Application of the proximal center decomposition method to distributed model predictive control

Ion Necoara, Dang Doan and Johan A.K. Suykens

***Abstract*— In this paper we present a dual-based decomposition method, called here the proximal center method, to solve distributed model predictive control (MPC) problems for coupled dynamical systems but with decoupled cost and constraints. We show that the centralized MPC problem can be recast as a separable convex problem for which our method can be applied. In [9] we have provided convergence proofs and efficiency estimates for the proximal center method which improves with one order of magnitude the bounds on the number of iterations of the classical dual subgradient method. The new method is suitable for application to distributed MPC since it is highly parallelizable, each subsystem uses local information and the coordination between the local MPC controllers is performed via the Lagrange multipliers corresponding to the coupled dynamics. Simulation results are also included.**

I. INTRODUCTION

Model predictive control (MPC) is one of the most successful advanced control technology implemented in industry due to its ability to handle complex systems with hard input and state constraints [6]–[8]. The essence of MPC is to determine a control profile that optimizes a cost criterion over a prediction window and then to apply this control profile until new process measurements become available. Then the whole procedure is repeated. Feedback is incorporated by using the measurements to update the optimization problem for the next step.

For the control problem of large-scale networked systems, centralized MPC is considered impractical, inflexible and unsuitable due to information requirement and computational aspects. The subsystems in the network may have different authorities that prevent sending all necessary information to one processing center. Moreover, the optimization problem yielded by centralized MPC is too big for real time computation. Networks of vehicles, production units in a power plant, networks of cameras in an airport are just a few examples. Distributed MPC is proposed for control of such large-scale systems, by decomposing the overall system into small subsystems with distinct MPC controllers for each subsystem that collaborate to achieve global decisions. In order to derive the local MPC controllers we decompose the MPC problem into a set of subproblems each solved by an individual agent using local information. The coordination of the subproblems is achieved by an active communication among the agents.

I. Necoara and J.A.K. Suykens are with the Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT–SCD, Kasteelpark Arenberg 10, B–3001 Leuven (Heverlee), Belgium. {ion.necoara,johan.suykens}@esat.kuleuven.be
Dang Doan is with the Technische Universiteit Delft, Delft Center of Systems and Control, Mekelweg 2, 2628 CD, Delft, The Netherlands. m.d.doan@student.tudelft.nl

Approaches to distributed MPC design differ from each other in the problem's setup. In [5], Camponogara *et al.* studied stability of coordination-based distributed MPC with several information exchange conditions. In [4], Dunbar and Murray proposed a distributed MPC scheme for problems with coupled cost function, utilizing prediction trajectories of the neighbors in each subsystem's optimization. Keviczky *et al.* proposed a distributed MPC scheme with a sufficient stability test for dynamically decoupled systems in [13], in which each subsystem optimizes the behaviors of its neighbors. Richards and How in [12] proposed a robust distributed MPC method for networks with coupled constraints, based on constraint tightening and serial approach.

A distributed MPC scheme for dynamically coupled systems was proposed by Venkat *et al.* in [1], [15], based on a parallel synchronous approach, i.e. iterating the Jacobi algorithm $p_{\max}$ times [2]. But, there is no guarantee of the Jacobi algorithm about how good is the approximation of the optimum after $p_{\max}$ iterations and moreover we need strictly convex functions to prove asymptotic convergence to the optimum. However, if we solve the MPC problem using the algorithm proposed in the present paper, we have a guaranteed upper bound on the approximation of the optimum after $p_{\max}$ iterations and it can be applied to general convex functions.

In this paper we explore the potential of the proximal center decomposition method for separable convex programs proposed in [9] in distributed MPC problems for dynamically coupled systems with decoupled constraints and cost. We show that the distributed MPC problem corresponding to linear systems with interacting subsystem dynamics and decoupled costs can be recast in the framework of separable convex problems for which our algorithm can be applied. The algorithm involves every agent optimizing an objective function that is the sum of his own objective function and a smoothing term while the coordination between agents is performed via the Lagrange multipliers. We show that the solution of our distributed proximal center algorithm converges to the solution of the centralized MPC problem and we also provide estimates for the rate of convergence. In [9] we were able to prove that the efficiency estimates for the new method improves the bounds on the number of iterations of the classical dual subgradient scheme by an order of magnitude (see [10] for more details). Therefore, the proximal center MPC algorithm is suitable for online implementation.

