2} H_{j} z_{j}^{T} H_{j} z_{j} + (c_{j} + \rho(-v_{j}^{i} + u_{j}^{i}))^{T} H_{j} z_{j}, \tag{18a} \]

s.t. \( G_{j} z_{j} \geq g_{j} \),

for each \( j \in \mathcal{M} \).

The update for \( v_{j} \), (17b), yields the explicit solution

\[ v_{j}^{i+1} = H_{j} z_{j}^{i+1} + u_{j}^{i} + \max(l/(M + 1), 0), \quad j \in \mathcal{M}, \]

where \( l = h - \sum_{j \in \mathcal{M}} H_{j} z_{j}^{i+1} + u_{j}^{i} \). Each subsystem can thus perform its own update of \( z_{j} \). Having computed \( l \) with a contribution from all the subsystems, \( v_{j} \) and \( u_{j} \) can be determined individually as well.

Algorithm 2 provides an overview of the ADMM steps described above. Under mild assumptions, the ADMM algorithm converges with a linear convergence rate to the optimal solution of the OCP [27, 35]. Note that we have replaced \( H_{j} z_{j}^{i+1} \) with \( \alpha H_{j} z_{j}^{i+1} - (1 - \alpha)(-v_{j}^{i}) + u_{j}^{i} \) in the recursions for \( v_{j} \) and \( u_{j} \). As described in [27, 36], such a relaxation often speeds up convergence. The relaxation parameter \( \alpha \in [0, 2] \) is tuned to the particular application.

To detect an optimal solution in Algorithm 2, we have adopted the stopping criteria proposed in [27]. For the specific problem formulation, (16), these criteria can be written as

\[ \| r_{j}^{i+1} \|^{2} \leq \epsilon_{p}, \quad \| s_{j}^{i+1} \|^{2} \leq \epsilon_{d}, \]

in which

\[ s_{j}^{i+1} = -\rho H_{j}^{T}(v_{j}^{i+1} - v_{j}), \quad r_{j}^{i+1} = \rho H_{j}^{T} z_{j}^{i+1} - v_{j}^{i+1}, \]

measure the primal and dual residual. These stopping criteria may be extended to include a relative measure as well [27].

As for the Dantzig-Wolfe decomposition algorithm, a warm start for Algorithm 2 can be constructed by shifting the closed-loop solution values, \( v^{*} \) and \( u^{*} \), forward in time. If such a solution is not available, the standard cold-starting point \((v^{0}, u^{0}) = (0, 0)\) is used. We remark that in comparison to the Dantzig-Wolfe decomposition algorithm, the initial point does not need to be feasible. Moreover, the extensions of Algorithm 2 are not restricted only to linear programming [28–30]. One could consider more general regularization terms in (4), e.g. \( \ell_{2} \)-regularization terms.

**Remark 5.** The optimization problem (18) is an OCP with a quadratic cost function and linear constraints. Efficient algorithms for such structured QPs include active-set methods [37–39], interior-point methods [10, 40–45] and first-order methods [44, 46].

---

**Algorithm 2 ADMM algorithm for the solution of (9)**

Require: \((\rho, \alpha, l_{\text{max}}, \epsilon_{p}, \epsilon_{d}), (v^{0}, u^{0}), (\epsilon_{p}, \epsilon_{d})\)

\(i = 0\), converged = false

while converged = false and \(i < l_{\text{max}}\) do

UPDATE VARIABLES

for \( j \in \mathcal{M} \) do

\[ z_{j}^{i+1} = \arg \min_{z_{j}} c_{j}^{T} z_{j} + \frac{\rho}{2} \| H_{j} z_{j} - v_{j}^{i} + u_{j}^{i} \|^{2}, \quad j \in \mathcal{M}, \]

end for

COMPUTE RESIDUALS

for \( j \in \mathcal{M} \) do

\[ r_{j}^{i+1} = \rho H_{j}^{T}(v_{j}^{i+1} - v_{j}), \quad s_{j}^{i+1} = -\rho H_{j}^{T} z_{j}^{i+1} - v_{j}^{i+1}, \]

