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

L.E. Sokoler[a,b], L. Standardi[a], K. Edlund[b], N.K. Poulsen[a], H. Madsen[a], J.B. Jørgensen[*,a]

[a] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*
[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 ℓ_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 ℓ_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].

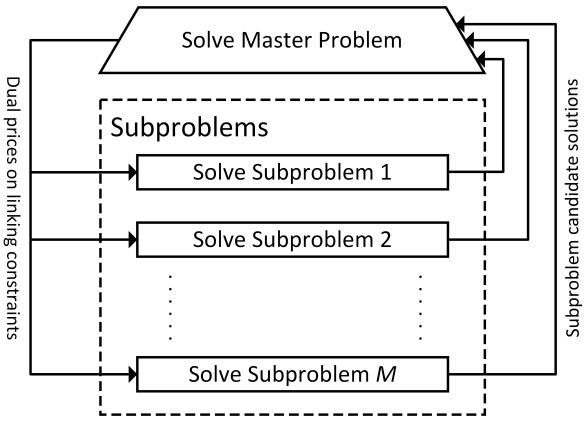


Figure 1: Flowchart of the delayed column generation procedure used in the Dantzig-Wolfe algorithm for EMPC of dynamically decoupled subsystems with linking constraints. In each iteration, dual prices associated with the linking constraints are obtained by solving the master problem. These prices are used by the subsystems to compute an updated solution that improves the overall objective function.

The block-angular constraint matrix structure appears for dy-

*Corresponding author
Email addresses: leso@dtu.dk (L.E. Sokoler), laus@dtu.dk (L. Standardi), kried@dongenergy.dk (K. Edlund), nkpo@dtu.dk (N.K. Poulsen), hmad@dtu.dk (H. Madsen), jbjo@dtu.dk (J.B. Jørgensen)

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 independent energy producers and consumers are controlled to balance power represent yet another instance of dynamically decoupled systems with linking constraints [18, 25]. The temperature regulation of multi-room buildings can also be formulated a dynamically decoupled system with linking constraints [26]. As is evident by this list of examples, dynamically decoupled subsystems with linking constraints are common in process systems.

To test a MATLAB implementation of the Dantzig-Wolfe decomposition algorithm, denoted `DWempc`, a simple energy systems management case study is presented. We show that as more units are added to a network of controllable generators, `DWempc` becomes increasingly favorable over state-of-the-art sparse linear programming solvers provided by `Gurobi`, `CPLEX`, and `MOSEK`. It is further demonstrated that a nearly optimal solution can be acquired, even if `DWempc` is terminated early. This is an attractive property in real-time applications such as EMPC, since only a limited amount of time is available for solving the OCP.

In addition to the general purpose solvers, `DWempc` is compared to an implementation of the alternating direction method of multipliers (ADMM) [27–30], denoted `ADMMempc`, with similar parallelization capabilities to `DWempc`. Simulations illustrate that unless a highly suboptimal control performance is tolerated, `DWempc` outperforms `ADMMempc` with a significant margin. Results also show that for both algorithms, a simple warm-start strategy yields a substantial improvement over cold start, and that the performance of this strategy increases with the weights on the ℓ_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.

1.1. Paper Organization

We have organized the paper as follows. In Section 2, the OCP solved in this paper is introduced. We decompose the problem using Dantzig-Wolfe decomposition in Section 3, and a column generation procedure for solving the decomposed problem is presented. Section 4 describes a distributed implementation of ADMM for solving the OCP. Section 5 reports 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

$$x_{j,k+1} = A_j x_{j,k} + B_j u_{j,k}, \qquad j \in \mathcal{M}, \tag{1a}$$
$$y_{j,k} = C_j x_{j,k}, \qquad j \in \mathcal{M}, \tag{1b}$$

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

$$y_{T,k} = \sum_{j\in\mathcal{M}} \Upsilon_j y_{j,k} = \sum_{j\in\mathcal{M}} \Upsilon_j C_j x_{j,k}, \tag{2}$$

in which $\Upsilon_j \in \mathbb{R}^{n_{y_T} \times n_y(j)}$ are subsystem multipliers.

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

$$\min_{u,x,y,y_T,\rho,\gamma} \quad \psi = \psi_{\text{eco}} + \psi_{\text{reg}}, \tag{3a}$$

with

$$\psi_{\text{eco}} = \sum_{k\in\mathcal{N}_0} \left(q_{k+1}^T \rho_{k+1} + \sum_{j\in\mathcal{M}} p_{j,k}^T u_{j,k} + r_{j,k+1}^T \gamma_{j,k+1} \right), \tag{3b}$$
$$\psi_{\text{reg}} = \sum_{k\in\mathcal{N}_0} \sum_{j\in\mathcal{M}} w_{j,k} \|\Delta u_{j,k}\|_1, \tag{3c}$$

and subject to the constraints

$$x_{j,k+1} = A_j x_{j,k} + B_j u_{j,k}, \qquad k \in \mathcal{N}_0, \quad j \in \mathcal{M}, \tag{3d}$$
$$y_{j,k} = C_j x_{j,k}, \qquad k \in \mathcal{N}_1, \quad j \in \mathcal{M}, \tag{3e}$$
$$y_{T,k} = \sum_{j\in\mathcal{M}} \Upsilon_j C_j x_{j,k}, \qquad k \in \mathcal{N}_1, \tag{3f}$$
$$\underline{u}_{j,k} \le u_{j,k} \le \overline{u}_{j,k}, \qquad k \in \mathcal{N}_0, \quad j \in \mathcal{M}, \tag{3g}$$
$$\Delta\underline{u}_{j,k} \le \Delta u_{j,k} \le \Delta\overline{u}_{j,k}, \qquad k \in \mathcal{N}_0, \quad j \in \mathcal{M}, \tag{3h}$$
$$\underline{y}_{j,k} - \gamma_{j,k} \le y_{j,k} \le \overline{y}_{j,k} + \gamma_{j,k}, \qquad k \in \mathcal{N}_1, \quad j \in \mathcal{M}, \tag{3i}$$
$$0 \le \gamma_{j,k} \le \overline{\gamma}_{j,k}, \qquad k \in \mathcal{N}_1, \quad j \in \mathcal{M}, \tag{3j}$$
$$\underline{y}_{T,k} - \rho_k \le y_{T,k} \le \overline{y}_{T,k} + \rho_k, \qquad k \in \mathcal{N}_1, \tag{3k}$$
$$0 \le \rho_k \le \overline{\rho}_k, \qquad k \in \mathcal{N}_1. \tag{3l}$$

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

The input data to (3) are the input limits, $(\underline{u}_{j,k}, \overline{u}_{j,k})$, the input rate limits, $(\Delta\underline{u}_{j,k}, \Delta\overline{u}_{j,k})$, the subsystem output limits, $(\underline{y}_{j,k}, \overline{y}_{j,k})$, the aggregated variable limits, $(\underline{y}_{T,k}, \overline{y}_{T,k})$, the input prices, $p_{j,k}$, the price for violating the subsystem output limits, $r_{j,k}$, and the

price for violating the aggregated variable limits q_k. The slack variables $\gamma_{j,k}$ and ρ_k account for the violation of the soft output constraints. We impose upper limits, $(\overline{\gamma}_{j,k}, \overline{\rho}_k)$, 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 ℓ_1-penalty on the input rate. Using an ℓ_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

$$\psi = \alpha\psi_{\text{eco}} + (1-\alpha)\psi_{\text{reg}}, \quad \alpha \in [0,\, 1], \tag{4}$$

where α is a user-defined parameter. [12] discusses the trade-off between the economic term and an ℓ_2-regularization term.