The layout of the paper is as follows. In Section II we define the centralized MPC problem followed by the decentralized formulation, i.e. the division of the general problem into decentralized subproblems. We show that the centralized MPC can be recast as a separable convex program. In Section III we describe briefly a dual-based decomposition method for separable convex problems developed recently in [9]. Finally, numerical simulations are included to compare the new approach with the approach in [1].

II. Distributed model predictive control

The application that we will discuss in this section is decentralized control of large-scale systems with interacting subsystem dynamics, which can be found in a broad spectrum of applications ranging from robotics to regulator systems. Distributed MPC is promising in applications for large-scale production systems in the factories, in water or electric distribution and transportation networks. In such applications there are not only the difficulties caused by complex interacting dynamics, but also the limitation of information structure due to organizational aspects. MPC or other centralized optimal control methods still cannot deal with these issues. A distributed MPC framework is appealing in this context since this framework allows us to design local subsystem-based controllers that take care of the interactions between different subsystems and physical constraints.

We assume that the overall system model can be decomposed into M appropriate subsystem models:

$$x^i(k+1) = \sum_{j\in\mathcal{N}(i)} A_{ij}x^j(k) + B_{ij}u^j(k)\ \forall i = 1\cdots M, \quad (1)$$

where $\mathcal{N}(i)$ denotes the set of subsystems that interact with the ith subsystem, including itself. The control and state sequence must satisfy local constraints:

$$x^i(k) \in \Omega_i,\ u^i(k) \in U_i\ \ \forall i = 1\cdots M \text{ and } \forall k \geq 0,$$

where the constraint sets $\Omega_i \subseteq \mathbb{R}^{n_{x_i}}$ and $U_i \subseteq \mathbb{R}^{n_{u_i}}$ are usually convex compact sets with the origin in their interior.

Remark 2.1 (i) Note that the settings considered in this paper are more general than those from [15]: we consider a more general model for the coupling dynamics (the states of the neighbors influences also the subsystem i) and moreover we also consider state constraints.

(ii) It is worth nothing that the method that is presented in this paper can also treat coupling inequalities (see [9] for more details).

In general the control objective is to steer the state of the system to origin or any other set point in a "best" way. Performance is expressed via a stage cost, which is composed of individual separate costs assumed to have the following form (see also [15]):

$$\ell(x,u) = \sum_{i=1}^{M} \ell_i(x^i,u^i),$$

where usually $\ell_i(x^i,u^i)$ is a convex quadratic function, but not necessarily strictly convex.

In MPC we must solve at each step k, given $x^i(k) = x^i$, a finite-horizon optimal control problem. The centralized MPC problem for this application is formulated as follows:

$$\begin{aligned}\min_{x_l^i,u_l^i} & \sum_{l=0}^{N-1}\sum_{i=1}^{M} \ell_i(x_l^i,u_l^i) + \sum_{i=1}^{M} \ell_i^f(x_N^i) \quad (2)\\ \text{s.t. : } & x_0^i = x^i,\ x_{l+1}^i = \sum_{j\in\mathcal{N}(i)} A_{ij}x_l^j + B_{ij}u_l^j\\ & x_N^i \in \Omega_i,\ x_l^i \in \Omega_i,\ u_l^i \in U_i\ \forall l = 0\cdots N-1,\ \forall i = 1\cdots M,\end{aligned}$$

where N denotes the prediction horizon and $\ell_i^f(x_N^i)$ denotes some terminal cost introduced for stability reasons. Note that a similar formulation of distributed MPC for coupled linear subsystems with decoupled costs was given in [15], but without state constraints (i.e. without imposing $x^i(k) \in \Omega_i$).

