

# Fast Coordinated Model Predictive Control of Large-Scale Distributed Systems with Single Coupling Constraint

Vedrana Spudić and Mato Baotić

**Abstract**—In this paper we describe an efficient implementation of a discrete-time model predictive controller for coordinated control of large-scale distributed systems. The approach is based on parametrization and splitting of the underlying optimization problem into local optimization problems coupled via one coordination problem. The local problems are solved off-line using the multi-parametric optimization approach. The coordination problem is solved on-line at every time step, on a centralized control hardware. Due to the properties of local parametric solutions the coordination problem has a specific structure that can be utilized to solve the on-line problem very efficiently. In case when there is only one linear coupling constraint between subsystems, the on-line computation time can be drastically decreased. The efficiency of proposed controller implementation is depicted on a design of the optimal wind farm controller applicable for implementation on very large wind farms.

## I. INTRODUCTION

In recent years, the Model Predictive Control (MPC) was established as a predominant methodology for dynamic optimal control of discrete-time systems with constraints. The reason for this is twofold: the MPC strategy is practical for design and it has a well established theoretical background. The idea of a discrete-time MPC is very simple: the mathematical model of the system is used to predict system evolution - on a (finite) prediction horizon and with a known (estimated or measured) initial state - as a function of control inputs. At every time step controller computes the values of control inputs that minimise an appropriately chosen cost function. The first control input is then applied to the system and the entire procedure is repeated at the next sampling instant.

Various aspects of the MPC strategy (e.g., optimality, stability, robustness, computational feasibility) are discussed thoroughly in the literature [1], [2]. The strength of the MPC lies in the fact that the process (variables) constraints can be systematically included in the design procedure. However, the application of the MPC to systems with small sampling time or complex systems with large number of control inputs, long prediction horizon and/or a large number of constraints can be a challenging task. In such cases the underlying optimization problem, which needs to be solved at every sampling instant, becomes computationally demanding and often cannot be solved within (one) time sample.

This research has been supported by the European Community's Seventh Framework Programme under grant No. 285939 (ACROSS) and Croatian Science Foundation under grant No. I-4463-2011 (MICROGRID).

V. Spudić and M. Baotić are with Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia  
{vedrana.spudic,mato.baotic}@fer.hr

The approach that has drastically increased the applicability of the MPC to fast systems is the synthesis of the so-called explicit model predictive controller, which is accomplished by an off-line computation of the optimal control law via multi-parametric optimization [3]. By solving the appropriate multi-parametric optimization problem one obtains the optimal control input as a closed form function of the process state. The implementation of the explicit MPC controller thus becomes equivalent to the implementation of a gain-scheduled state feedback controller. The downside of the explicit model predictive controller is that the complexity of control law often increases fast with complexity of the original optimization problem, which practically means that it is not possible to compute an explicit model predictive controller for systems with a large number of states, control signals and/or constraints.

The mathematical models of large-scale distributed systems (i.e. systems that consist of a large number of subsystems equipped with local control hardware) contain a large number of control variables and states. Therefore, using the MPC approach for coordinated optimal control of such systems is typically inapplicable, in both explicit and implicit form. Typical approach found in literature for implementation of such controllers consists in decentralization of computation – the optimal control problem is solved on several control hardwares in parallel, which can, if required, negotiate between themselves during the optimization process. The overview of different approaches for distributed MPC can be found in survey papers, e.g. [4], [5]. In the distributed MPC scheme the convergence towards a globally optimal solution is attempted by employing an iterative communication-optimization procedure.

In this paper we pursue a different, *centralized* approach that combines the on-line and off-line computation when solving an MPC problem for coordinated control of large-scale distributed system. We consider a linear MPC problem with convex quadratic cost function. A certain amount of computational effort is transferred off-line – the contribution of each subsystem to the overall cost function and its optimal control actions are determined off-line as a closed form function in subsystem's parameters. On-line, the globally optimal control action is determined, in a centralized manner, by solving an optimization program based on the off-line solutions. The on-line solved optimization problem has a specific structure that rests on the theoretic properties of multi-parametric solution of quadratic cost based MPC formulation. The problem structure enables design of a very efficient on-line solver that, by utilizing efficient primal

decomposition of the optimization problem, allows implementation on very large distributed systems.

In Section II the control problem is defined. Section III describes the proposed implementation approach. In Section IV the efficiency of the proposed approach is tested on a design of an optimal controller of a wind farm. Section V concludes the paper.