end for

CHECK IF CONVERGED

if \( \| r_{j}^{i+1} \|_{2} \leq \epsilon_{p} \) and \( \| s_{j}^{i+1} \|_{2} \leq \epsilon_{d} \) then

converged = true

end if

\( i \leftarrow i + 1 \)

end while

---

5. Smart Energy Systems Case Study

To handle the volatile and unpredictable power generation associated with technologies such as wind, solar and wave power, energy systems management has emerged as a promising application area for EMPC. In EMPC of energy systems, the power production planning is handled in real-time by computing an optimal production plan based on the most recent information available such as forecasts of energy prices, wind power production, and district heating consumption [47–52].

In this section we use a conceptual energy systems management case to test a MATLAB implementation of Algorithm 1, Dwempc, and a MATLAB implementation of Algorithm 2, ADMMempc. The energy system considered, consists of a collection of power generating units in the form

\[ Y_{j}(s) = \frac{1}{(\tau_{j,s} + 1)^{b}} (U_{j}(s) + D_{j}(s)) + E_{j}(s), \quad j \in \mathcal{M}, \]  \( \tag{19} \)

where \( D_{j}(s) \) is the process noise, \( E_{j}(s) \) is the measurement noise, \( U_{j}(s) \) is the input (power production setpoint) to the \( j \)th power unit and \( Y_{j}(s) \) is its power production. The third order model, (19), has been validated against actual measurement data in [53]. This system is a dynamic multi-plant system. Fig. 3 illustrates the Dantzig-Wolfe decomposition algorithm for a dynamic multi-plant system.

To represent different types of power generating units, we vary the time constants, \( \tau_{j} \); values in the range 40–80 are associated with slow units such as centralized thermal power plants, while values in the range 20–40 represent units with faster dynamics such as diesel generators and gas turbines.
Figure 3: EMPC diagram for the Dantzig-Wolfe decomposition algorithm for a dynamic multi-plant system with linking constraints.

In the case study, the controller must compute the most cost-efficient feasible power setpoint for each power generating unit such that the total power production satisfies the time-varying power demand.

The total power produced by the \( M \) generating units is

\[
Y_T(s) = \frac{1}{\tau_j + 1} \left( U_j(s) + D_j(s) \right).
\]

Using a discrete state space representation, (19)-(20) may be expressed as

\[
\begin{align*}
  x_{jk+1} &= A_{jk}x_{jk} + B_{jk}u_{jk} + E_{jk}d_{jk}, & j \in M, \\
  y_{jk} &= C_{jk}x_{jk} + e_{jk}, & j \in M, \\
  y_{\tau k} &= \sum_{j \in M} C_{jk}x_{jk}.
\end{align*}
\]

In the resulting model structure, \( u_{jk} \in \mathbb{R} \) is the unit input (power setpoint), \( y_{jk} \in \mathbb{R} \) is the unit power production, and \( y_{\tau k} \in \mathbb{R} \) is the total power production. We assume that \( x_{j0} \sim N(x_{j0}, P_{j0}) \), \( d_{jk} \sim N(0, R_{j0}) \), and that \( e_{jk} \sim N(0, \sigma^2 I) \). By employing the Kalman filter, the separation principle, and the certainty equivalence principle, the OCP in EMPC for (21) can be stated in the form (3) with \( \Upsilon_j = 1 \) for all \( j \in M \), see e.g. [33].

5.1. Suboptimality Measure

The Dantzig-Wolfe decomposition algorithm and the ADMM algorithm satisfy the subsystem constraints, (9b), in every iteration. Therefore, a set of feasible but not necessarily optimal inputs, \( \bar{u}_j \), is available for the power generating units at each iteration of the algorithms. Consequently, the algorithms may be terminated early and still provide a feasible suboptimal solution. Using (8), we can compute the cost associated with the suboptimal inputs as

\[
\hat{\psi} = \check{q}^T \hat{\rho} + \sum_{j \in M} p_j^T \bar{u}_j + r_j^T \hat{\gamma}_j + w_j^T \hat{\eta}_j,
\]

where \( \hat{\rho}, \hat{\gamma}_j, \) and \( \hat{\eta}_j \) are completely determined by \( \bar{u}_j \). Based on \( \hat{\psi} \) and the optimal value \( \psi^* \), we define the level of suboptimality as

\[
\omega = 100 \frac{\hat{\psi} - \psi^*}{\max(|\psi^*|, 1)}.
\]

This definition of suboptimality provides a quality measure of the current available inputs.