The optimization problem (2) becomes interesting if the computations can be distributed among the subsystems (agents) and the amount of information that the agents must exchange is limited. Now, we show that the centralized MPC problem (2) that must be solved at each step k can be recast as a separable convex problem, i.e. separable objective function but with linear coupling constraints, for which the distributed algorithm presented in Section III can be applied. Let us introduce the following notation:

$$\begin{aligned}&\mathbf{x}_i = (x_1^i\cdots x_N^i\ u_0^i\cdots u_{N-1}^i),\ \ X_i = \Omega_i^N \times U_i^N,\\ &\psi_i(\mathbf{x}^i) = \sum_{l=0}^{N-1} \ell_i(x_l^i,u_l^i) + \ell_i^f(x_N^i),\end{aligned}$$

where ψ_i's are convex quadratic functions, not necessarily strictly convex. Then, the control problem (2) can be recast as a separable convex program:

$$\min_{\mathbf{x}_i\in X_i} \big\{ \sum_{i=1}^{M} \psi_i(\mathbf{x}_i) : \sum_{i=1}^{M} C_i\mathbf{x}_i - b = 0\big\}, \quad (3)$$

where the matrices C_i and b are defined accordingly.

In [15] the optimization problem (2) (or equivalently (3)) was solved in a decentralized fashion, iterating the Jacobi algorithm $p_{\max}$ times [2]: i.e. at each iteration p, where $1 \leq p \leq p_{\max}$, solve in parallel for $i_0 = 1\cdots M$

$$\min_{\mathbf{x}_{i_0}\in X_{i_0}} \big\{\psi_{i_0}(\mathbf{x}_{i_0}) : C_{i_0}\mathbf{x}_{i_0} + \sum_{i\neq i_0, i=1}^{M} C_i\mathbf{x}_i^{p-1} - b = 0\big\}, \quad (4)$$

where $\mathbf{x}_i^{p-1}$'s are the values computed at $(p-1)$th iteration for the ith subsystem.

But, there is no theoretical guarantee of the Jacobi algorithm about how good the approximation of the optimum of (3) is after $p_{\max}$ iterations. Moreover, in order to ensure asymptotic convergence of the Jacobi algorithm we need the ψ_i's to be strictly convex (see e.g. [2]), which is not necessarily the case in the MPC settings. However, in the next section we describe briefly a decomposition algorithm for separable convex problems of the form (3) developed recently in [9] which guarantees a priori an upper bound on the approximation of the optimum after $p_{\max}$ iterations and

it applies to general convex functions. Our algorithm can be an alternative to the classical methods (e.g. Jacobi algorithm, dual subgradient method, etc), leading to a new method of solution.

III. A DUAL DECOMPOSITION METHOD FOR SEPARABLE CONVEX PROBLEMS

In this section we describe a dual decomposition method recently introduced in [9] for separable convex problems, in which the Lagrange multipliers are updated according to a first-order optimal method. We also present efficiency estimates of the described method. Throughout the paper $\|\cdot\|$ denotes the Euclidian norm. For simplicity of the exposition of the method we restrict ourselves to problems with only two agents (subsystems), i.e. $M = 2$:

$$f^* = \min_{x_i \in X_i} \{\psi_1(x_1) + \psi_2(x_2) : \ C_1x_1 + C_2x_2 - b = 0\}, \quad (5)$$

where ψ_i are continuous convex functions and X_i are given compact convex sets, $i = 1, 2$.

Remark 3.1 Note that the method developed in this paper can treat also coupled inequalities $C_1x_1 + C_2x_2 \leq b$ (see [9] for more details). This is a very important feature of our method compared to existing distributed MPC algorithms which cannot deal in general with coupled inequalities.

We assume that the constraint qualification condition [2], [9] holds for (5). The dual function associated to (5) is defined as follows:

$$f_0(\lambda) = \min_{x_i \in X_i} \psi_1(x_1) + \psi_2(x_2) + \langle \lambda, C_1x_1 + C_2x_2 - b \rangle,$$

where λ denotes the Lagrange multipliers associated with the equality constraints and $\langle \cdot, \cdot \rangle$ denotes the standard scalar product on some Euclidian space. From standard duality theory, the convex problem (5) is equivalent to solving an unconstrained maximization problem having the objective function f_0. The usual approach is to apply the subgradient method (steepest ascent update of the multipliers) to the dual function f_0 or the Jacobi algorithm directly to the optimization problem (5) (see e.g. [2], [14]). Convergence of these methods can be guaranteed under the assumption that the functions ψ_i's are strictly convex. However, in distributed MPC problems the cost function is not necessarily strictly convex. In the sequel we will describe a dual decomposition method for general convex functions ψ_i's whose efficiency estimates improves with one order of magnitude the bounds on the number of iterations of the classical dual subgradient method (see [9] for more details).