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

$$\begin{aligned}\psi_{\text{reg}} = \sum_{k\in\mathcal{N}_0}\Bigg(&\sum_{j\in\mathcal{M}} w^x_{j,k+1}\|x_{j,k+1} - \bar{x}_{j,k+1}\|_1 \\ &+ \sum_{j\in\mathcal{M}} w^u_{j,k}\|u_{j,k} - \bar{u}_{j,k}\|_1 + w^{\Delta u}_{j,k}\|\Delta u_{j,k}\|_1\Bigg),\end{aligned} \tag{5}$$

in which $\{\bar{x}_{j,k+1}, \bar{u}_{j,k}\}_{k\in\mathcal{N}_0, j\in\mathcal{M}}$ 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 ℓ_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

$$y_{j,k} = C_j A_j^k x_{j,0} + \sum_{i\in\mathcal{N}_0} H_{j,k-i}u_{j,i}, \qquad j\in\mathcal{M},\, k\in\mathcal{N}_1,$$

where the impulse response coefficients are given by

$$H_{j,k} = C_j A_j^{k-1} B_j, \qquad j\in\mathcal{M},\, k\in\mathcal{N}_1.$$

Consequently

$$y_{T,k} = \sum_{j\in\mathcal{M}}\left(\Upsilon_j C_j A_j^k x_{j,0} + \sum_{i\in\mathcal{N}_0}\Upsilon_j H_{j,k-i}u_{j,i}\right),\ k\in\mathcal{N}_1.$$

Define the vectors

$$y_j = \begin{bmatrix} y_{j,1}^T & y_{j,2}^T & \cdots & y_{j,N}^T\end{bmatrix}^T, \qquad j\in\mathcal{M}, \tag{6a}$$

$$u_j = \begin{bmatrix} u_{j,0}^T & u_{j,1}^T & \cdots & u_{j,N-1}^T\end{bmatrix}^T, \qquad j\in\mathcal{M}, \tag{6b}$$

and the matrices

$$\Gamma_j = \begin{bmatrix} H_{j,1} & 0 & \cdots & 0 \\ H_{j,2} & H_{j,1} & & \\ \vdots & \vdots & \ddots & \\ H_{j,N} & H_{j,N-1} & \cdots & H_{j,1}\end{bmatrix}, \quad \Phi_j = \begin{bmatrix} C_jA_j \\ C_jA_j^2 \\ \vdots \\ C_jA_j^{N-1}\end{bmatrix},$$

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

$$y_j = \Gamma_j u_j + \Phi_j x_{j,0}, \qquad j\in\mathcal{M}. \tag{7}$$

By introducing $\tilde{\Gamma}_j$ and $\tilde{\Phi}_j$ accordingly, it follows that $y_T = \sum_{j\in\mathcal{M}}\tilde{\Gamma}_j u_j + \tilde{\Phi}_j x_{j,0}$.

The notation is simplified further with

$$\underline{u}_j = \begin{bmatrix}\underline{u}_{j,0}^T & \underline{u}_{j,1}^T & \cdots & \underline{u}_{j,N-1}^T\end{bmatrix}^T, \qquad j\in\mathcal{M},$$

$$\overline{u}_j = \begin{bmatrix}\overline{u}_{j,0}^T & \overline{u}_{j,1}^T & \cdots & \overline{u}_{j,N-1}^T\end{bmatrix}^T, \qquad j\in\mathcal{M},$$

and similarly we define $\Delta\underline{u}_j$, $\Delta\overline{u}_j$, $\underline{y}_j$, $\overline{y}_j$, $\underline{y}_T$, $\overline{y}_T$, $\bar{\gamma}_j$, $\bar{\rho}$, ρ, $\bar{\eta}_j$, η_j, q, p_j, r_j, w_j and γ_j. Using these definitions, the OCP, (3), may be written in the form

$$\min_{u,\rho,\gamma,\eta}\ \psi = q^T\rho + \sum_{j\in\mathcal{M}} p_j^T u_j + r_j^T\gamma_j + w_j^T\eta_j \tag{8a}$$

subject to a set of decoupled constraints

$$\underline{u}_j \le u_j \le \overline{u}_j, \qquad j\in\mathcal{M}, \tag{8b}$$

$$\Delta\underline{u}_j - I_0 u_{j,-1} \le \Lambda u_j \le \Delta\overline{u}_j - I_0 u_{j,-1}, \qquad j\in\mathcal{M}, \tag{8c}$$

$$\underline{y}_j - \gamma_j \le \Gamma_j u_j + \Phi_j x_{j,0} \le \overline{y}_j + \gamma_j, \qquad j\in\mathcal{M}, \tag{8d}$$

$$I_0 u_{j,-1} - \eta_j \le \Lambda u_j \le I_0 u_{j,-1} + \eta_j, \qquad j\in\mathcal{M}, \tag{8e}$$

$$0 \le \eta_j \le \overline{\eta}, \qquad j\in\mathcal{M}, \tag{8f}$$

$$0 \le \gamma_j \le \overline{\gamma}, \qquad j\in\mathcal{M}, \tag{8g}$$

and a set of linking constraints

$$\underline{y}_T - \rho \le \sum_{j\in\mathcal{M}}\tilde{\Gamma}_j u_j + \tilde{\Phi}_j x_{j,0} \le \overline{y}_T + \rho. \tag{8h}$$

$$0 \le \rho \le \bar{\rho}, \tag{8i}$$

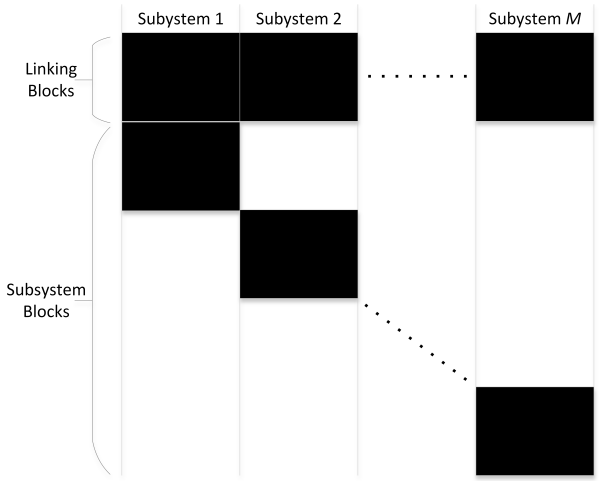


Figure 2: The block-angular structure of the constraint matrix in (8). The efficiency of the Dantzig-Wolfe decomposition method depends on this structure.

where Λ and I_0 are defined as

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

We remark that (8e)-(8f) imply that $\eta_{j,k} \geq |\Delta u_{j,k}|$. 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

$$\min_{z} \ \psi = \sum_{j\in\bar{\mathcal{M}}} c_j^T z_j, \tag{9a}$$

$$\text{s.t. } G_j z_j \geq g_j, \qquad j \in \bar{\mathcal{M}}, \tag{9b}$$

$$\sum_{j\in\bar{\mathcal{M}}} H_j z_j \geq h, \tag{9c}$$

with $\bar{\mathcal{M}} = 1, 2, \ldots, M+1$, and

$$z_j = \begin{bmatrix} u_j^T & \gamma_j^T & \eta_j^T \end{bmatrix}^T, \qquad c_j = \begin{bmatrix} p_j^T & r_j^T & w_j^T \end{bmatrix}^T, \quad j \in \mathcal{M},$$
$$z_{M+1} = \rho^T, \qquad c_{M+1} = q^T.$$

(9b) represents the decoupled constraints (8b)-(8g), and (9c) represents the linking constraints (8h)-(8i).