Remark 6. In Dantzig-Wolfe decomposition, the solution to the restricted master problem, (10), provides an upper bound on the optimal objective value. Moreover, a lower bound can be determined without much extra work via the Lagrangian relaxation techniques described in [54]. Therefore, a bound on (22) can be computed in each iteration of Algorithm 1.

5.2. Simulation Parameters

In the simulations presented below, the prediction and control horizon is \( N = 60 \) time steps, and a sampling time of \( T_s = 5 \) seconds is used. Each generating unit is represented by a system in the form (19) with a time constant, \( \tau_j \), sampled from the uniform distribution over the interval \([20, 80]\). For simplicity, it is assumed that \( d_{jk} \sim N(0, (10\pi)^2 I) \), \( e_{jk} \sim N(0, \sigma^2 I) \), and that full initial state information is given such that \( x_{j0} \sim (0, 0) \).

The power generating unit input price is \( p_{jk} = 1/\tau_j \). This implies that fast units are more expensive to use than slow units. The conflict between response time and operating costs represents a common situation in the power industry: 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.

We define the input limits and the input rate limits as

\[
(u_{jk}, \bar{u}_{jk}, \Delta u_{jk}, \Delta u_{\tau k}) = (0, 8/M, -M/4, M/4).
\]

In this way the possible contribution from each unit to the overall power production diminishes as the number of units is increased. Local output constraints in the form (3i)-(3j) are not present. The local output variables, \( y_{jk} \), and the local slack variables, \( y_{\tau k} \), are thus excluded from the optimization problem.

The penalty for not satisfying the electricity demand, (3k), is fixed to \( \rho_k = 10 \). For ADMMmpc, we use the algorithm parameters \( \rho = 1 \) and \( \alpha = 1.8 \). These parameters have been carefully tuned to this particular application. The tolerance parameter for D\textsuperscript{max}mpc is set to \( \varepsilon = 1e-4 \). ADMMmpc uses the following primal and dual tolerance specifications: \( \varepsilon_p = \varepsilon_D = 1e-2 \). Both D\textsuperscript{max}mpc and ADMMmpc use CPLEX for solving the subproblems. Although the subproblems are solved sequentially, we refer to their effective CPU time in this paper, assuming that the subproblems are solved in parallel. The reason for this is to report the full potential of the distributed optimization algorithms.

5.3. Closed-Loop Simulations

We first consider an example with \( M = 2 \) power generating units. Table 1 lists the system and controller parameters.

Fig. 4 illustrates closed-loop simulations for different values of the noise parameter, \( \sigma \), and the regularization weights,
is up to approximately 30% suboptimal when cold-started, and not more than than 5% suboptimal when warm-started. Hence, although the number of iterations only decreases slightly when DWempc is warm-started, the quality of the solution obtained after terminating early improves significantly. By the same token, warm-start reduces the level of suboptimality for ADMMempc by several orders of magnitude.

Provided that the number of iterations is small, the effort per iteration is approximately equal for DWempc and ADMMempc. Table 2 reports that ADMMempc requires many more iterations than DWempc. Accordingly, we expect DWempc to provide a more accurate solution than ADMMempc within the same time frame. This hypothesis is confirmed by Fig. 5. Note however, that the computing time per iteration is constant for ADMMempc, while each iteration of DWempc requires an increasing work-load since extreme points are added to the master problem, (10), on the fly. Nonetheless, in all our simulations DWempc outperforms ADMMempc by a significant margin.