For given compact sets X_i we can choose finite and positive constants D_{X_i} such that

$$D_{X_i} \geq \max_{x_i \in X_i} \|x_i\|^2 \text{ for } i = 1, 2.$$

Let us introduce the following family of functions:

$$f_c(\lambda) = \min_{x_i \in X_i} \psi_1(x_1) + \psi_2(x_2) + \langle \lambda, C_1x_1 + C_2x_2 - b \rangle + c\big(\|x_1\|^2 + \|x_2\|^2\big), \quad (6)$$

where c is a positive *smoothness parameter* that will be defined in the sequel (see Theorem 3.5). Note that by adding the smoothness term $c(\|x_1\|^2 + \|x_2\|^2)$ the objective function in (6) remains separable in x_i, i.e.

$$f_c(\lambda) = -\langle \lambda, b \rangle + \min_{x_1 \in X_1} [\psi_1(x_1) + \langle \lambda, C_1x_1 \rangle + c\|x_1\|^2] + \min_{x_2 \in X_2} [\psi_2(x_2) + \langle \lambda, C_2x_2 \rangle + c\|x_2\|^2]. \quad (7)$$

Denote by $x_i(\lambda)$ the optimal solution of the minimization problems in x_i in (7). Then, the function f_c has the following smoothness properties:

Theorem 3.2: [9] The function f_c is concave and continuously differentiable at any λ. Moreover, its gradient $\nabla f_c(\lambda) = C_1x_1(\lambda) + C_2x_2(\lambda) - b$ is Lipschitz continuous with Lipschitz constant

$$L_c = \frac{\|C_1\|^2}{2c} + \frac{\|C_2\|^2}{2c}.$$

The following inequalities also hold:

$$f_c(\lambda) \geq f_0(\lambda) \geq f_c(\lambda) - c(D_{X_1} + D_{X_2}) \ \ \forall \lambda.$$

We now describe a distributed optimization method for (5), called in [9] the *proximal center algorithm*, that has the nice feature that the coordination between the agents involves the maximization of a smooth convex objective function (i.e. with Lipschitz continuous gradient). Moreover, the resource allocation stage consists in solving in *parallel* by the each agent of a minimization problem with strongly convex objective using only local information. The new method belongs to the class of two-level algorithms [3] and is particularly suitable for separable convex problems where the minimizations over x_i's in (7) are easily carried out.

We apply Nesterov's accelerated method [10], [11], based only on first-order information, to the *unconstrained* maximization problem whose objective function is the concave function f_c that has a Lipschitz continuous gradient:

$$\max_{\lambda} f_c(\lambda). \quad (8)$$

The proximal center algorithm can be described as follows:

Algorithm 3.3: [9] **for** $p \geq 0$ **do**

1. given λ^p compute in *parallel*

$$x_i^{p+1} = \arg \min_{x_i \in X_i} \psi_i(x_i) + \langle \lambda^p, C_ix_i \rangle + c\|x_i\|^2$$

2. compute $\nabla f_c(\lambda^p) = C_1x_1^{p+1} + C_2x_2^{p+1} - b$

3. find

$$u^p = \arg\max_{\lambda} \langle \nabla f_c(\lambda^p), \lambda - \lambda^p \rangle - \frac{L_c}{2}\|\lambda - \lambda^p\|^2$$

4. find

$$v^p = \arg\max_{\lambda} -\frac{L_c}{2}\|\lambda\|^2 + \sum_{l=0}^{p} \frac{l+1}{2} \langle \nabla f_c(\lambda^l), \lambda - \lambda^l \rangle$$

5. set $\lambda^{p+1} = \frac{p+1}{p+3}u^p + \frac{2}{p+3}v^p$.

Note that the maximization problems in Steps 3 and 4 of Algorithm 3.3 can be solved explicitly and thus computationally very efficient. The main computational effort is done in Step 1 of Algorithm 3.3. However, in some

applications, e.g. distributed MPC, Step 1 can be performed also very efficiently (see Section IV), making it suitable for online implementation. The proximal center algorithm can be applied in decomposition since it is highly parallelizable: the agents can solve their corresponding local minimization problems in parallel.

In the next two theorems we show that the solution generated by our distributed proximal center algorithm converges to the solution of the original problem (5) and we provide also estimates for the rate of convergence.