## II. PROBLEM DEFINITION

Consider a large-scale system that comprises  $M$  locally controllable subsystems with linear time-invariant discrete-time dynamics:

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t), \quad (1)$$

where  $t \in \mathbb{Z}$  denotes discrete time,  $i \in \{1, \dots, M\}$  is the subsystem index,  $x_i(t) \in \mathbb{R}^{n_{x,i}}$  and  $u_i(t) \in \mathbb{R}^{n_{u,i}}$  denote state and input of  $i$ -th subsystem at time  $t$ , respectively, while  $A_i \in \mathbb{R}^{n_{x,i} \times n_{x,i}}$  and  $B_i \in \mathbb{R}^{n_{x,i} \times n_{u,i}}$  are constant matrices. For simplicity of exposition, but without loss of generality, we assume that  $n_{x,i} = n_x$ ,  $n_{u,i} = n_u$ ,  $\forall i \in \{1, \dots, M\}$ .

We consider the following Constrained Finite-Time Optimal Control (CFTOC) problem for optimal coordinated control of  $M$  distributed subsystems (1) subject to state and input constraints:

$$\begin{aligned} \min_{U_1, \dots, U_M} \quad & \sum_{i=1}^M U_i' R_i U_i + X_i' Q_i X_i, \\ \text{subject to} \quad & \begin{cases} X_i = A_i x_{0,i} + B_i U_i, & i = 1, \dots, M, \\ X_i \in \mathbb{X}_i, U_i \in \mathbb{U}_i, & i = 1, \dots, M, \\ x_{0,i} = x_i(t), & i = 1, \dots, M, \\ \sum_{i=1}^M [U_i' \ X_i'] K_i \preceq K, \end{cases} \end{aligned} \quad (2)$$

where index  $i \in \{1, \dots, M\}$  denotes the subsystem,  $R_i = R_i' \in \mathbb{R}^{N n_u}$ ,  $R_i \succ 0$  and  $Q_i = Q_i' \in \mathbb{R}^{(N+1)n_x}$ ,  $Q_i \succeq 0$  are cost weighting matrices; vectors  $X_i := [x'_{0,i} \ \dots \ x'_{N,i}]'$  and  $U_i := [u'_{0,i} \ \dots \ u'_{N-1,i}]'$  collect the subsystem states and inputs over the prediction horizon; matrices  $A_i \in \mathbb{R}^{(N+1)n_x \times n_x}$  and  $B_i \in \mathbb{R}^{(N+1)n_x \times N n_u}$  describe the subsystem dynamics (obtained from  $A_i$ ,  $B_i$ , see [6]);  $\mathbb{X}_i \subset \mathbb{R}^{(N+1)n_x}$  and  $\mathbb{U}_i \subset \mathbb{R}^{N n_u}$  are polyhedra that describe the subsystem state and input constraints. The last (in)equality in (2), with  $K_i \in \mathbb{R}^{(N n_u + (N+1)n_x) \times n_k}$  and  $K \in \mathbb{R}^{1 \times n_k}$ , is considered component-wise. It describes  $n_k$  *coupling constraints* between subsystems.

In this paper we use the following:

*Assumption 1:* The CFTOC problem (2) has one coupling constraint, i.e.  $n_k = 1$ ,  $K_i \in \mathbb{R}^{N n_u + (N+1)n_x}$ ,  $K \in \mathbb{R}$ . Clearly, Assumption 1 limits the applicability to the systems in which the coupling between subsystems can be described by (at most) one linear combination of all states and inputs. Such optimization problems typically arise in resource allocation problems (see e.g. [7]) and distributed production problems (see e.g. [8]). An example of the optimal distributed production problem is the wind farm control problem –

the wind farm controller needs to optimally distribute the production between individual wind turbines in such manner that the power required by the grid is met, while the loads of individual wind turbines (that depend on the wind speed at its location) are minimised. This example is further described in Sec. IV.

## III. CONTROLLER DESIGN

The classical MPC approach consists in solving (2) at every time sample  $t$  and applying the first element of the optimizer  $U_i^*$  to  $i$ -th subsystem (1), i.e. using the control law  $u_i(t) = u_{0,i}^*$ ,  $i = 1, \dots, M$ .

The proposed approach for solving large CFTOC problems utilizes a decomposition of the overall control problem (2) into: (i) smaller decoupled local problems related to individual subsystems, and (ii) a centralized optimization problem that coordinates individual subsystems. The local problems are parameterised and solved off-line to reduce the on-line computation burden. The proposed approach obtains the same solution as the centralized approach, i.e. global optimum of (2) is attained.