Fig. 6 depicts the level of suboptimality as a function of the CPU time including both DWempc and ADMMempc, for a single instance of the OCP with 128 generating units. Initially, ADMMempc finds the best solution. The quality of this solution is however far from optimal, making it economically very inefficient. For DWempc, fast convergence is observed after 0.2

---

**Table 1: Case study simulation and controller parameters.**

<table>
<thead>
<tr>
<th>Generating Unit</th>
<th>( \tau_i )</th>
<th>( p_k )</th>
<th>( u_k )</th>
<th>( \bar{u}_k )</th>
<th>( \Delta u_k )</th>
<th>( \Delta \bar{u}_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65</td>
<td>1/65</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>1/75</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

As indicated by Fig. 4(b), the closed-loop input variance increases significantly if no penalty is imposed on the input rate. This happens even for small values of the noise parameter. By assigning a penalty to the input-rate, the solution becomes more well behaved and better suited for practical applications. Table 2 shows that the addition of regularization also reduces the computing time for DWempc as well as ADMMempc. E.g. for scenario \( \sigma = 0.01 \) and \( w = 0.1 \), the average number of iterations performed by DWempc is reduced by more than 40% compared to the case without regularization, i.e. the case with \( w = 0 \). Also observe that while warm-start only leads to a marginal improvement in the iteration count for DWempc, a substantial reduction in the number of iterations is achieved for ADMMempc.

Fig. 5 shows the level of suboptimality, \( \omega \), computed via (22), for scenario \( \sigma \) when the run time of DWempc and ADMMempc is limited to 0.01 seconds. We observe that DWempc
Table 2: Iteration information table for the closed-loop simulation scenarios depicted in Fig. 4. The minimum, maximum and average number of iterations is listed for both cold start and for warm start (in parentheses).

<table>
<thead>
<tr>
<th>σ</th>
<th>w</th>
<th>DWempc</th>
<th>ADMMempc</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>0</td>
<td>[6(2), 16(17), 12(11)]</td>
<td>[47(2), 485(410), 097(66)]</td>
</tr>
<tr>
<td>s2</td>
<td>0</td>
<td>[6(2), 15(18), 10(09)]</td>
<td>[35(3), 469(410), 088(56)]</td>
</tr>
<tr>
<td>s3</td>
<td>0</td>
<td>[2(2), 17(18), 10(09)]</td>
<td>[33(6), 359(280), 149(48)]</td>
</tr>
<tr>
<td>s4</td>
<td>0.01</td>
<td>[7(2), 18(19), 13(11)]</td>
<td>[47(2), 485(410), 094(65)]</td>
</tr>
<tr>
<td>s5</td>
<td>0.01</td>
<td>[5(2), 13(16), 10(09)]</td>
<td>[35(2), 469(410), 088(56)]</td>
</tr>
<tr>
<td>s6</td>
<td>0.01</td>
<td>[5(2), 13(16), 10(09)]</td>
<td>[32(6), 359(280), 145(49)]</td>
</tr>
<tr>
<td>s7</td>
<td>0.1</td>
<td>[7(2), 17(20), 12(11)]</td>
<td>[46(2), 485(410), 091(66)]</td>
</tr>
<tr>
<td>s8</td>
<td>0.1</td>
<td>[6(2), 17(16), 09(09)]</td>
<td>[35(2), 469(410), 084(60)]</td>
</tr>
<tr>
<td>s9</td>
<td>0.1</td>
<td>[5(2), 14(14), 07(06)]</td>
<td>[32(6), 359(279), 144(47)]</td>
</tr>
</tbody>
</table>

Fig. 5: Suboptimality level of the closed-loop solution obtained by DWempc and ADMMempc when terminated after 0.01 seconds.

Figure 6: Level of suboptimality as a function of the CPU time, for a single instance of the OCP with 128 generating units.

Table 3: Tolerance specifications for DWempc and ADMMempc.