The data structures in (9) are defined as

$$G_j = \begin{bmatrix} \bar{G}_j \\ -\underline{\bar{G}}_j \end{bmatrix}, \ g_j = \begin{bmatrix} \underline{g}_j \\ -\overline{g}_j \end{bmatrix}, \ H_j = \begin{bmatrix} \bar{H}_j \\ -\underline{\bar{H}}_j \end{bmatrix}, \ h = \begin{bmatrix} \underline{h} \\ -\overline{h} \end{bmatrix},$$

in which

$$\left[\begin{array}{c|c|c} \bar{G}_j & \underline{g}_j & \overline{g}_j \end{array}\right] = \left[\begin{array}{ccc|c|c} I & 0 & 0 & \underline{u}_j & \overline{u}_j \\ \Lambda & 0 & 0 & \Delta\underline{u}_j & \Delta\tilde{u}_j \\ \Gamma_j & I & 0 & \underset{\sim}{y}_j & \infty \\ \Gamma_j & -I & 0 & -\infty & \tilde{y}_j \\ 0 & I & 0 & 0 & \overline{\gamma}_j \\ -\Lambda & 0 & I & -I_0 u_{j,-1} & \infty \\ \Lambda & 0 & I & I_0 u_{j,-1} & \infty \\ 0 & 0 & I & 0 & \overline{\eta}_j \end{array}\right],$$

$$\left[\begin{array}{c|c|c} \bar{H}_j & \underline{h} & \overline{h} \end{array}\right] = \left[\begin{array}{cc|c|c} \tilde{\Gamma}_j & 0 & \underset{\sim}{y}_T & \infty \\ \tilde{\Gamma}_j & 0 & -\infty & \tilde{y}_T \end{array}\right],$$

for $j \in \mathcal{M}$, with

$$\underset{\sim}{y}_T = \underline{y}_T - \sum_{j\in\mathcal{M}} \tilde{\Phi}_j x_{j,0}, \qquad \tilde{y}_T = \overline{y}_T - \sum_{j\in\mathcal{M}} \tilde{\Phi}_j x_{j,0},$$
$$\underset{\sim}{y}_j = \underline{y}_j - \Phi_j x_{j,0}, \qquad \tilde{y}_j = \overline{y}_j - \Phi_j x_{j,0}, \qquad j \in \mathcal{M},$$
$$\Delta\underline{u}_j = \Delta\underline{u}_j + I_0 u_{j,-1}, \qquad \Delta\tilde{u}_j = \Delta\overline{u}_j + I_0 u_{j,-1}, \qquad j \in \mathcal{M},$$

In the special case $j = M+1$

$$\left[\begin{array}{c|c|c} \bar{G}_{M+1} & \underline{g}_{M+1} & \overline{g}_{M+1} \end{array}\right] = \left[\begin{array}{c|c|c} I & 0 & \overline{\rho} \end{array}\right],$$
$$\bar{H}_{M+1} = \begin{bmatrix} I & -I \end{bmatrix}^T.$$

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 $j \in \bar{\mathcal{M}}$, the set of points satisfying the decoupled constraints (9b) may be written as

$$\mathcal{G}_j = \{z_j | G_j z_j \geq g_j\},$$
$$= \left\{ z_j | z_j = \sum_{i\in\mathcal{P}} \lambda_j^i z_j^i, \sum_{i\in\mathcal{P}} \lambda_j^i = 1, \lambda_j^i \geq 0 \ \ \forall i \in \mathcal{P} \right\},$$

where z_j^i are the extreme points of $\mathcal{G}_j$, and λ_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

$$\min_{\lambda} \ \psi = \sum_{j\in\bar{\mathcal{M}}} \sum_{i\in\mathcal{P}} c_j^i \lambda_j^i, \tag{10a}$$

$$\text{s.t. } \sum_{j\in\bar{\mathcal{M}}} \sum_{i\in\mathcal{P}} H_j^i \lambda_j^i \geq h, \tag{10b}$$

$$\sum_{i\in\mathcal{P}} \lambda_j^i = 1, \qquad j \in \bar{\mathcal{M}}, \tag{10c}$$

$$\lambda_j^i \geq 0, \qquad j \in \bar{\mathcal{M}}, \ \ i \in \mathcal{P}, \tag{10d}$$

where we have defined

$$H_j^i = H_j z_j^i, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}, \tag{11a}$$
$$c_j^i = c_j^T z_j^i, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}. \tag{11b}$$

Given a solution, λ^*, to the master problem (10), a solution to the original problem, (9) can be obtained as

$$z_j^* = \sum_{i \in \mathcal{P}} (\lambda^*)_j^i z_j^i, \qquad j \in \bar{\mathcal{M}}.$$

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

$$\max_{\alpha,\beta}\ \phi = \alpha^T h + \sum_{j \in \bar{\mathcal{M}}} \beta_j, \tag{12a}$$
$$\text{s.t. } (H_j^i)^T \alpha + \beta_j \le c_j^i, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}, \tag{12b}$$
$$\alpha \ge 0, \tag{12c}$$

in which $\alpha \in \mathbb{R}^{4N}$ and $\beta \in \mathbb{R}^{M+1}$ 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

$$\sum_{j \in \bar{\mathcal{M}}} \sum_{i \in \mathcal{P}} H_j^i \lambda_j^i \ge h, \tag{13a}$$
$$\sum_{i \in \mathcal{P}} \lambda_j^i = 1, \qquad j \in \bar{\mathcal{M}}, \tag{13b}$$
$$\lambda_j^i \ge 0, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}, \tag{13c}$$
$$c_j^i - (H_j^i)^T \alpha - \beta_j \ge 0, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}, \tag{13d}$$
$$\alpha \ge 0, \tag{13e}$$
$$\lambda_j^i (c_j^i - (H_j^i)^T \alpha - \beta_j) = 0, \qquad j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P}, \tag{13f}$$

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 $j \in \bar{\mathcal{M}}$, and define $(\tilde{\lambda}, \tilde{\alpha}, \tilde{\beta})$ as a primal-dual solution to (10) and (12) restricted to the subset $\tilde{\mathcal{P}}$. Then the solution

$$\begin{aligned}
\alpha^* &= \alpha, \\
\beta_j^* &= \beta_j, && j \in \bar{\mathcal{M}}, \\
(\lambda^*)_j^i &= \begin{cases} \tilde{\lambda}_j^i & \text{if } i \in \tilde{\mathcal{P}} \\ 0 & \text{if } i \in \mathcal{P} \setminus \tilde{\mathcal{P}} \end{cases}, && j \in \bar{\mathcal{M}},\ \ i \in \mathcal{P},
\end{aligned}$$

satisfies the conditions, (13), if the optimal objective value of the subproblem

$$\min_{\tilde{z}_j}\ \varphi_j = (c_j - H_j^T \alpha^*)^T \tilde{z}_j - \beta_j^* \tag{14a}$$
$$\text{s.t. } G_j \tilde{z}_j \ge g_j, \tag{14b}$$

is non-negative for each $j \in \bar{\mathcal{M}}$.

Proof. The solution $(\lambda^*, \alpha^*, \beta^*)$ satisfies (13a) since

$$\sum_{j \in \bar{\mathcal{M}}} \sum_{i \in \mathcal{P}} H_j^i (\lambda^*)_j^i = \sum_{j \in \bar{\mathcal{M}}} \sum_{i \in \tilde{\mathcal{P}}} H_j^i \tilde{\lambda}_j^i \ge h,$$

which follows from the definition of $(\tilde{\lambda}, \tilde{\alpha}, \tilde{\beta})$. Similarly, it can be verified that the conditions (13b), (13c), (13e) and (13f) are fulfilled.

