Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Separable Model Predictive Control via Alternating Direction Method of Multipliers for Large-scale Systems

Liang Lu *

* *Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden (e-mail: liangup@gmail.com)*
State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China.

Abstract: In this paper, an alternating direction method of multipliers (ADMM) based real-time model predictive control (MPC) algorithm is presented. With the use of indicator function and by introducing extra consensus constraints, the constrained MPC problem can be formulated as a separable MPC problem, which can be computed very efficiently by projected gradient descent ADMM update steps and Riccati recursions. The sequence of the objective value of this constrained real-time ADMM-type MPC algorithm satisfies a linear convergence rate. The procedure is also extended to distributed systems with constraints, in which the variables of each subsystems communicate with their neighbors and update in the Gauss-Seidel way. An illustrative example shows the effectiveness of this approach.

1. INTRODUCTION

Model predictive control (MPC) is a flourishing research field, partly because of its way to handle constraints. At each sampling time, an optimization problem is solved, which results in a sequence of control inputs, of which only the first one is implemented. For constrained linear or piecewise affine systems with a quadratic cost, explicit piecewise affine control laws, so called explicit model predictive control (EMPC), can be obtained by solving a multi-parametric (mixed-integer) quadratic programming (QP) problem offline, and the online implementation boils down to a point location problem. However, explicit MPC laws are often only tractable for small scale systems.

Developing efficient real-time MPC optimization algorithms for higher dimensional systems has been for many years a central topic for MPC research. By exploiting or rearranging the structure in the QPs in the MPC formulation, Wang and Boyd [2010], Rao et al. [1998] proposed structured interior point methods, in which Riccati recursion based block elimination can be utilized to compute each Newton step. Especially in Wang and Boyd [2010] a primal barrier interior point method was employed, and as the aim is to find a control with a warm-starting technique, some simplified solutions by fixing a barrier parameter in the primal barrier interior point method or by fixing the iteration times for each Newton step are also presented to obtain suboptimal MPC controllers. In Richter et al. [2012], the authors gave the execution time certification for Nesterov's fast gradient method, i.e., suboptimal performance bound is derived for early terminating the algorithm iterations at each sampling instant to accelerate the online computation, and Giselsson [2012, 2013a] extended the result to dual formulation with accelerated gradient methods (Beck and Teboulle [2009], Tseng [2008]). In Annergren et al. [2012], the authors propose an alternating direction method of multipliers (ADMM) for ℓ_1 regularized MPC problem.

For large scale systems, by decomposition, we can break the problem into a bunch of small subproblems, which can be easily managed. So solving the problem in parallel, i.e. solving the subproblems iteratively, and communicating cooperatively is preferable (see, for instance, Camponogara et al. [2002], Christofides et al. [2013], Stewart et al. [2010], Scattolini [2009] and the references therein). Dual decomposition (Bertsekas [1999]) and alternating direction method of multipliers (ADMM) (Boyd et al. [2010]) are the two popular distributed optimization schemes. In Giselsson and Rantzer [2010], the distributed model predictive control (DMPC) law is designed based on dual decomposition with sub-gradient method for each subgroup of variables update. As the convergence rate is slow ($\mathcal{O}(\frac{1}{\sqrt{k}})$), the corresponding early stop criterion is presented in order to reduce the dual decomposition iterations. The paper Farokhi et al. [2013] extended the results of Giselsson and Rantzer [2010] to the systems with nonlinear interconnected dynamics and cost functions. In Giselsson et al. [2013], the authors considered the distributed MPC setup with a separable quadratic cost matrix, and the coupling in the cost function is the same as the coupling in the dynamics, thus the dual problem can be solved in a parallel fashion with accelerated gradient method, which has a convergence rate of $\mathcal{O}(\frac{1}{k^2})$. The result of a bound on the number of dual variable iterations to guarantee certain accuracy of this problem was proposed in Giselsson [2013b]

⋆ This work has been partially supported by the RT-Group in Linköping University, Sweden, the National Natural Science Foundation of China (Grant No. 61304091) and the Research Foundation for the Doctoral Program of Higher Education of China (Grant No. 20120042120035).

by incorporating Hessian information to specify the step matrix in each iteration, which significantly reduced the needed iterations, and the early termination condition was derived in Giselsson and Rantzer [2013] to obtain some suboptimal performance. However, for the algorithm of dual decomposition to converge, there are a lot of assumptions, like the cost function needs strong convexity requirement and Slater's constraint qualification. It is also possible to achieve distributed solution using second order Newton type method, but the Hessian matrix requires to expose a block diagonal structure for each part of the decomposed subsystems. Some application of distributed Newton method can be found for network utility optimization problem Wei et al. [2013] and for cross-layer network optimization Liu et al. [2013], but this has not been studied for MPC.