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>ε</th>
<th>εp</th>
<th>εD</th>
</tr>
</thead>
<tbody>
<tr>
<td>High (h)</td>
<td>1e-6</td>
<td>1e-4</td>
<td>1e-4</td>
</tr>
<tr>
<td>Medium (m)</td>
<td>1e-5</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td>Low (l)</td>
<td>1e-4</td>
<td>1e-2</td>
<td>1e-2</td>
</tr>
</tbody>
</table>

5.4. Large-Scale Simulations

We compare the performance of the algorithms presented in this paper to the performance of Gurobi, CPLEX and MOSEK. These state-of-the-art linear programming solvers are invoked via a MEX interface in MATLAB. We use their default tolerance settings. The algorithms are run on an Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz with 16 GB RAM running a 64-bit Windows 8.1 Pro operating system. For each solver, the computation time of solving the OCP (3) is reported as a function of the number of generating units. Table 3 lists the different accuracy settings used by DWempc and ADMMempc in our benchmarks.

Fig. 7 and Table 4 report the CPU time of solving the OCP for different number of generating units and optimization algorithms. For large problems, ADMMempc does not converge to high accuracy solutions within a reasonable amount of time. Therefore, Table 4 is incomplete.

For large problems, DWempc is faster than all other solvers tested in our case study. Observe also that Gurobi, CPLEX and MOSEK perform almost as well as DWempc in terms of CPU time. For high accuracy solutions, DWempc is 2 times faster than CPLEX and 5 times faster than Gurobi. DWempc and ADMMempc...
can easily accommodate very large problems in memory while Gurobi, CPLEX and MOSEK fail due to insufficient memory. The threshold when memory becomes an issue is around \( M = 3000 \) generating units. Consequently, when considering both CPU time and memory requirements, \( \text{DWempc} \) is an attractive optimization algorithm for large scale dynamically decoupled energy management problems.

Note from Fig. 6 that \( \text{ADMMempc} \) needs many more iterations to converge than \( \text{DWempc} \) for the high accuracy tolerance specification, (h). Table 5 further shows that the number of iterations increases with the problem size for \( \text{ADMMempc} \). Therefore, \( \text{ADMMempc} \) is less attractive from a scalability point of view. Apparently, the number of iterations used by \( \text{DWempc} \) does not depend on the number of generating units, \( M \).

Table 6 lists the suboptimality level of the solution determined by \( \text{DWempc} \) and \( \text{ADMMempc} \) for different values of \( M \). As observed from Table 6, \( \text{DWempc} \) is not only faster than \( \text{ADMMempc} \) for the tolerance specifications listed in Table 3, but the solution accuracy is also significantly better.

### 6. Conclusions

In this paper, we developed and presented a warm-started possibly early terminated Dantzig-Wolfe decomposition algorithm for \( \ell_1 \)-regularized linear EMPC of dynamically decoupled subsystems. Simulations show that a MATLAB implementation of the proposed algorithm, denoted \( \text{DWempc} \), is faster than CPLEX, Gurobi and MOSEK, as well as an implementation of ADMM denoted \( \text{ADMMempc} \). Both \( \text{DWempc} \) and \( \text{ADMMempc} \) have similar parallelization capabilities. They are able to handle much larger problems than the general purpose solvers. The simulations also demonstrate that in combination with warm-start, early termination of \( \text{DWempc} \) yields a highly accurate solution after only a few iterations. In contrast to \( \text{ADMMempc} \), the number of iterations required by \( \text{DWempc} \) to achieve a certain tolerance level does not grow with the problem size.

For cases when the number of \( \text{DWempc} \) iterations is large, \( \text{DWempc} \) may be slower than \( \text{ADMMempc} \). The reason is that the computing time per iteration of \( \text{DWempc} \) grows with the iteration number. Conversely, the time spent per iteration by \( \text{ADMMempc} \) is constant. Although this is a potential drawback of the Dantzig-Wolfe decomposition algorithm that favors the ADMM algorithm, we have not observed this being the case in any of our simulations. In all our simulations, \( \text{DWempc} \) outperforms \( \text{ADMMempc} \); in some cases by several orders of magnitude.