Theorem 3.4: [9] After p iterations we obtain an approximate solution to the problem (5)

$$\hat{x}_i = \sum_{l=0}^{p} \frac{2(l+1)}{(p+1)(p+2)} x_i^{l+1} \text{ and } \hat{\lambda} = \lambda^p$$

$i = 1, 2$, which satisfy the following duality gap:

$$[\psi_1(\hat{x}_1) + \psi_2(\hat{x}_2)] - f_0(\hat{\lambda}) \leq c(D_{X_1} + D_{X_2}) - \max_{\lambda} \big[- \frac{2L_c}{(p+1)^2} \|\lambda\|^2 + \langle C_1\hat{x}_1 + C_2\hat{x}_2 - b, \lambda \rangle \big].$$

Theorem 3.5: Taking $c = \frac{\epsilon}{D_{X_1} + D_{X_2}}$ and

$$p + 1 = 2\sqrt{(\|C_1\|^2 + \|C_2\|^2)(D_{X_1} + D_{X_2})}\, \frac{1}{\epsilon},$$

then after p iterations

$$-\|\lambda^*\| \|C_1\hat{x}_1 + C_2\hat{x}_2 - b\| \leq \psi_1(\hat{x}_1) + \psi_2(\hat{x}_2)] - f^* \leq \epsilon$$

and the constraints satisfy

$$\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\| \leq \epsilon\big(\|\lambda^*\| + \sqrt{\|\lambda^*\|^2 + 2}\big),$$

where λ^* is the minimum norm optimal multiplier.

Proof:

$$\max_{\lambda} -\frac{2L_c}{(p+1)^2}\|\lambda\|^2 + \langle C_1\hat{x}_1 + C_2\hat{x}_2 - b, \lambda \rangle = \frac{(p+1)^2}{8L_c}\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\|^2.$$

We obtain the following bound on the duality gap (see Theorem 3.4):

$$[\psi_1(\hat{x}_1) + \psi_2(\hat{x}_2)] - f_0(\hat{\lambda}) \leq c(D_{X_1} + D_{X_2}) - \frac{(p+1)^2}{8L_c}\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\|^2 \leq c(D_{X_1} + D_{X_2}).$$

It follows that taking $c = \frac{\epsilon}{D_{X_1} + D_{X_2}}$, the duality gap is less than ϵ. For the constraints we get that $\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\|$ satisfies the second order inequality in y (see [9] for more details): $\frac{(p+1)^2}{8L_c}y^2 - \|\lambda^*\|y - \epsilon \leq 0$. Therefore, $\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\|$ must be less than the largest root of the corresponding second-order equation, i.e.

$$\|C_1\hat{x}_1 + C_2\hat{x}_2 - b\| \leq \big(\|\lambda^*\| + \sqrt{\|\lambda^*\|^2 + \frac{\epsilon(p+1)^2}{2L_c}}\big)\frac{4L_c}{(p+1)^2}.$$

After some long but straightforward computations we get that after p iterations, where p defined as in the theorem, we also get the bound on the constraint violation. ■

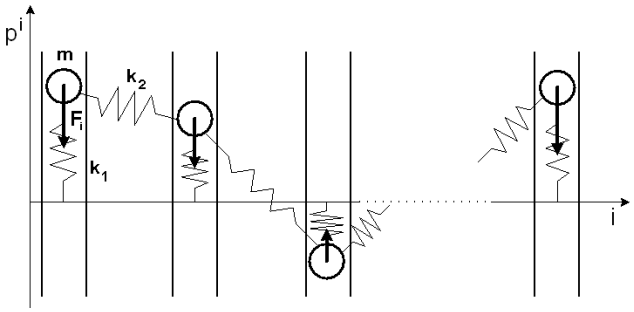

Fig. 1. Setup of coupled oscillators

From Theorem 3.5 we obtain that the complexity of Algorithm 3.3 for finding an ϵ-approximation of the optimum for the centralized MPC problem (2) is of the order $\mathcal{O}(\frac{1}{\epsilon})$, better than most non-smooth optimization schemes such as the dual subgradient method that have an efficiency estimate of the order $\mathcal{O}(\frac{1}{\epsilon^2})$ (see e.g. [10]). The main advantage of our scheme is that it is fully automatic, the parameter c is chosen unambiguously, which is crucial for justifying the convergence properties of Algorithm 3.3. Moreover, the algorithm is suitable for solving distributed MPC problems since the control inputs to each subsystem can be computed based on local information and the information that the agents must exchange is limited. Finally, our method can also take into account coupled inequalities (see [9] for more details) which arise very often in these type of applications (they represent in general shared resources between agents).