The result of DMPC via ADMM algorithm was presented in Summers and Lygeros [2012] and Farokhi et al. [2013], and the difference between these two papers is that Summers and Lygeros [2012] integrates the variable constraints directly in these variable's update steps, resulting in some small inequality constrained QP problems in each iteration and using CVX to solve, which is in general expensive, while Farokhi et al. [2013] demonstrates the use of indicator function in augmented Lagrangian and utilize projected gradient descent method to solve the problem. The advantage of the method of multipliers over dual decomposition is that the cost function does not have to be strongly convex. Recent paper Hong and Luo [2013], derived the linear convergence result of ADMM for the sum of arbitrary number of generalized cost functions, which allows us to handle the constraints more efficiently and makes it possible to derive ADMM-type MPC algorithm for large-scale systems. This paper describes the ADMM-type update procedures of constrained real-time MPC problem for both centralized and distributed systems.

2. PROBLEM SETUP

Consider the linear system

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = \bar{x}, \tag{1}$$

$$y_k = Cx_k + Du_k + g_k, \tag{2}$$

$$z_k = Ex_k + Fu_k + h_k, \tag{3}$$

where $x_k \in \mathbf{R}^n$, $u_k \in \mathbf{R}^m$, $y_k \in \mathbf{R}^p$ and $z_k \in \mathbf{R}^q$ are the state, input, output and auxiliary vectors at the time instant k, respectively. Note that the matrices A, B, C, D, E and F can be time-varying matrices.

Define the vectorized variables $\mathbf{x} = [(x_0)^{\mathrm{T}}, \cdots, (x_{N-1})^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{u} = [(u_0)^{\mathrm{T}}, \cdots, (u_{N-1})^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{y} = [(y_0)^{\mathrm{T}}, \cdots, (y_{N-1})^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{z} = [(z_0)^{\mathrm{T}}, \cdots, (z_{N-1})^{\mathrm{T}}]^{\mathrm{T}}$.

Let

$$f(x_N, \mathbf{y}) = \|x_N\|_{2,Q}^2 + \sum_{k=0}^{N-1} \|y_k\|_2^2. \tag{4}$$

Consider the model predictive control (MPC) setup,

$$\min_{x_N,\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}} f(x_N, \mathbf{y}), \tag{5}$$
$$\text{s.t.} \quad (1), (2), (3) \text{ and } \mathbf{z} \geq 0.$$

3. ADMM FOR SEPARABLE CONVEX OPTIMIZATION

Consider the optimization problem with separable objective function and linear constraints

$$\min_{\nu_1,\cdots,\nu_K} \theta_1(\nu_1) + \theta_2(\nu_2) + \cdots + \theta_K(\nu_K), \tag{6}$$
$$\text{s.t.} \quad L_1\nu_1 + L_2\nu_2 + \cdots + L_K\nu_K = l.$$

where $\theta_i : \mathbf{R}^{n_\nu^i} \to \mathbf{R}$ $(i = 1, 2, \cdots, K)$ are convex functions but not necessary smooth, and $L_i \in \mathbf{R}^{l \times n_\nu^i}$, $l \in \mathbf{R}^l$.

One possible algorithm for solving the above separable optimization problem is ADMM algorithm. First, we define the augmented Lagrangian of the constrained optimization problem (6) as

$$L_\rho(\nu_1, \nu_2, \cdots, \nu_K; \nu_d) = \sum_{i=1}^K \theta_i(\nu_i) + \nu_d^{\mathrm{T}} \left(\sum_{i=1}^K L_i\nu_i - l\right) + \frac{\rho}{2} \left\|\sum_{i=1}^K L_i\nu_i - l\right\|_2^2, \tag{7}$$

where $\nu_d \in \mathbf{R}^l$ is the dual variable and $\rho > 0$ is the penalty parameter.

The scaled augmented Lagrangian is

$$L_\rho(\nu_1, \nu_2, \cdots, \nu_K; \nu_s) = \sum_{i=1}^K \theta_i(\nu_i) + \frac{\rho}{2} \left\|\sum_{i=1}^K L_i\nu_i - l + \nu_s\right\|_2^2 + \text{constant}, \tag{8}$$

where $\nu_s = \nu_d/\rho$ are the scaled dual variables.