### A. Parametrisation and decomposition of the optimization program

We define two types of parameters related to the  $i$ -th subsystem: local parameters  $\Phi_i \in \mathbb{R}^{n_\Phi}$  and a coordination parameter  $\Theta_i \in \mathbb{R}$ . The local parameters  $\Phi_i$  lump all (time-varying) data that are locally available (obtained from measurements/estimations) at time  $t$ :

$$\Phi_i := x_{0,i}. \quad (3)$$

The coordination parameter  $\Theta_i$  describes the contribution of  $i$ -th subsystem to the coupling constraint and is defined as:

$$\Theta_i := [U_i' \ X_i'] \cdot K_i. \quad (4)$$

Now, with straightforward algebraic manipulation, we can recast the CFTOC problem (2) as an optimization problem (in new variables) of the following form:

$$\begin{aligned} \min_{\bar{U}_1, \dots, \bar{U}_M, \Theta_1, \dots, \Theta_M} \quad & \sum_{i=1}^M [\Phi_i' \ \Theta_i] Q_{pp,i} [\Phi_i' \ \Theta_i]' + \\ & + \bar{U}_i' Q_{uu,i} \bar{U}_i + [\Phi_i' \ \Theta_i] Q_{pu,i} \bar{U}_i, \\ \text{subject to} \quad & \begin{cases} C_{u,i} \bar{U}_i \leq C_{c,i} + C_{p,i} [\Phi_i' \ \Theta_i]', \\ i = 1, \dots, M, \\ \sum_{i=1}^M \Theta_i \preceq K, \end{cases} \end{aligned} \quad (5)$$

where  $Q_{pp,i} \in \mathbb{R}^{(n_\Phi+1) \times (n_\Phi+1)}$ ,  $Q_{pp,i} = Q'_{pp,i} \succeq 0$ ,  $Q_{uu,i} \in \mathbb{R}^{(N n_u - 1) \times (N n_u - 1)}$ ,  $Q_{uu,i} = Q'_{uu,i} \succ 0$ ,  $Q_{pu,i} \in \mathbb{R}^{(n_\Phi+1) \times (N n_u - 1)}$  are the matrices describing the cost function, which are obtained by inserting the state transition equation (1) and parameter definitions (3) and (4) into the cost (2),  $C_{u,i} \in \mathbb{R}^{n_c \times (N n_u - 1)}$ ,  $C_{c,i} \in \mathbb{R}^{n_c}$ ,  $C_{p,i} \in \mathbb{R}^{n_c \times (n_\Phi+1)}$  are matrices that describe the feasible space of state, input and parameter, obtained from  $\mathbb{X}_i$  and  $\mathbb{U}_i$ , using the state transition equation and parameter definitions. The optimization

vector  $\bar{U}_i$  is obtained by eliminating one component from  $U_i$ , according to (4). Thus  $\Theta_i$  becomes an optimization variable of the problem. This variable substitution does not change the solution of the optimization problem, i.e. the solution  $U_i^*$  of (2) can be trivially obtained from solutions of (5),  $\bar{U}_i^*$  and  $\Theta_i^*$ .

*Remark 1:* The local parameters  $\Phi_i$  do not necessarily have to be defined as in (3). It is sufficient that the definition of  $\Phi_i$  is such that the CFTOC problem (2) can be recasted as (5). Such an example is given in Section IV.

Now we define the  $i$ -th *local problem* as:

$$\begin{aligned} J_i^*(\Phi_i, \Theta_i) &= [\Phi_i' \Theta_i] Q_{pp,i} [\Phi_i' \Theta_i]' + \\ &+ \min_{\bar{U}_i} \bar{U}_i' Q_{uu,i} \bar{U}_i + [\Phi_i' \Theta_i] Q_{pu,i} \bar{U}_i, \quad (6) \\ \text{subject to} \quad &C_{u,i} \bar{U}_i \leq C_{c,i} + C_{p,i} [\Phi_i' \Theta_i]'. \end{aligned}$$

The optimization problem (6) belongs to a class of multi-parametric Quadratic Programs (mp-QP) with parameter  $[\Phi_i' \Theta_i]$ . Note that in (6)  $\Theta_i$  is treated as a parameter, while in (5) it is one of the optimization variables. Hence, the problem (6) must be solved *for all* admissible  $\Theta_i$ , while the optimal coordination parameters  $\Theta_i^*$  are determined at the coordination level, as is explained later in this section. Utilizing the multi-parametric optimization approach [9], the local problems can be solved parameterically, to obtain a closed-form description of the optimizer and the optimal cost function (value function). The properties of the optimizer and value function are described in the following theorem.

*Theorem 1 ([3]):* Consider the mp-QP (6), where  $Q_{uu,i}' \succ 0$  and  $Q_{pp,i} = Q_{pp,i}' \succeq 0$ . The set of feasible parameters  $\mathcal{P}_i := \{[\Phi_i' \Theta_i]' \mid \exists \bar{U}_i : C_{u,i} \bar{U}_i \leq C_{c,i} + C_{p,i} [\Phi_i' \Theta_i]'\}$  is a polyhedral set, the value function  $J_i^* : \mathcal{P}_i \rightarrow \mathbb{R}$  is a piecewise quadratic function on polyhedra (PPWQ), convex and continuous, and the optimizer  $\bar{U}_i^* : \mathcal{P}_i \rightarrow \mathbb{R}^{N_{u,i}-1}$  is a continuous piecewise affine function on polyhedra (PPWA).