Provided that $(\lambda^*, \alpha^*, \beta^*)$ is optimal, (13d) yields

$$c_j^i - (H_j^i)^T \alpha^* - \beta_j^* = (c_j - H_j^T \alpha^*)^T z_j^i - \beta_j^* \ge 0, \tag{15}$$

for all $j \in \bar{\mathcal{M}}$ and $i \in \mathcal{P}$. By construction of the solution, (15) is satisfied for all $i \in \tilde{\mathcal{P}}$. To check that the condition holds for all $i \in \mathcal{P} \setminus \tilde{\mathcal{P}}$, we consider the optimization problem (14). Since this linear program minimizes the left hand side of (15) over all possible extreme points, $\tilde{z}_j$, of $\mathcal{G}_j$, the solution $(\lambda^*, \alpha^*, \beta^*)$ also satisfies the remaining optimality condition (15) if φ_j is non-negative for all $j \in \bar{\mathcal{M}}$. □

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

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].

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, $\{z_j^0\}_{j \in \bar{\mathcal{M}}}$, at the current sampling instant is constructed from the solution at the previous sampling instant.

Given the solution to (14)

$$\begin{aligned}
u_j^* &= \begin{bmatrix} u_{j,0}^{*T} & \cdots & u_{j,N-1}^{*T} \end{bmatrix}^T, && j \in \mathcal{M}, \\
\gamma_j^* &= \begin{bmatrix} \gamma_{j,1}^{*T} & \cdots & \gamma_{j,N}^{*T} \end{bmatrix}^T, && j \in \mathcal{M}, \\
\eta_j^* &= \begin{bmatrix} \eta_{j,0}^{*T} & \cdots & \eta_{j,N-1}^{*T} \end{bmatrix}^T, && j \in \mathcal{M}, \\
\rho^* &= \begin{bmatrix} \rho_1^{*T} & \cdots & \rho_N^{*T} \end{bmatrix}^T,
\end{aligned}$$

we construct an initial point for the following sampling instant as

$$z_j^0 = \begin{bmatrix} (u_j^0)^T & (\gamma_j^0)^T & (\eta_j^0)^T \end{bmatrix}^T, \qquad j \in \mathcal{M},$$

Algorithm 1 Column generation procedure for solution of (10).

Require: $(i_{\max}, \varepsilon), \{z_j^0\}_{j\in\bar{\mathcal{M}}}$
  $i = 0$, `converged = false`
  while not `converged` **and** $i < i_{\max}$ **do**
    $\tilde{\mathcal{P}} = \{0, 1, \ldots, i\}$
    `COMPUTE PROBLEM DATA`
    for $j \in \bar{\mathcal{M}}$ **do**
      $H_j^i = H_j z_j^i$
      $c_j^i = c_j^T z_j^i$
    end for
    `SOLVE RESTRICTED MASTER PROBLEM`
    $(\phi^*, \lambda^*, \alpha^*, \beta^*) \leftarrow$ solve (10) with $\mathcal{P} = \tilde{\mathcal{P}}$
    `SOLVE SUBPROBLEMS`
    for $j \in \bar{\mathcal{M}}$ **do**
      $(\varphi_j^*, \tilde{z}_j^*) \leftarrow$ solve (14)
    end for
    `CHECK IF CONVERGED`
    if $|\varphi_j| \geq \varepsilon \ \ \forall j \in \bar{\mathcal{M}}$ **then**
      `converged = true`
    else
      `UPDATE EXTREME POINTS`
      for $j \in \bar{\mathcal{M}}$ **do**
        $z_j^{i+1} = \tilde{z}_j^*$
        $i = i + 1$
      end for
    end if
  end while

where

$$u_j^0 = \begin{bmatrix} u_{j,1}^{*T} & \cdots & u_{j,N-1}^{*T} & (u_{j,N}^0)^T \end{bmatrix}^T, \qquad j \in \mathcal{M},$$
$$\gamma_j^0 = \begin{bmatrix} \gamma_{j,2}^{*T} & \cdots & \gamma_{j,N}^{*T} & (\gamma^0)_{j,N+1}^T \end{bmatrix}^T, \qquad j \in \mathcal{M},$$
$$\eta_j^0 = \begin{bmatrix} \eta_{j,1}^{*T} & \cdots & \eta_{j,N-1}^{*T} & (\eta^0)_{j,N}^T \end{bmatrix}^T, \qquad j \in \mathcal{M}.$$

Finally

$$z_{M+1}^0 = \rho^0 = \begin{bmatrix} \rho_2^{*T} & \cdots & \rho_N^{*T} & (\rho^0)_{N+1}^T \end{bmatrix}^T.$$

The original solution values are thus shifted forward in time, and $u_{j,N}^0$, $\gamma_{j,N+1}^0$, $\eta_{j,N}^0$ and ρ_{N+1}^0 are appended to the resulting initial point.

In our implementation

$$u_{j,N}^0 = u_{j,N-1}^*, \qquad j \in \mathcal{M},$$
$$\eta_{j,N}^0 = 0, \qquad j \in \mathcal{M}.$$

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

$$\gamma_{j,N+1}^0 = \max(\underline{y}_{j,N+1} - y_{j,N+1}^0, 0) + \max(y_{j,N+1}^0 - \overline{y}_{j,N+1}, 0),$$

for each $j \in \mathcal{M}$, and

$$\rho_{N+1}^0 = \max(\underline{y}_{T,N+1} - y_{T,N+1}^0, 0) + \max(y_{T,N+1}^0 - \overline{y}_{T,N+1}, 0).$$

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, $\{z_j^0\}_{j\in\bar{\mathcal{M}}}$, in Algorithm 1 can be constructed by adjusting the slack variables, γ_j^0 and ρ^0. Let $\{u_j^0\}_{j\in\mathcal{M}}$ 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 $u_j^0 = \underline{u}_j$ for each $j \in \mathcal{M}$. Then, in a similar way as for the warm-start strategy, we compute

$$\gamma_{j,k}^0 = \max(\underline{y}_{j,k} - y_{j,k}^0, 0) + \max(y_{j,k}^0 - \overline{y}_{j,k}, 0),$$
$$\rho_k^0 = \max(\underline{y}_{T,k} - y_{T,k}^0, 0) + \max(y_{T,k}^0 - \overline{y}_{T,k}, 0),$$

where $k \in \mathcal{N}_1, \quad j \in \mathcal{M}$. The values, $y_{j,k}^0$ and $y_{T,k}^0$ are the subsystem outputs and the aggregated variables associated with the inputs, $\{u_j^0\}_{j\in\mathcal{M}}$, 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

$$\begin{aligned} \min_{z,v} \ & \psi = \sum_{j\in\bar{\mathcal{M}}} c_j^T z_j, \\ \text{s.t.} \ & G_j z_j \geq g_j, && j \in \bar{\mathcal{M}}, \\ & H_j z_j = v_j, && j \in \bar{\mathcal{M}}, \\ & \sum_{j\in\bar{\mathcal{M}}} v_j \geq h, \end{aligned}$$

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

$$\min_{z,v} \ \psi = \sum_{j\in\bar{\mathcal{M}}} \left(c_j^T z_j + I_{\mathcal{Z}_j}(z_j) \right) + I_{\mathcal{V}}(v), \tag{16a}$$
$$\text{s.t.} \ H_j z_j = v_j, \qquad j \in \bar{\mathcal{M}} \tag{16b}$$

where $\mathcal{Z}_j = \left\{ z_j | G_j z_j \geq g_j \right\}$, $\mathcal{V} = \left\{ v | \sum_{j\in\bar{\mathcal{M}}} v_j \geq h \right\}$, and $I_{\mathbb{A}}$ is the indicator function of a set $\mathbb{A}$ defined as

$$I_{\mathbb{A}}(x) = \begin{cases} 0 & \text{if } x \in \mathbb{A}, \\ \infty & \text{otherwise.} \end{cases}$$

For the problem (16), the ADMM recursions described in [27] can be written as

$$z_j^{i+1} = \underset{z_j \in \mathcal{Z}_j}{\operatorname{argmin}}\, c_j^T z_j + \tfrac{\rho}{2}\|H_j z_j - v_j^i + u_j^i\|_2^2, \quad j \in \bar{\mathcal{M}}, \tag{17a}$$

$$v^{i+1} = \underset{v \in \mathcal{V}}{\operatorname{argmin}}\, \tfrac{\rho}{2} \sum_{j \in \bar{\mathcal{M}}} \|H_j z_j^{i+1} - v_j + u_j^i\|_2^2, \tag{17b}$$

$$u_j^{i+1} = u_j^i + H_j z_j^{i+1} - v_j^{i+1}, \quad j \in \bar{\mathcal{M}}, \tag{17c}$$

where u^i is a scaled dual variable.