In each iteration of ADMM algorithm, it consists "alternating direction" minimization and updating of the scaled dual variables. The primal variables are updated inexactly by one Gauss-Seidel sweep. The update steps are

$$\nu_i^{r+1} := \mathrm{argmin}_{\nu_i} L_\rho(\nu_1^{r+1}, \cdots, \nu_{k-1}^{r+1}, \nu_k, \nu_{k+1}^r, \cdots, \nu_K^r; \nu_s^r),$$
$$\nu_s^{r+1} := \nu_s^r + \left(\sum_{i=1}^K L_i\nu_i^{r+1} - l\right), \quad i = 1, 2, \cdots, K.$$

Note that under some assumptions of the objective functions θ_i's and the matrices L_i's in the coupling equality constraints, the updates of ν_i's in each step can be computed very efficiently, like if θ_i's are quadratic, the updates of ν_i's are boil down to solve linear equations. More examples can be found in Boyd et al. [2010] and the references therein.

In the next section, we will reformulate the MPC problem (5) in the form of (6).

4. SEPARABLE MPC FORMULATION

We can formulate the constraints in the MPC problem (5) into the following two constraint sets:

$$\mathcal{C}_1 = \{(x_N, \mathbf{x}, \mathbf{u}, \mathbf{y}, \mathbf{z}) | (1), (2), (3)\},$$

and

$$\mathcal{C}_2 = \{\mathbf{z} | \mathbf{z} \geq 0\}.$$

Then, the MPC problem (5) is equivalent to

$$\min_{x_N,\mathbf{x},\mathbf{u},\mathbf{y},\mathbf{z}} f(x_N,\mathbf{y}) + I_{\mathcal{C}_1}(x_N,\mathbf{x},\mathbf{u},\mathbf{y},\mathbf{z}) + I_{\mathcal{C}_2}(\mathbf{z}),$$

where $I_{\mathcal{C}}(\cdot)$ is the indicator function for the closed convex set $\mathcal{C}$ (i.e. $I_{\mathcal{C}}(\nu) = 0$ for $\nu \in \mathcal{C}$, and $I_{\mathcal{C}}(\nu) = \infty$ for $\nu \notin \mathcal{C}$). However, this is not a separable cost function, but by introducing extra variables $\mathbf{x}_c$, $x_{c,N}$, $\mathbf{u}_c$, $\mathbf{y}_c$ and $\mathbf{z}_c$ and extra so-called consensus constraints, we can write it equivalently as

$$\begin{aligned} \min_{x_N,x_{c,N},\mathbf{x},\mathbf{u},\mathbf{x}_c,\mathbf{u}_c,\mathbf{y},\mathbf{z},\mathbf{y}_c,\mathbf{z}_c} & f(x_N,\mathbf{y}) + I_{\mathcal{C}_1}(x_{c,N},\mathbf{x}_c,\mathbf{u}_c,\mathbf{y}_c,\mathbf{z}) \\ & + I_{\mathcal{C}_2}(\mathbf{z}_c), \qquad (9) \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{x}_c, \\ & x_N = x_{c,N}, \\ & \mathbf{u} = \mathbf{u}_c, \\ & \mathbf{y} = \mathbf{y}_c, \\ & \mathbf{z} = \mathbf{z}_c. \end{aligned}$$

The MPC formulation (9) can be adapted to the multi-block optimization problem (6) with $K = 2$ if we denote $\nu_1 := (x_N,\mathbf{x},\mathbf{u},\mathbf{y},\mathbf{z}_c)$, $\nu_2 := (x_{c,N},\mathbf{x}_c,\mathbf{u}_c,\mathbf{y}_c,\mathbf{z})$, $\theta_1(\nu_1) := f(x_N,\mathbf{y}) + I_{\mathcal{C}_2}(\mathbf{z}_c)$ and $\theta_2(\nu_2) := I_{\mathcal{C}_1}(x_{c,N},\mathbf{x},\mathbf{u},\mathbf{x}_c,\mathbf{u}_c,\mathbf{y}_c,\mathbf{z})$ with $L_1 = -L_2 = I$ and $l = 0$.