By using the value functions of local problems,  $J_i^*(\Phi_i, \Theta_i)$ ,  $i = 1, \dots, M$ , the optimization problem (5) can be stated as the following *coordination problem*:

$$\begin{aligned} \min_{\Theta_1, \dots, \Theta_M} \quad &\sum_{i=1}^M J_i^*(\Phi_i, \Theta_i), \quad (7) \\ \text{subject to} \quad &\begin{cases} [\Phi_i' \Theta_i]' \in \mathcal{P}_i, \\ \sum_{i=1}^M \Theta_i \preceq K. \end{cases} \end{aligned}$$

## B. Controller implementation

The controller implementation consists of two parts: pre-computation done off-line and an algorithm that is run on-line. Off-line, the local problems (6) are solved parameterically to obtain  $\bar{U}_i^*(\Phi_i, \Theta_i)$  and  $J_i^*(\Phi_i, \Theta_i)$ . The algorithms for solving mp-QPs are well researched and readily available in software, see e.g. [9].

On-line, at every time sample  $t$ , the computation is done in the following steps:

- 1) *Local evaluation:* Obtain measurements/estimations of local parameters  $\hat{\Phi}_i$  and evaluate the local solutions for

$\hat{\Phi}_i$ :

$$\begin{aligned} \tilde{U}_i(\Theta_i) &:= \bar{U}_i^*(\hat{\Phi}_i, \Theta_i), \quad \tilde{U}_i : \mathcal{I}_i \rightarrow \mathbb{R}^{N_{u,i}-1}, \\ \tilde{J}_i(\Theta_i) &:= J_i^*(\hat{\Phi}_i, \Theta_i), \quad \tilde{J}_i : \mathcal{I}_i \rightarrow \mathbb{R}, \end{aligned} \quad (8)$$

where  $\mathcal{I}_i$  is an interval in  $\mathbb{R}$ , obtained by projecting  $\mathcal{P}_i$  on  $\Phi_i = \hat{\Phi}_i$ .

- 2) *Coordination:* Given the evaluated local solutions (8), solve the coordination problem to obtain the optimal value of the coordination parameters  $\Theta_1^*, \dots, \Theta_M^*$ :

$$\begin{aligned} \min_{\Theta_1, \dots, \Theta_M} \quad &\sum_{i=1}^M \tilde{J}_i(\Theta_i) \\ \text{subject to} \quad &\begin{cases} \Theta_i \in \mathcal{I}_i, \quad i = 1, \dots, M, \\ \sum_{i=1}^M \Theta_i \preceq K. \end{cases} \end{aligned} \quad (9)$$

- 3) *Local evaluation:* Evaluate the optimal control input for the  $i$ -th subsystem:

$$\bar{U}_i^* := \tilde{U}_i(\Theta_i^*), \quad (10)$$

and apply  $u_i(t) = u_{0,i}^*$  as a control input to the  $i$ -th subsystem. (Note that  $u_{0,i}^*$  is either contained in  $\bar{U}_i^*$  or can be straightforwardly computed from  $\Theta_i^*$  and  $\bar{U}_i^*$ .)

The overview of the on-line steps is given in Figure 1. The local evaluation is done in parallel on the local (subsystems) controllers, while the coordination is done on a centralized hardware. The local controllers send the description of the cost function  $\tilde{J}_i$  to the centralized control hardware ( $3n_{\tau,i} + 1$  numbers), while the centralized controller returns the value of the optimal coordination parameter (1 number).

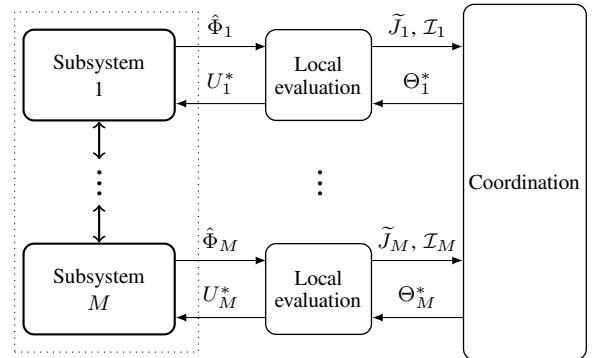


Fig. 1. MPC controller implementation

The local computation comes down to evaluation of PPWA and PPWQ functions, which is a relatively simple task that can be solved efficiently. The challenge lies in the design of an efficient solver for (9). Further on we describe an extremely efficient way of solving the coordination problem by utilizing the specific structure of the problem at hand. Note that, up to this point, we have not utilized Assumption 1.