The update of z_j, (17a), thus consists of solving the constrained quadratic program

$$\min_{z_j} \ \tfrac{\rho}{2} z_j^T H_j^T H_j z_j + (c_j + \rho(-v_j^i + u_j^i)^T H_j)^T z_j, \tag{18a}$$

$$\text{s.t. } G_j z_j \geq g_j \tag{18b}$$

for each $j \in \bar{\mathcal{M}}$.

The update for v, (17b), yields the explicit solution

$$v_j^{i+1} = H_j z_j^{i+1} + u_j^i + \max(l/(M+1), 0), \qquad j \in \bar{\mathcal{M}},$$

where $l = h - \sum_{j \in \bar{\mathcal{M}}} H_j z_j^{i+1} + u_j^i$. 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 $\alpha H_j z^{i+1} - (1-\alpha)(-v_j^i)$ 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

$$\|r^i\|_2 \leq \epsilon_p, \qquad \|s^i\|_2 \leq \epsilon_d,$$

in which

$$s_j^{i+1} = -\rho H_j^T (v^{i+1} - v^i), \qquad r_j^{i+1} = \rho H_j z_j^{i+1} - v_j^{i+1},$$

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 $(v^0, u^0) = (\mathbf{0}, \mathbf{0})$ 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. ℓ_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: $(\rho, \alpha, i_{\max}, \epsilon_P, \epsilon_D)$, (v^0, u^0), (ϵ_P, ϵ_D)
  $i = 0$, `converged = false`
  while `converged = false` **and** $i < i_{\max}$ **do**
    `UPDATE VARIABLES`
    for $j \in \bar{\mathcal{M}}$ **do**
      $z_j^{i+1} = \underset{z_j \in \mathcal{Z}_j}{\operatorname{argmin}}\, c_j^T z_j + \frac{\rho}{2}\|H_j z_j - v_j^i + u_j^i\|_2^2$
    end for
    $l = h - \sum_{j \in \bar{\mathcal{M}}} \alpha H_j z^{i+1} - (1-\alpha)(-v_j^i) + u_j^i$
    for $j \in \bar{\mathcal{M}}$ **do**
      $v_j^{i+1} = \alpha H_j z^{i+1} - (1-\alpha)(-v_j^i) + u_j^i + \max(l/(M+1), 0)$
      $u_j^{i+1} = u_j^i + \alpha H_j z^{i+1} - (1-\alpha)(-v_j^i) - v_j^{i+1}$
    end for
    `COMPUTE RESIDUALS`
    for $j \in \bar{\mathcal{M}}$ **do**
      $r_j^{i+1} = \rho H_j z_j^{i+1} - v_j^{i+1}$
      $s_j^{i+1} = -\rho H_j^T (v^{i+1} - v^i)$
    end for
    `CHECK IF CONVERGED`
    if $\|r^{i+1}\|_2 \leq \epsilon_P$ **and** $\|s^{i+1}\|_2 \leq \epsilon_D$ **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

$$Y_j(s) = \frac{1}{(\tau_j s + 1)^3}\Big(U_j(s) + D_j(s)\Big) + E_j(s), \quad j \in \mathcal{M}, \tag{19}$$

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, τ_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.

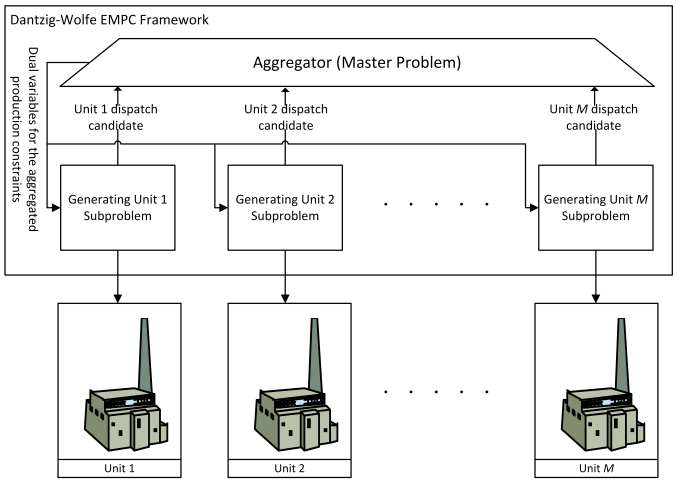


Figure 3: EMPC diagram for the Dantzig-Wolfe decomposition algorithm for a dynamic multi-plant system with linking constraints.

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

The total power produced by the M generating units is

$$Y_T(s) = \sum_{j=1}^{M} \frac{1}{(\tau_j s + 1)^3} \Big(U_j(s) + D_j(s) \Big). \tag{20}$$

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

$$x_{j,k+1} = A_j x_{j,k} + B_j u_{j,k} + E_j \boldsymbol{d}_{j,k}, \qquad j \in \mathcal{M}, \tag{21a}$$
$$y_{j,k} = C_j x_{j,k} + \boldsymbol{e}_{j,k}, \qquad j \in \mathcal{M}, \tag{21b}$$
$$y_{T,k} = \sum_{j \in \mathcal{M}} C_j x_{j,k}, \tag{21c}$$

In the resulting model structure, $u_{j,k} \in \mathbb{R}$ is the unit input (power setpoint), $y_{j,k} \in \mathbb{R}$ is the unit power production, and $y_{T,k} \in \mathbb{R}$ is the total power production. We assume that $\boldsymbol{x}_{j,0} \sim N(\hat{x}_{j,0}, P_{j,0})$, $\boldsymbol{d}_{j,k} \sim N(0, R_{j,d})$, and that $\boldsymbol{e}_{j,k} \sim N(0, R_{j,e})$. 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 $j \in \mathcal{M}$, 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, $\{\hat{u}_j\}_{j=1}^{M}$, 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

$$\hat{\psi} = q^T \hat{\rho} + \sum_{j \in \mathcal{M}} p_j^T \hat{u}_j + r_j^T \hat{\gamma}_j + w_j^T \hat{\eta},$$

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

$$\omega = 100 \frac{\hat{\psi} - \psi^*}{\max(|\psi^*|, 1)}. \tag{22}$$

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, τ_j, sampled from the uniform distribution over the interval $[20, 80]$. For simplicity, it is assumed that $\boldsymbol{d}_{j,k} \sim N(0, (10\sigma)^2 I)$, $\boldsymbol{e}_{j,k} \sim N(0, \sigma^2 I)$, and that full initial state information is given such that $x_{j,0} \sim (0, 0)$.