IV. Example

A. Problem description

In this section we explore the potential of the Algorithm 3.3 described in Section III in solving a distributed MPC problem corresponding to coupled oscillators. The network is a ring of 20 coupled oscillators that can move only on the vertical axis and the goal is to stabilize the network around the equilibrium, which in this case is the horizontal axis passing through origin. The setup is shown in Figure 1. Each oscillator is considered as one subsystem and is influenced by two neighbors next to it, i.e. $\mathcal{N}(i) = \{i-1, i, i+1\}\ \ \forall i$. The continuous dynamic equations of each subsystem i are considered to have the following form:

$$m\ddot{p}^i = k_1 p^i - f_s\dot{p}^i + k_2[p^{i-1} - p^i] + k_2[p^{i+1} - p^i] + u_i,$$

where p^i denotes the position of oscillator i, u^i denotes a vertical force applied mainly to subsystem i and the parameters are defined as

k_1: stiffness of vertical spring at each oscillator
k_2: stiffness of springs that connect the oscillators
m: mass of each oscillator
f_s: friction coefficient of movements

From a given initial state the system needs to be stabilized subject to the following state and input constraints:

$$\Omega_i = \{x^i : \|x^i\| \leq 2\},\ U_i = \{u^i : \|u^i\| \leq 1\},$$

where we recall that $\|\cdot\|$ denotes the Euclidian norm. Note that in our decomposition method we can consider any compact convex sets Ω_i and U_i not necessarily Euclidian balls. This choice is made only to illustrate the computational advantages of our method in this case.

We choose for each subsystem i the state vector $x^i = [p^i\ \dot{p}^i]^T$ and input u^i, that means each state contains the position and velocity variables of the corresponding subsystem. By choosing sampling time T_s and do discretization, we get discrete linear coupled dynamics for each subsystem which has the formula of general case described in (1).

For this example we formulate the distributed MPC problem as described in Section II, with the specific values:

$$\ell_i(x^i,u^i) = x^{iT}Q_ix^i + u^{iT}R_iu^i,\ \ell_i^f(x^i) = x^{iT}P_ix^i$$
$$P_i = Q_i = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix},\ R_i = 1\ \ \forall i = 1\cdots M$$
$$\Omega_i = \{x^i : \|x^i\| \le 2\},\ U_i = \{u^i : \|u^i\| \le 1\}$$
$$T_s = 0.01s,\ M = 20,\ N = 20,\ p_{\max} = 500$$
$$k_1 = 0.4, k_2 = 0.3, f_s = 0.4, T_s = 0.05, m = 1.$$

Note that with this choice of Q_i we are only interested in penalizing the position of each oscillator. Therefore the stage cost ℓ_i associated to each subsystem i is not a strictly quadratic function in both states and inputs. This situation is encountered in many MPC applications.

Constructing the variables $\mathbf{x}_i$, the sets X_i and the matrices C_i, b as described in Section II, the centralized MPC problem for this example can be rewritten as:

$$\min_{\mathbf{x}^i \in X_i} \{\sum_{i=1}^{M} \mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i : \sum_{i=1}^{M} C_i\mathbf{x}_i - b = 0\}, \tag{9}$$

where $\mathbf{Q}_i = \text{diag}(Q_i, \cdots, Q_i, R_i, \cdots, R_i)$ with N terms of Q_i and N terms of R_i. Note that $X_i \subseteq \{\mathbf{x} : \|\mathbf{x}\| \le r_i\}$, where $r_i = \sqrt{N \cdot 2^2 + N \cdot 1^2}$.