5. ADMM-TYPE MPC ALGORITHM

The scaled augmented Lagrangian in (8) for the optimization problem (9) is

$$\begin{aligned} L_\rho = & f(x_N,\mathbf{y}) + I_{\mathcal{C}_1}(x_{c,N},\mathbf{x}_c,\mathbf{u}_c,\mathbf{y}_c,\mathbf{z}) + I_{\mathcal{C}_2}(\mathbf{z}_c) \\ & + \frac{\rho}{2}(\|\mathbf{x} - \mathbf{x}_c + \mathbf{x}_s\|_2^2 + \|x_N - x_{c,N} + x_{s,N}\|_2^2 \\ & + \|\mathbf{u} - \mathbf{u}_c + \mathbf{u}_s\|_2^2 + \|\mathbf{y} - \mathbf{y}_c + \mathbf{y}_s\|_2^2 \\ & + \|\mathbf{z} - \mathbf{z}_c + \mathbf{z}_s\|_2^2) + \text{constant}, \end{aligned}$$

where $(\mathbf{x}_s, x_{s,N}, \mathbf{u}_s, \mathbf{y}_s, \mathbf{z}_s)$ are the scaled dual variables.

The ADMM algorithm for the MPC setup (9) consists of three steps.

5.1 Step 1

$$\begin{aligned} & (\mathbf{x}^{r+1}, x_N^{r+1}, \mathbf{u}^{r+1}, \mathbf{y}^{r+1}, \mathbf{z}_c^{r+1}) \\ & = \operatorname{argmin}_{\mathbf{x},x_N,\mathbf{u},\mathbf{y},\mathbf{z}_c} L_\rho(\mathbf{x}, x_N, \mathbf{u}, \mathbf{y}, \mathbf{z}^r, \mathbf{x}_c^r, x_{c,N}^r, \mathbf{u}_c^r, \mathbf{y}_c^r, \mathbf{z}_c) \\ & = \operatorname{argmin}_{\mathbf{x},x_N,\mathbf{u},\mathbf{y},\mathbf{z}_c} [f(x_N,\mathbf{y}) + I_{\mathcal{C}_2}(\mathbf{z}_c) + \frac{\rho}{2}(\|\mathbf{x} - \mathbf{x}_c^r + \mathbf{x}_s^r\|_2^2 \\ & \quad + \|x_N - x_{c,N}^r + x_{s,N}^r\|_2^2 + \|\mathbf{u} - \mathbf{u}_c^r + \mathbf{u}_s^r\|_2^2 \\ & \quad + \|\mathbf{y} - \mathbf{y}_c^r + \mathbf{y}_s^r\|_2^2 + \|\mathbf{z}^r - \mathbf{z}_c + \mathbf{z}_s^r\|_2^2)] \end{aligned}$$

The solutions are

$$\begin{aligned} x_k^{r+1} &= x_{c,k}^r - x_{s,k}^r, \quad k = 0, 1, \cdots, N-1, \\ x_N^{r+1} &= (2Q + \rho I_{n_x})^{-1} \rho (x_{c,N}^r - x_{s,N}^r), \\ u_k^{r+1} &= u_{c,k}^r - u_{s,k}^r, \quad k = 0, 1, \cdots, N-1, \\ y_k^{r+1} &= (2 + \rho)^{-1} \rho (y_{c,k}^r - y_{s,k}^r), \quad k = 0, 1, \cdots, N-1, \\ z_{c,k}^{r+1} &= \Pi_{\mathcal{C}_2}(z_k^r + z_{s,k}^r) \Leftrightarrow z_{c,k}^{r+1} = \min\{z_k^r + z_{s,k}^r, 0\}, \\ & \quad k = 0, 1, \cdots, N-1. \end{aligned}$$

5.2 Step 2

$$\begin{aligned} & (x_{c,N}^{r+1}, \mathbf{y}_c^{r+1}, \mathbf{z}^{r+1}, \mathbf{x}_c^{r+1}, \mathbf{u}_c^{r+1}) \\ & = \operatorname{argmin}_{x_{c,N},\mathbf{y}_c,\mathbf{z},\mathbf{x}_c,\mathbf{u}_c} L_\rho(\mathbf{x}_c, \mathbf{x}^{r+1}, x_N^{r+1}, \mathbf{u}_c, \mathbf{u}^{r+1}, \mathbf{y}^{r+1}, \mathbf{z}, x_{c,N}, \mathbf{y}_c, \mathbf{z}_c^{r+1}) \\ & = \operatorname{argmin}_{x_{c,N},\mathbf{y}_c,\mathbf{z},\mathbf{x}_c,\mathbf{u}_c} I_{\mathcal{C}_1}(x_{c,N}, \mathbf{x}_c, \mathbf{u}_c, \mathbf{y}_c, \mathbf{z}) + \frac{\rho}{2}(\|\mathbf{x}^{r+1} - \mathbf{x}_c \\ & \quad + \mathbf{x}_s^r\|_2^2 + \|x_N^{r+1} - x_{c,N} + x_{s,N}^r\|_2^2 + \|\mathbf{u}^{r+1} - \mathbf{u}_c + \mathbf{u}_s^r\|_2^2 \\ & \quad + \|\mathbf{y}^{r+1} - \mathbf{y}_c + \mathbf{y}_s^r\|_2^2 + \|\mathbf{z} - \mathbf{z}_c^{r+1} + \mathbf{z}_s^r\|_2^2) \end{aligned} \qquad (10)$$