The power generating unit input price is $p_{j,k} = 1/\tau_j$. 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

$$(\underline{u}_{j,k}, \overline{u}_{j,k}, \Delta\overline{u}_{j,k}, \Delta\underline{u}_{j,k}) = (0, 8/M, -M/4, M/4).$$

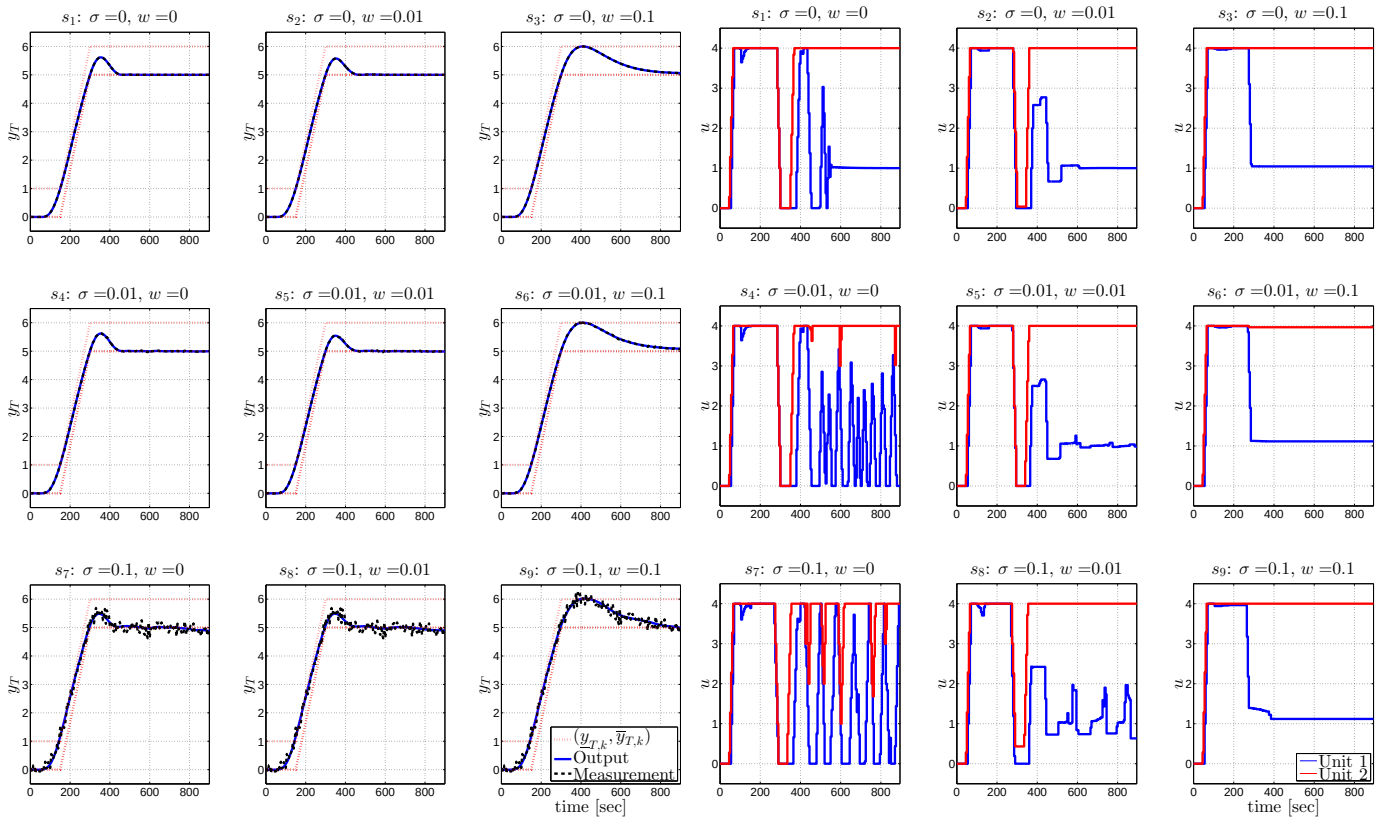
In this way the possible contribution from each unit to the overall power production diminishes as the number of units is increased. Local output constraints in the form (3i)-(3j) are not present. The local output variables, $y_{j,k}$, and the local slack variables, $\gamma_{j,k}$, are thus excluded 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 `DWempc` is set to $\varepsilon =$ `1e-4`. `ADMMmpc` uses the following primal and dual tolerance specification: $\epsilon_P = \epsilon_D =$ `1e-2`. Both `DWempc` and `ADMMempc` 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, σ, and the regularization weights,



(a) Total power output, y_T.

(b) Inputs (power setpoints), u_1 and u_2, for each power plant.

Figure 4: Closed-loop simulations of the system (21) controlled by EMPC. The OCP (3) representing the EMPC is solved to a specified tolerance using CPLEX. The figures illustrate the total output and the inputs for different values of the noise parameter, σ, and the regularization weights, w. The effect of the regularization is most clearly observed in the inputs. At the expense of slightly less tight control on the total power output, the inputs become less volatile when the regularization weight is increased.

Table 1: Case study simulation and controller parameters.

	τ_i	p_k	$\underline{u}_k$	$\overline{u}_k$	$\Delta\underline{u}_k$	$\Delta\overline{u}_k$
Generating Unit 1	65	1/65	0	4	-1	1
Generating Unit 2	75	1/75	0	4	-1	1

$w = w_{j,k}$. 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 s_6, corresponding to $\sigma = 0.01$ 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`.

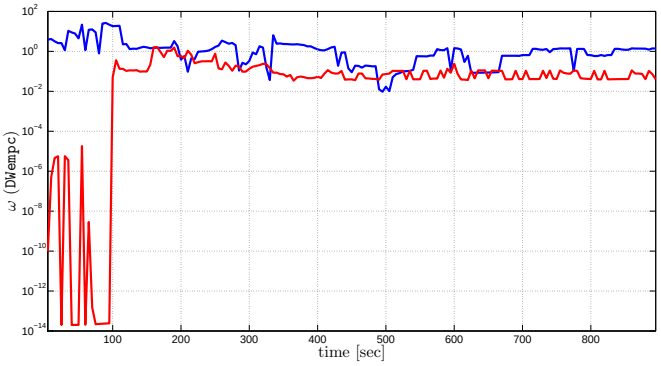
Fig. 5 shows the level of suboptimality, ω, computed via (22), for scenario s_5 when the run time of `DWempc` and `ADMMempc` is limited to 0.01 seconds. We observe that `DWempc` is up to approximately 30% suboptimal when cold-started, and not more than than 5% suboptimal when warm-started. Hence, although 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.

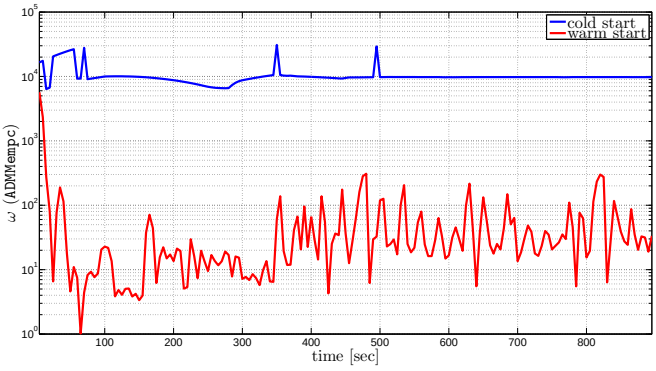
Fig. 6 depicts the level of suboptimality as a function of the CPU time including both `DWempc` and `ADMMempc`, for a single instance of the OCP with 128 generating units. Initially, `ADMMempc` finds the best solution. The quality of this solution is however far from optimal, making it economically very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

Table 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).