1) *Solving the coordination problem:* The first interesting property of (9) arises from the fact that the function  $\tilde{J}_i(\Theta_i)$  inherits the properties of the  $J_i^*(\Phi_i, \Theta_i)$  – it is a convex and continuous PPWQ function defined on a polyhedral partition

$\mathcal{I}_i$ . Therefore, the program (9) can be cast as a convex mixed-integer quadratic program and the globally optimal solution can be obtained by solving a series of QPs, in the manner we outline in [10].

If Assumption 1 is fulfilled then a different approach can be pursued. We utilise the fact that  $\tilde{J}_i$  is a one-dimensional function defined on intervals:

$$\begin{aligned} \tilde{J}_i : \mathcal{I}_i &\rightarrow \mathbb{R}, \mathcal{I}_i = \{I_{i,j-1} \leq \Theta_i \leq I_{i,j}\}_{j=1}^{n_{r,i}}, \\ \forall \Theta_i \in [I_{i,j-1}, I_{i,j}] : \tilde{J}_i(\Theta_i) &:= \frac{1}{2}a_{i,j}\Theta_i^2 + b_{i,j}\Theta_i + c_{i,j}, \end{aligned} \quad (11)$$

where  $a_{i,j}, b_{i,j}, c_{i,j}, I_{i,j} \in \mathbb{R}, i = 1, \dots, M, j = 1, \dots, n_{r,i}$ . The cost function of (9) is a PPWQ function defined on a Cartesian product of intervals, i.e. on hyperrectangles of dimension  $M$  arranged on a grid. The solutions are constrained to an intersection of a box defined by  $\Theta_i \in \mathcal{I}_i$  and an  $(M-1)$ -dimensional subspace (in case the coordination requirements is given as an equality) or an  $M$ -dimensional subspace (in case the coordination requirements is given as an inequality).

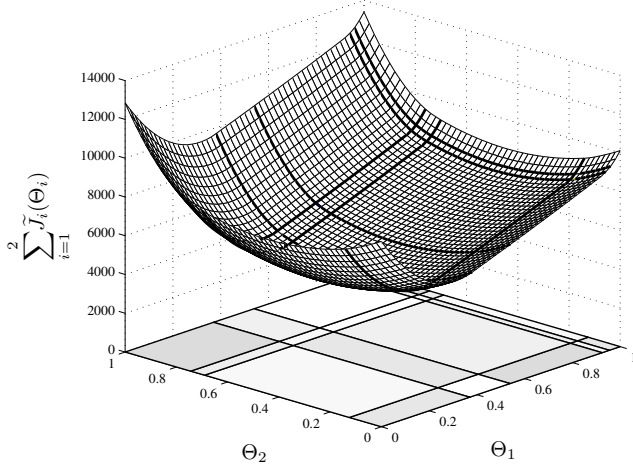


Fig. 2. Piecewise quadratic function defined on Cartesian product of intervals.

**Proposition 1:** Let  $f(z), f : \mathcal{Z} \rightarrow \mathbb{R}$ , be a (strictly) convex piecewise quadratic function defined on a partition of the interval  $\mathcal{Z} = [Z_0, Z_N] \subset \mathbb{R}$  as:

$$f(z) = f_i(z) := \frac{1}{2}h_i z^2 + g_i z + y_i, \quad \text{if } z \in [Z_{i-1}, Z_i], \quad i = 1, \dots, N_z, \quad (12)$$

where  $h_i \in \mathbb{R}, h_i > 0, g_i \in \mathbb{R}, y_i \in \mathbb{R}$  and  $N_z < \infty$ . Let  $z^*$  be the optimizer of the function  $f(z)$ , i.e.

$$z^* := \underset{z}{\operatorname{arg\,min}} f(z), \quad \text{subject to } z \in [Z_0, Z_{N_z}]. \quad (13)$$

Then

$$z^* = Z_0 + \sum_{i=1}^{N_z} w_i^*, \quad (14)$$

where  $w_i^*, i = 1, \dots, N_z$ , is the optimizer of the following convex quadratic program:

$$\begin{aligned} \min_{w_i} \quad & \frac{1}{2}h_i w_i^2 + (h_i Z_{i-1} + g_i)w_i \\ \text{subject to} \quad & 0 \leq w_i \leq Z_i - Z_{i-1}. \end{aligned} \quad (15)$$

*Proof:* Let the optimizer  $z^*$  belong to the  $j$ -th region, i.e.  $Z_{j-1} \leq z^* \leq Z_j$ . Let  $z_i^*$  be the optimizer of the following QP:

$$\begin{aligned} \min_{z_i} \quad & f_i(z_i), \\ \text{subject to} \quad & Z_{i-1} \leq z_i \leq Z_i. \end{aligned} \quad (16)$$

- 1) For all  $i < j$  we have  $z_i^* = Z_i$  since  $f_i(\cdot)$  is monotonically decreasing function in  $z_i, \forall i < j$ .
- 2) For  $i = j$  we have  $z_i^* = z_j^*$ .
- 3) For all  $i > j$  we have  $z_i^* = Z_{i-1}$  since  $f_i(\cdot)$  is monotonically increasing function in  $z_i, \forall i > j$ .