### References

10. L. E. Sokoler, G. Frison, K. Edlund, A. Skajaa, J. B. Jørgensen, A Riccati Based Homogeneous and Self-Dual Interior-Point Method for Linear
Table 5: The number of iterations performed by $\text{DWempc}$ and $\text{ADMMempc}$ in solving (3) for an increasing number of generating units, $M$.

<table>
<thead>
<tr>
<th>solver/M</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{DWempc}(1)$</td>
<td>9.8e-1</td>
<td>1.86e-1</td>
<td>3.54e-1</td>
<td>9.56e-1</td>
<td>9.54e-1</td>
<td>3.18e-1</td>
<td>3.13e-1</td>
<td></td>
</tr>
<tr>
<td>$\text{DWempc}(m)$</td>
<td>1.66e-5</td>
<td>3.87e-3</td>
<td>7.40e-3</td>
<td>1.94e-2</td>
<td>4.74e-2</td>
<td>5.29e-2</td>
<td>1.63e-1</td>
<td>9.37e-1</td>
</tr>
<tr>
<td>$\text{DWempc}(h)$</td>
<td>2.46e-10</td>
<td>1.13e-9</td>
<td>6.46e-6</td>
<td>1.98e-4</td>
<td>7.55e-4</td>
<td>1.06e-3</td>
<td>2.32e-3</td>
<td>4.82e-2</td>
</tr>
<tr>
<td>$\text{ADMMempc}(1)$</td>
<td>4.56e-1</td>
<td>6.65e-1</td>
<td>7.92e1</td>
<td>2.09e2</td>
<td>2.84e2</td>
<td>5.50e2</td>
<td>8.13e2</td>
<td>1.33e3</td>
</tr>
<tr>
<td>$\text{ADMMempc}(m)$</td>
<td>8.16e-2</td>
<td>8.36e-2</td>
<td>9.97</td>
<td>1.38e1</td>
<td>1.53e1</td>
<td>6.62e1</td>
<td>6.96e1</td>
<td></td>
</tr>
<tr>
<td>$\text{ADMMempc}(h)$</td>
<td>1.87e-1</td>
<td>1.86e-2</td>
<td>3.36e-5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6: Suboptimality level of $\text{DWempc}$ and $\text{ADMMempc}$ in the solution of (3) for an increasing number of generating units, $M$.

<table>
<thead>
<tr>
<th>solver/M</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{DWempc}(1)$</td>
<td>2.08e-2</td>
<td>1.67e-2</td>
<td>1.53e-1</td>
<td>1.86e-1</td>
<td>9.65e-1</td>
<td>9.44e-1</td>
<td>3.18e1</td>
<td>3.13e-1</td>
</tr>
<tr>
<td>$\text{DWempc}(m)$</td>
<td>1.94e-2</td>
<td>5.38e-3</td>
<td>7.40e-3</td>
<td>1.94e-2</td>
<td>4.74e-2</td>
<td>5.29e-2</td>
<td>1.63e-1</td>
<td>9.37e-1</td>
</tr>
<tr>
<td>$\text{DWempc}(h)$</td>
<td>2.46e-10</td>
<td>1.13e-9</td>
<td>6.46e-6</td>
<td>1.98e-4</td>
<td>7.55e-4</td>
<td>1.06e-3</td>
<td>2.32e-3</td>
<td>4.82e-2</td>
</tr>
<tr>
<td>$\text{ADMMempc}(1)$</td>
<td>4.56e-1</td>
<td>6.65e-1</td>
<td>7.92e1</td>
<td>2.09e2</td>
<td>2.84e2</td>
<td>5.50e2</td>
<td>8.13e2</td>
<td>1.33e3</td>
</tr>
<tr>
<td>$\text{ADMMempc}(m)$</td>
<td>8.16e-2</td>
<td>8.36e-2</td>
<td>9.97</td>
<td>1.38e1</td>
<td>1.53e1</td>
<td>6.62e1</td>
<td>6.96e1</td>
<td></td>
</tr>
<tr>
<td>$\text{ADMMempc}(h)$</td>
<td>1.87e-1</td>
<td>1.86e-2</td>
<td>3.36e-5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>