	σ	w	DWempc	ADMMempc
s_1	0	0	[6(2), 16(17), 12(11)]	[47(2), 485(410), 097(66)]
s_2	0	0.01	[6(2), 15(18), 10(09)]	[35(3), 469(410), 088(56)]
s_3	0	0.1	[5(2), 15(17), 07(07)]	[33(6), 359(280), 149(48)]
s_4	0.01	0	[7(2), 18(19), 13(11)]	[47(2), 485(410), 094(65)]
s_5	0.01	0.01	[6(2), 17(17), 10(09)]	[35(2), 469(410), 088(58)]
s_6	0.01	0.1	[5(2), 13(16), 07(06)]	[32(6), 380(290), 145(50)]
s_7	0.1	0	[7(2), 17(20), 12(11)]	[46(2), 485(410), 091(66)]
s_8	0.1	0.01	[6(2), 17(16), 09(09)]	[35(2), 469(410), 084(60)]
s_9	0.1	0.1	[5(2), 14(14), 07(06)]	[32(6), 359(279), 144(47)]



(a) DWempc



(b) ADMMempc

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

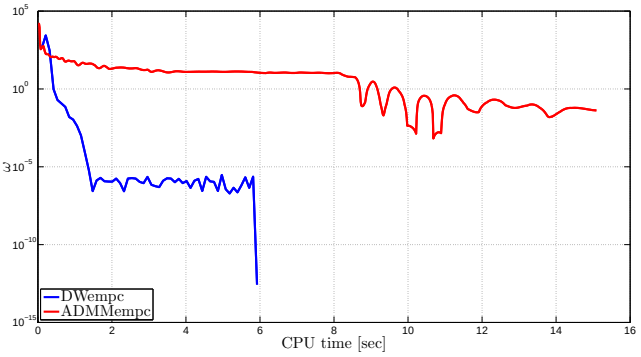


Figure 6: Level of suboptimality as a function of the CPU time, for a single instance of the OCP with 128 generating units.

Table 3: Tolerance specifications for DWempc and ADMMempc.

Accuracy	ε	ϵ_P	ϵ_D
High (h)	1e-6	1e-4	1e-4
Medium (m)	1e-5	1e-3	1e-3
Low (l)	1e-4	1e-2	1e-2

seconds, and at 0.3 seconds a solution which is less than 1% suboptimal is found. Moreover, while DWempc keeps improving until a highly accurate solution is found, ADMMempc suffers from a much slower convergence rate. Only after 10 seconds is a solution with a suboptimality level of 1% found by this algorithm.

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

Table 4: CPU time for solving (3) with an increasing number of generating units, M.

solver/M	16	32	64	128	256	512	1024	2048
Gurobi	1.16e-1	2.93e-1	8.22e-1	1.85	3.94	9.76	2.49e1	5.00e1
CPLEX	1.73e-1	3.49e-1	1.86	9.54e-1	1.83	4.02	8.28	1.83e1
MOSEK	1.60e-1	4.14e-1	8.66e-1	2.23	3.59	7.96	2.19e1	4.54e1
DWempc(l)	4.70e-2	6.25e-2	9.04e-2	1.83e-1	2.94e-1	6.78e-1	1.22	3.54
DWempc(m)	4.95e-2	7.20e-2	1.54e-1	2.61e-1	4.59e-1	1.28	2.78	5.20
DWempc(h)	6.48e-2	9.48e-2	2.03e-1	3.58e-1	6.74e-1	1.80	4.91	9.61
ADMMempc(l)	9.13e-1	1.65	3.38	4.50	6.44	1.16e1	2.32e1	5.70e1
ADMMempc(m)	1.76	3.06	4.28	1.03e1	2.45e1	7.76e1	2.58e2	-
ADMMempc(h)	8.49	7.88e1	5.59e2	-	-	-	-	-

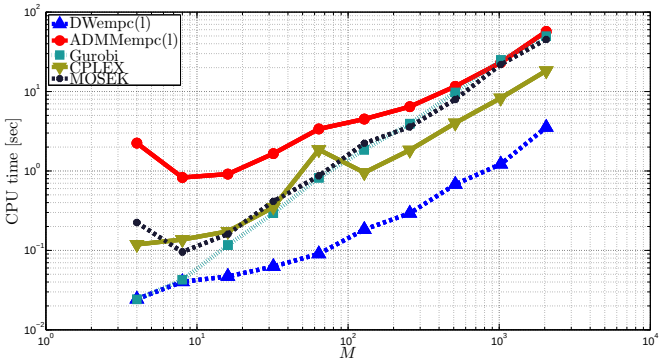


Figure 7: CPU time for the different solvers as a function of the number of power generating units, M.

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, DWempc is an attractive optimization algorithm for large scale dynamically decoupled energy management problems.

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

Table 6 lists the suboptimality level of the solution determined by DWempc and ADMMempc for different values of M. As observed from Table 6, DWempc is not only faster than 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 ℓ_1-regularized linear EMPC of dynamically decoupled subsystems. Simulations show that a MATLAB implementation of the proposed algorithm, denoted DWempc, is faster than CPLEX, Gurobi and MOSEK, as well as an implementation of ADMM denoted ADMMempc. Both DWempc and 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 DWempc yields a highly accurate solution after only a few iterations. In contrast to ADMMempc, the number of iterations required by DWempc to achieve a certain tolerance level does not grow with the problem size.

For cases when the number of DWempc iterations is large, DWempc may be slower than ADMMempc. The reason is that the computing time per iteration of DWempc grows with the iteration number. Conversely, the time spent per iteration by 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, DWempc outperforms ADMMempc; in some cases by several orders of magnitude.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814.

[2] J. B. Rawlings, Tutorial overview of model predictive control, IEEE Control Systems 20 (3) (2000) 38–52.

[3] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (7) (2003) 733–764.

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

[5] M. Diehl, R. Amrit, J. B. Rawlings, A Lyapunov Function for Economic Optimizing Model Predictive Control, IEEE Transactions on Automatic Control 56 (3) (2011) 703–707.

[6] L. Grüne, Economic receding horizon control without terminal constraints, Automatica 49 (3) (2013) 725–734.

[7] J. B. Rawlings, D. Angeli, 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, J. B. Rawlings, On Average Performance and Stability of Economic Model Predictive Control, IEEE Transactions on Automatic Control 57 (7) (2012) 1615–1626.

[9] J. B. Rawlings, D. Bonne, J. B. Jørgensen, A. N. Venkat, S. B. Jørgensen, Unreachable Setpoints in Model Predictive Control, IEEE Transactions on Automatic Control 53 (9) (2008) 2209–2215.

[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 `DWempc` and `ADMMempc` in solving (3) for an increasing number of generating units, M.

`solver`/M	16	32	64	128	256	512	1024	2048
`DWempc(l)`	9	8	7	7	6	6	5	5
`DWempc(m)`	10	9	10	9	8	9	8	6
`DWempc(h)`	12	12	11	12	11	10	11	11
`ADMMempc(l)`	93	157	303	352	395	436	530	615
`ADMMempc(m)`	178	286	370	705	1284	2520	4994	-
`ADMMempc(h)`	865	7468	50000	-	-	-	-	-

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

`solver`/M	16	32	64	128	256	512	1024	2048
`DWempc(l)`	2.08e-2	1.67e-2	1.53e-1	1.86e-1	9.65e-1	9.44e-1	3.18	3.13
`DWempc(m)`	1.66e-5	3.87e-3	7.40e-3	1.94e-2	4.74e-2	5.29e-2	1.63e-1	9.37e-1
`DWempc(h)`	-2.46e-10	-1.13e-9	6.46e-6	1.98e-4	7.55e-4	1.06e-3	2.32e-3	4.82e-2
`ADMMempc(l)`	4.58e-1	6.65e-1	7.92e1	2.09e2	2.84e2	5.50e2	8.13e2	1.33e3
`ADMMempc(m)`	8.16e-2	8.36e-2	9.97	1.38e-1	1.53e1	6.62e-1	6.96	-
`ADMMempc(h)`	1.87e-1	1.86e-2	3.36e-5	-	-	-	-	-

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, 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, L. T. Biegler, Optimizing process economics online using model predictive control, Computers & Chemical Engineering 58 (0) (2013) 334–343.