With affine transformation of variables

$$w_i := z_i - Z_{i-1} \quad (17)$$

the optimization (16) becomes equivalent to (15) (when we disregard the constant term). Consequently, from 1), 2) and 3) it follows that

$$\begin{aligned} \sum_{i=1}^{N_z} w_i^* &= \sum_{i=1}^{j-1} (z_i^* - Z_{i-1}) + (z_j^* - Z_{j-1}) + \sum_{i=j+1}^{N_z} (z_i^* - Z_{i-1}) = \\ &= \sum_{i=1}^{j-1} (Z_i - Z_{i-1}) + (z_j^* - Z_{j-1}) + \sum_{i=j+1}^{N_z} (Z_{i-1} - Z_{i-1}) = \\ &= z_j^* - Z_0. \end{aligned} \quad (18)$$

Proposition (1) can be readily utilized to recast (9) as:

$$\begin{aligned} \min_{\Theta_{1,1}, \dots, \Theta_{1,n_{r,1}}, \dots, \Theta_{M,1}, \dots, \Theta_{M,n_{r,M}}} \quad & \sum_{i=1}^M \sum_{j=1}^{n_{r,i}} \frac{1}{2}a_{i,j}\Theta_{i,j}^2 + \\ & + (a_{i,j}I_{i,j-1} + b_{i,j})\Theta_{i,j}, \\ \text{subject to} \quad & \begin{cases} 0 \leq \Theta_{i,j} \leq I_{i,j} - I_{i,j-1}, \\ j = 1, \dots, n_{r,i} \\ i = 1, \dots, M, \\ \sum_{i=1}^M \sum_{j=1}^{n_{r,i}} \Theta_{i,j} + I_{i,0} \leq K, \end{cases} \end{aligned} \quad (19)$$

and:

$$\Theta_i^* = I_{i,0} + \sum_{j=1}^{n_{r,i}} \Theta_{i,j}^*, \quad (20)$$

where  $\Theta_{i,j}^*$  are optimizers of (19). According to Proposition 1,  $\Theta_i^*$  obtained from (20) are optimizers of (7).

The optimization programs (19) and (2) are convex quadratic programs (QPs). In practical examples it can be expected that (19) and (2) have similar size of optimization vectors. Problem (2) has  $\sum_{i=1}^M N n_u$  optimization variables, while (19) has  $\sum_{i=1}^M n_{r,i}$ . The number of constraints is expected to be lower for the problem (19) in comparison to (2): (19) has  $\sum_{i=1}^M 2n_{r,i} + 1$  constraints, while (2) has  $\sum_{i=1}^M 2N n_u + 2(N+1)n_x + 1$  constraints. However, the largest benefit in terms of computational efficiency is obtained from the fact that the problem (19) has a specific structure that enables the use of very efficient solvers.

The program (19) has a separable cost function, separable box constraints and one linear (in)equality constraint. This type of QP is recognised in literature as a continuous

quadratic knapsack problem. There are two very efficient approaches for solving this type of problems known in literature: breakpoint searching algorithm ([11],[12]) and variable fixing algorithm ([13]). Defining  $n := \sum_{i=1}^M n_{r,i}$ , the breakpoint searching algorithm solves (19) in  $\mathcal{O}(n)$  steps, where each step uses a median search. The variable fixing algorithm is simpler implementation-wise since it requires no sorting nor median operations, only elementary algebra<sup>1</sup>. On average the algorithm requires  $\mathcal{O}(n)$  steps and the worst case performance is  $\mathcal{O}(n^2)$ . According to [13] the variable fixing algorithm outperforms the breakpoint searching algorithm in terms of average run time. In the following example we use the variable fixing algorithm to solve (19).

#### IV. EXAMPLE: WIND FARM CONTROLLER

The efficiency of the proposed controller implementation is demonstrated on a design of an optimal wind farm controller. We tackled the problem of optimal wind farm controller design in our previous work: the modeling details can be found in [14] and the formulation details in [15] and [16]. Here, we provide results on the efficiency of the proposed controller design for such control problem.