B. Computational complexity

We apply Algorithm 3.3 to solve the separable convex problem (9). Note that the sub-optimization problems occurred in Step 1 of Algorithm 3.3 can be solved very efficiently, due to the structure of (9) (we must solve quadratic programming with quadratic constraints problems (QPQC) with a special structure). Below we describe a fast algorithm to solve this type of QPQC problems. Following Algorithm 3.3, we must first add the *smoothness term* $c\sum_{i=1}^{M}\|\mathbf{x}_i\|_2^2$ to the Lagrangian of (9). Note that in this example $D_{X_i} = r_i^2$. For each iteration p, where $1 \le p \le p_{\max}$, we must solve in Step 1 of the Algorithm 3.3 the following minimization problems:

$$\sum_{i=1}^{M} \min_{\mathbf{x}_i \in X_i} \mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \langle \lambda^p, C_i\mathbf{x}_i\rangle + c\|\mathbf{x}_i\|^2,$$

where the Lagrange multipliers λ^p was computed at previous iteration $p-1$. Observe that each minimization problem is a convex program with separable and strictly convex cost and decoupled constraints. In fact, since $\mathbf{Q}_i$ has a diagonal structure and the variables $\mathbf{x}_i$ are coupled via the linear term $C_i\mathbf{x}_i$ for each i, we can further decompose each minimization problem into $2N$ QPQC problems with a particular structure:

$$\min_{\|x\| \le r} x^TQx + \langle q, x\rangle, \tag{10}$$

where Q is a positive definite diagonal matrix (for our example $Q = Q_i + cI_2$ for the state variables x_l^i or $Q = R_i + c$ for the input variables u_l^i). Here x represents the state or control variable at one step and q contains two or one entry of the vector $C_i^T\lambda^p$ that corresponds to x. Using duality theory we can show that the optimization problem (10) can be solved efficiently:

$$\max_{\mu \ge 0} \min_x x^TQx + \langle q, x\rangle + \mu(\|x\|^2 - r^2)$$

or equivalently

$$\max_{\mu \ge 0} \min_x x^THx + \langle q, x\rangle - \mu r^2, \tag{11}$$

where $H_\mu = Q + \mu I$ is a diagonal matrix and thus its inverse can be computed immediately. Replacing x with $H_\mu^{-1}q$ in (11) we obtain a maximization problem in the scalar variable μ whose optimal solution can be computed easily by solving a fourth (for the state variable x_l^i) or second (for the input variable u_l^i) order scalar equation (e.g. we can solve it quickly by bisection algorithm or analytically). Therefore, our algorithm has the advantage that at each iteration p we must solve MN QPQC's of dimension $n_x (= 2$ for this particular example) and MN QPQC's of dimension n_u $(= 1)$ as in (10). Note that with the Jacobi algorithm we must solve M QPQC's of dimension $Nn_x + Nn_u$ and with Nn_x additional equality constraints (see (4)).

Remark 4.1 From the previous discussion we can observe that in general the proximal center method applied to distributed MPC problems with quadratic stage costs leads to decomposition in both "space" and "time", i.e. the centralized MPC problem can be decomposed into small subproblems corresponding to the spatial structure of the system (M subsystems) but also to the prediction horizon (N the length of the prediction). Note that this is *not* the case with Jacobi algorithm.

Once the optimal μ^* is determined, we substitute its value into $H_{\mu^*}^{-1}$ and find the optimal solution of (10): $x^* = -H_{\mu^*}^{-1}q$. Following the same reasoning as above it is easy to see that the maximization problems in Steps 3 and 4 of Algorithm 3.3 can be solved explicitly and thus computationally very efficient. We also solved the centralized MPC problem (9) with the same parameters, using the Jacobi algorithm described in [15]. The simulation results will be given in the next section.

C. Simulations

We compare the two algorithms (proximal center algorithm 3.3 and the Jacobi algorithm) by looking at how the position of the 10th subsystem evolves and how the global cost of the full system decreases during simulations.

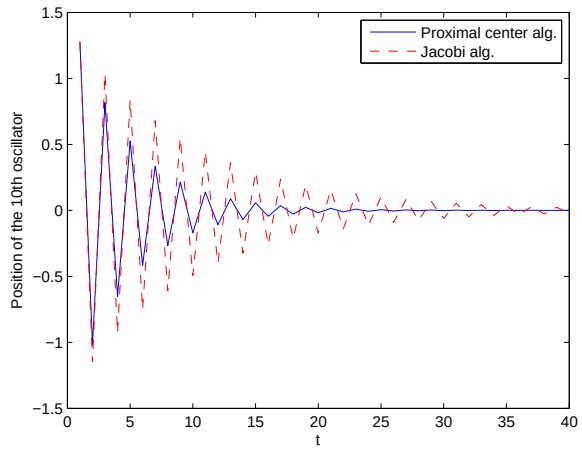

Fig. 2. Evolution of position of the 10th subsystem: full-line proximal center algorithm 3.3, dashed-line Jacobi algorithm [15].