[13] A. J. Conejo, E. Castillo, R. Minguez, R. Garcia-Bertrand, Decomposition Techniques in Mathematical Programming: Engineering and Science Applications, Springer, 2006.

[14] G. B. Dantzig, P. Wolfe, Decomposition Principle for Linear Programs, Operations Research 8 (1) (1960) 101–111.

[15] R. K. Martin, Large Scale Linear and Integer Optimization: A Unified Approach, Kluwer Academic Publishers, 1999.

[16] R. Cheng, J. F. Forbes, W. S. Yip, Dantzig-Wolfe decomposition and plant-wide MPC coordination, Computers & Chemical Engineering 32 (7) (2008) 1507–1522.

[17] R. Bourdais, J. Buisson, D. Dumur, H. Guéguen, P.-D. Morosan, Distributed MPC Under Coupled Constraints Based on Dantzig-Wolfe Decomposition, in: J. M. Maestre, R. R. Negenborn (Eds.), Distributed Model Predictive Control Made Easy, Intelligent Systems, Control and Automation: Science and Engineering, Springer-Verlag GmbH, 2013, Ch. 6, pp. 101–115.

[18] K. Edlund, J. D. Bendtsen, J. B. Jørgensen, Hierarchical model-based predictive control of a power plant portfolio, Control Engineering Practice 19 (10) (2011) 1126–1136.

[19] L. S. Lasdon, Optimization Theory for Large Systems, Dover Publications Inc., 2002.

[20] H. P. Williams, Model Building in Mathematical Programming, Third Revised Edition, John Wiley & Sons, 1995.

[21] A. Alabi, J. Castro, Dantzig-Wolfe and Block Coordinate-descent Decomposition in Large-scale Integrated Refinery-planning, Computers & Operations Research 36 (8) (2009) 2472–2483.

[22] J. J. Siirola, T. F. Edgar, Process energy systems: Control, economic, and sustainability objectives, Computers & Chemical Engineering 47 (0) (2012) 134–144.

[23] V. Gunnerud, B. Foss, E. Torgnes, Parallel Dantzig-Wolfe decomposition for real-time optimization-Applied to a complex oil field, Journal of Process Control 20 (9) (2010) 1019–1026.

[24] V. Gunnerud, B. Foss, Oil production optimization-A piecewise linear model, solved with two decomposition strategies, Computers & Chemical Engineering 34 (11) (2010) 1803–1812.

[25] M. Soroush, D. J. Chmielewski, Process systems opportunities in power generation, storage and distribution, Computers & Chemical Engineering 51 (0) (2013) 86–95.

[26] P.-D. Morosan, R. D. Bourdais, J. Didier Buisson, Distributed MPC for Multi-Zone Temperature Regulation with Coupled Constraints, in: Proceedings of the 18th IFAC World Congress, 2011, pp. 1552–2557.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, E. Jonathan, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[28] C. Conte, T. Summers, M. N. Zeilinger, M. Morari, C. N. Jones, Computational aspects of distributed optimization in model predictive control, in: 2012 IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp. 6819–6824.

[29] T. H. Summers, J. Lygeros, Distributed model predictive consensus via the Alternating Direction Method of Multipliers, in: 2012 50th Annual Allerton Conference on Communication, Control and Computing (Allerton), 2012, pp. 79–84.

[30] M. Annergren, A. Hansson, B. Wahlberg, An ADMM algorithm for solving ℓ_1 regularized MPC, in: 2012 IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp. 4486–4491.

[31] A. Capolei, E. Suwartadi, B. Foss, J. B. Jørgensen, A Mean-Variance Objective for Robust Production Optimization in Uncertain Geological Scenarios, Journal of Petroleum Science and Engineering (2013) Submitted.

[32] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, Marco Zugno, Integrating Renewables in Electricity Markets, International Series in Operations Research & Management Science, Springer, 2014.

[33] L. E. Sokoler, G. Frison, A. Skajaa, R. Halvgaard, J. B. Jørgensen, A Homogeneous and Self-Dual Interior-Point Linear Programming Algorithm for Economic Model Predictive Control, IEEE Transactions on Automatic Control (2013) Submitted.

[34] G. Frison, J. B. Jørgensen, A Fast Condensing Method for Solving Linear-Quadratic Control Problems, in: 2013 IEEE 52th Annual Conference on Decision and Control (CDC), 2013, pp. 7715–7720.

[35] M. Hong, Z.-Q. Luo, On the Linear Convergence of the Alternating Direction Method of Multipliers, Mathematical Programming (2013) Submitted.

[36] J. Eckstein, M. C. Ferris, Operator-Splitting Methods for Monotone Affine Variational Inequalities, with a Parallel Application to Optimal Control, Informs Journal on Computing 10 (2) (1998) 218–235.

[37] J. B. Jørgensen, J. B. Rawlings, 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.

[38] R. A. Bartlett, L. T. Biegler, QPSchur: A dual, active-set, schur-complement method for large-scale and structured convex quadratic programming, Optimization and Engineering 7 (2006) 5–32.
[39] H. J. Ferreau, H. G. Bock, M. Diehl, An online active set strategy to overcome the limitations of explicit MPC, International Journal of Robust and Nonlinear Control 18 (8) (2008) 816–830.
[40] Y. Wang, S. Boyd, Fast Model Predictive Control Using Online Optimization, IEEE Transactions on Control Systems Technology 18 (2) (2010) 267–278.
[41] C. V. Rao, S. J. Wright, J. B. Rawlings, Application of Interior-Point Methods to Model Predictive Control, Journal of Optimization Theory and Applications 99 (3) (1998) 723–757.
[42] A. Shahzad, G. A. Kerrigan, E. C. Constantinides, A fast well-conditioned interior point method for predictive control, in: 2010 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 508–513.
[43] K. Edlund, L. E. Sokoler, 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.
[44] S. Richter, C. N. Jones, M. Morari, Real-time input-constrained MPC using fast gradient methods, in: 48th IEEE Conference on Decision and Control, held jointly with 28th Chinese Control Conference, 2009, pp. 7387–7393.
[45] J. B. Jørgensen, G. Frison, N. F. Gade-Nielsen, 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.
[46] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, M. Morari, Embedded Online Optimization for Model Predictive Control at Megahertz Rates, IEEE Transactions on Automatic Control (2013) Accepted.
[47] T. G. Hovgaard, L. F. S. Larsen, 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.
[48] J. Ma, S. J. Qin, B. Li, T. Salsbury, Economic model predictive control for building energy systems, in: 2011 IEEE PES Innovative Smart Grid Technologies (ISGT), 2011, pp. 1–6.
[49] R. Halvgaard, N. K. Poulsen, H. Madsen, 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.
[50] R. Halvgaard, N. K. Poulsen, H. Madsen, J. B. Jørgensen, F. Marra, 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.
[51] L. Standardi, L. E. Sokoler, N. K. Poulsen, J. B. Jørgensen, Computational Efficiency of Economic MPC for Power Systems Operation, in: 2013 IEEE PES Innovative Smart Grid Technologies (ISGT), 2013, p. Published.
[52] L. E. Sokoler, L. Standardi, K. Edlund, J. B. Jørgensen, A Dantzig-Wolfe Decomposition Algorithm for Optimal Operation of Distributed Energy Systems, in: 2013 IEEE PES Innovative Smart Grid Technologies (ISGT), Vol. 2013 IEEE, 2013, p. Published.
[53] K. Edlund, T. Mølbak, 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.
[54] G. Desaulniers, J. Desrosiers, M. M. Solomon, Column Generation, Springer, 2005.