The wind farm controller needs to enable the control of a wind farm according to requirements from the electrical grid. To fulfill the grid requirements a wind farm controller needs to control the output power of the wind farm in such manner that the wind farm power reference issued by the Transmission System Operator (TSO) is tracked, as well as possible given the constraints imposed by the available power from the wind at given moment. The wind farm controller task is to distribute the power references to individual wind turbines. The local control system ensures the wind turbine power references are tracked if the required power is available from wind. When the wind farm tracks the provided wind farm power reference, which is lower than the currently available power, the power production can be distributed between individual wind turbines in a number of different ways.

In [16] we showed that if, instead of providing time-invariant power references, the power references of individual wind turbines are varied, i.e. adapted to turbulent wind on a particular wind turbine, the wind turbine structural loads can be significantly reduced. Such reduction of structural loads is very beneficial for wind farm operation since it enables longer operational lifetime of turbines, less malfunctions and possibly cheaper turbine design. However, the wind turbine power references can not be perturbed independently on every wind turbine because the wind farm power reference in such case would not be tracked. Therefore, the wind farm controller is designed to coordinate the power perturbations on different wind turbines in such manner that the output wind farm power tracks the provided wind farm power reference, while the power on the turbines are adapted to disturbances from wind. Such controller is easily designed as an MPC controller.

<sup>1</sup>It should be noted that to pose the problem (19) one does not need to sort the intervals of  $\mathcal{I}_i$ .

The real-time implementation of the MPC controller is challenging since the wind farms built today consist of several hundreds of turbines, while, in order to enable structural loads reduction, the sampling time of the controller needs to be kept small (1 second sampling time is chosen). The overall wind farm control problem can be cast to a form (2), given a slight modification that enables the tracking formulation, see [10]. The local parameter contains the initial state of the system, the references for inputs and outputs and the constraint on the maximal available power (which changes with the wind speed). The coordination parameter is the power produced at individual wind turbines, which is required to be equal to the provided reference.

In the following we compare the computation time required to compute the solution to a classical on-line MPC implementation and the proposed approach. The comparison is based on a number of simulations for different wind farm sizes (i.e. different number of wind turbines in a wind farm). The simulations are done using Matlab 7.11.0 (R2010b) on a personal computer with Intel(R)Core 2 Quad CPU processor on 2.4 GHz with 3 GB RAM, on Windows XP operating system.

A classical approach (solving a QP (2) at every sampling instant) is tested using a compiled solver qpOASES [17], developed to particular suite MPC applications. The simulations are done for wind farms consisting of 25, 50 and 75 wind turbines. The wind farm simulation for random data was run for 300 seconds (i.e. 300 computations of optimal control input are done). The obtained computation time is depicted in Figure 3. It can be seen that, given the sampling time of 1 second, such computation approach can not be used at wind farms with already 50 wind turbines, which is a modest size for the current wind farm offshore projects.

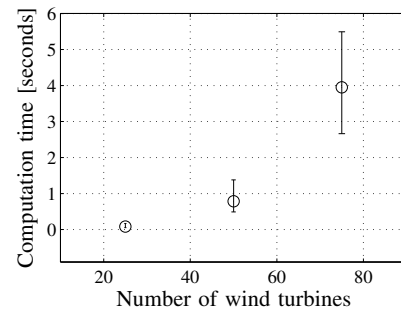


Fig. 3. Computation time for the classical computation approach. Circles denote mean computation time, while errorbars denote the range of obtained computation times during simulation.

The test results of the proposed approach are shown in Figures 4 and 5. Figure 4 shows the histogram of computation times per instance of local evaluation (i.e. for evaluation on one wind turbine) of  $\tilde{J}_i$  and  $\tilde{U}_i$ . This plot is not depicted as a function of the wind farm size since the length of this computation does not depend on the farm size. In a very large number of local computations performed the computation time never exceeded 2.5 milliseconds, while the average computation time is around 1 millisecond.

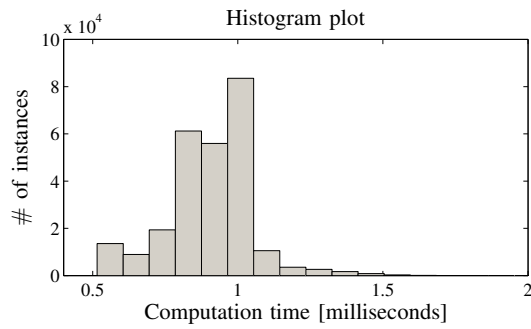


Fig. 4. Histogram of the computation time for the local evaluation.