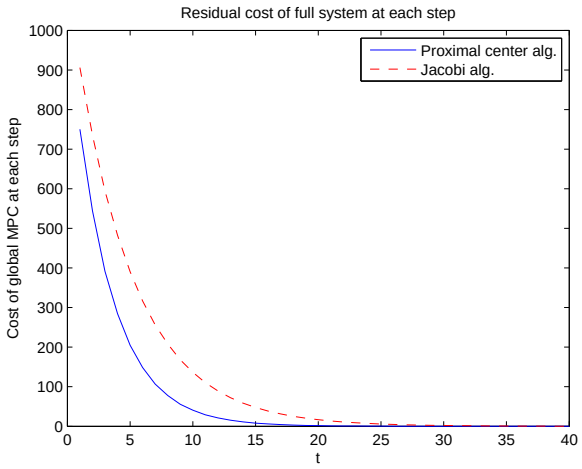

Fig. 3. The global cost: full-line proximal center algorithm 3.3, dashed-line Jacobi algorithm [15].

Figure 2 displays the position of the 10th subsystem over the simulation period. Note that the position of the 10th oscillator converges to the equilibrium faster using the proximal center algorithm than using the Jacobi algorithm. Figure 3 displays the global cost calculated at each step over the simulation period. We note that for the same number of iterations $p_{\max}$ our algorithm produces a better cost than Jacobi algorithm. Computationally, with our non-optimized matlab code we found out that Algorithm 3.3 is also faster than Jacobi algorithm (see also Remark 4.1).

V. Conclusions

The proximal center decomposition method developed in [9] is applied to distributed MPC problems for dynamically coupled subsystems but with decoupled cost and constraints. However, our method can be adapted easily to include also coupled constraints that represents shared resources between agents. It was shown that the centralized MPC problem for this type of coupled subsystems can be rewritten as a separable convex program for which our algorithm can be applied. The algorithm involves every subsystem (agent) optimizing at each step an objective function that is the sum of his own local objective function and a smoothing term while the coordination between the subsystems is performed via the Lagrange multipliers corresponding to the coupled dynamics. We proved that the solution generated by our distributed proximal center algorithm converges to the solution of the centralized problem and we provided also estimates for the rate of convergence. It was also proved that the main steps of the algorithm can be computed efficiently and thus making this method suitable for online implementation of the corresponding distributed MPC scheme. The simulation results confirm that the proposed distributed MPC method works well in practice.

Acknowledgment. We acknowledge the financial support by Research Council K.U. Leuven: GOA AMBioRICS, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07; Research communities (ICCoS, ANMMM, MLDM); AWI: BIL/05/43, IWT: PhD Grants; Belgian Federal Science Policy Office: IUAP DYSCO.

References

[1] J.B. Rawlings A.N. Venkat and S.J. Wright. Stability and optimality of distributed model predictive control. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 6680–6685, 2005.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] G. Cohen. Optimization by decomposition and coordination: A unified approach. *IEEE Transactions on Automatic Control*, AC–23(2):222–232, 1978.

[4] W.B. Dunbar and R.M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42:549–558, April 2006.

[5] B.H. Krogh E. Camponogara, D. Jia and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[6] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice — A survey. *Automatica*, 25(3):335–348, May 1989.

[7] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, England, 2002.

[8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(7):789–814, June 2000.

[9] I. Necoara and J.A.K. Suykens. Application of a smoothing technique to decomposition in convex programming. Technical Report 08–07, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), January 2008, submitted for publication.

[10] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

[11] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming (A)*, 103(1):127–152, 2005.

[12] A.G. Richards and J.P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[13] F. Borrelli T. Keviczky and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42:2105–2115, 2006.

[14] H. Uzawa. Iterative methods for concave programming. In K. Arrow, L. Hurwicz, and H. Uzawa, editors, *Studies in Linear and Nonlinear Programming*, pages 154–165, 1958.

[15] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, to appear, 2007.