The computation time for solving the coordination program (19) is given in Figure 5, for a range of wind farm sizes up to 1000 wind turbines. To reduce the number of off-line computations, for each wind farm size the operating points are uniquely determined and then the computations are done for random data to simulate measurements from turbines. A slight bias noticed at Figure 5 occurs due to that reason – at different operating points the wind turbine dynamics is different, which changes the local solutions and therefore also  $n_r^i$ . The biases are entirely lost when the data is plotted against  $\sum_{i=1}^M n_r^i$  and the linear increase in computation time is clear. The obtained computation times are three orders of magnitude smaller than the ones obtained by the classical approach, which means that the proposed approach is very suitable for this application.

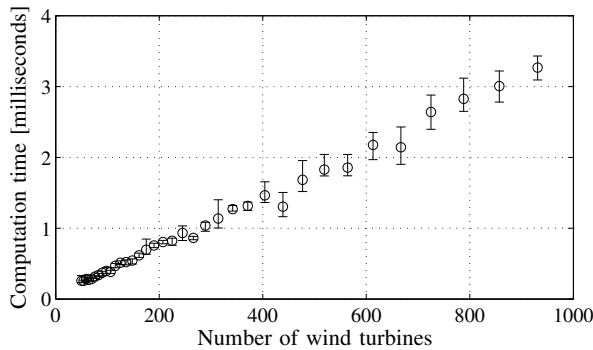


Fig. 5. Computation time for solving the coordination program (19). Circles denote mean computation time, while errorbars denote the range of obtained computation times.

## V. CONCLUSION

In this paper we described an efficient implementation for the MPC controller for coordinated control of a class of large-scale distributed systems. The proposed method consists in distribution and parametrisation of the overall control problem, which enables a significant part of the computation burden to be transferred off-line. The on-line computation is done on the central control hardware that communicates to all subsystems. There is no conservatism in the solution in comparison to the classical centralized approach.

Using an example of wind farm controller design it is shown that this implementation can lead to drastic reduction of on-line computation time – the on-line computation time for the proposed approach is three orders of magnitude smaller than the on-line computation time when the classical MPC controller implementation is used. Therefore, this approach provides an opportunity for optimal control of large-scale systems with constraints at small sampling times.

The described approach is limited to the class of MPC problems with a single coupling constraint. Since it utilizes the particular structure of the problem the extension to a broader problem class is not trivial.

## REFERENCES

- [1] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [2] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [3] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [4] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, “Distributed model predictive control,” *IEEE Control Systems Magazine*, pp. 44–52, February 2002.
- [5] P. D. Christofides, R. Scattolini, D. M. de la Peña, and J. Liu, “Distributed model predictive control: A tutorial review and future research directions,” *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.
- [6] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education Limited, Essex, 2002.
- [7] H. Voos, *Multiagent Systems*. I-Tech, 2009, ch. Agent-Based Distributed Resource Allocation in Continuous Dynamic Systems.
- [8] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*, second edition ed. John Wiley & Sons, 1996.
- [9] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari, “Multi Parametric Toolbox (MPT),” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, Volume 2993. Springer Verlag, March 2004, pp. 448–462.
- [10] V. Spudić, M. Baotić, and M. Jelavić, “Wind farm load reduction via parametric programming based controller design,” in *Proceedings of the 18th IFAC World Congress*, 2011.
- [11] P. Brucker, “An  $O(n)$  algorithm for quadratic knapsack problems,” *Operations Research Letters*, vol. 3, no. 3, pp. 163–166, 1984.
- [12] K. C. Kiwiel, “Breakpoint searching algorithms for the continuous quadratic knapsack problem,” *Mathematical Programming*, vol. 112, no. 2, pp. 473–491, 2008.
- [13] —, “Variable fixing algorithms for the continuous quadratic knapsack problem,” *Journal of Optimization Theory and Applications*, vol. 136, no. 3, pp. 445–458, 2008.
- [14] V. Spudić, M. Jelavić, and M. Baotić, “Wind turbine power references in coordinated control of wind farms,” *Automatika - Journal for control, measurements, electronics, computing and communications*, vol. 52, no. 2, pp. 82–94, 2011.
- [15] V. Spudić, M. Jelavić, M. Baotić, and N. Perić, “Hierarchical wind farm control for power/load optimization,” in *Torque 2010: The science of making torque from wind*, S. Voutsinas and T. Chaviaropoulos, Eds., Heraklion, Crete, Greece, 2010, pp. 681–692.
- [16] V. Spudić, M. Jelavić, and M. Baotić, “Explicit model predictive control for reduction of wind turbine structural loads,” in *51st IEEE Conference on Decision and Control*, Maui, HI, USA, 2010, pp. 1721–1726.
- [17] H. J. Ferreau, A. Potschka, and C. K. et al., “qpOASES software v2.0,” K.U.Leuven Optimization in Engineering Center, <http://www.kuleuven.be/optec/software/qpOASES>, 2010.