Denote

$$\begin{aligned} (\mathbf{x}_p^r, x_{p,N}^r, \mathbf{u}_p^r, \mathbf{y}_p^r, \mathbf{z}_p^r) = (& \mathbf{x}^{r+1} + \mathbf{x}_s^r, x_N^{r+1} + x_{s,N}^r, \mathbf{u}^{r+1} + \mathbf{u}_s^r, \\ & \mathbf{y}^{r+1} + \mathbf{y}_s^r, \mathbf{z}_c^{r+1} - \mathbf{z}_s^r), \end{aligned}$$

the optimization problem (10) becomes an Euclidean projection onto the set $\mathcal{C}_1$, i.e.

$$(\mathbf{x}_c^{r+1}, x_{c,N}^{r+1}, \mathbf{u}_c^{r+1}, \mathbf{y}_c^{r+1}, \mathbf{z}^{r+1}) = \Pi_{\mathcal{C}_1}(\mathbf{x}_p^r, x_{p,N}^r, \mathbf{u}_p^r, \mathbf{y}_p^r, \mathbf{z}_p^r), \qquad (11)$$

which boils down to the following optimization problem

$$\min_{x_{c,N},\mathbf{y}_c,\mathbf{z},\mathbf{x}_c,\mathbf{u}_c} \|(\mathbf{x}_c^{r+1}, x_{c,N}^{r+1}, \mathbf{u}_c^{r+1}, \mathbf{y}_c^{r+1}, \mathbf{z}^{r+1}) - (\mathbf{x}_p^r, x_{p,N}^r, \mathbf{u}_p^r, \mathbf{y}_p^r, \mathbf{z}_p^r)\|_2^2, \qquad (12)$$

$$\begin{aligned} \text{s.t.} \quad & x_{c,k+1} = Ax_{c,k} + Bu_{c,k}, \quad k = 0, \cdots, N-1, \\ & y_{c,k} = Cx_{c,k} + Du_{c,k}, + g_k, \quad k = 0, \cdots, N-1, \\ & z_k = Ex_{c,k} + Fu_{c,k} + h_k, \quad k = 0, \cdots, N-1, \end{aligned}$$

which can be solved efficiently using Riccati recursion Rao et al. [1998].

5.3 Step 3

Update the scaled dual variables

$$\begin{aligned} & (\mathbf{x}_s^{r+1}, x_{s,N}^{r+1}, \mathbf{u}_s^{r+1}, \mathbf{y}_s^{r+1}, \mathbf{z}_s^{r+1}) = (\mathbf{x}_s^r, x_{s,N}^r, \mathbf{u}_s^r, \mathbf{y}_s^r, \mathbf{z}_s^r) \\ & + ((\mathbf{x}^{r+1}, x_N^{r+1}, \mathbf{u}^{r+1}, \mathbf{y}^{r+1}, \mathbf{z}^{r+1}) - (\mathbf{x}_c^{r+1}, x_{c,N}^{r+1}, \mathbf{u}_c^{r+1}, \\ & \mathbf{y}_c^{r+1}, \mathbf{z}_c^{r+1})). \end{aligned}$$

Remark 1. The MPC formulation of (9) satisfies all the Assumptions (a)-(g) in Hong and Luo [2013], we can conclude that if we choose the step size ρ small enough, the sequence of the above iterates of the primal and dual variables and the sequence of function values $f(x_N^r, \mathbf{y}^r)$ converge R-linearly (see Theorem 3.1 and Corollary 3.1 in Hong and Luo [2013]).

6. DISTRIBUTED MPC VIA ADMM

For the linear systems (1)-(3). Introduce a non-overlapping partition of M subsystems $\mathcal{S}_i$, $i = 1, \cdots, M$. Let $x_k^i \in \mathbf{R}^{n_i}$ denote the state of subsystem $\mathcal{S}_i$, i.e.

$$x_k = [(x_k^1)^{\mathrm{T}}, (x_k^2)^{\mathrm{T}}, \cdots, (x_k^M)^{\mathrm{T}}]^{\mathrm{T}},$$

with $\sum_{i=1}^M n_i = n$. Each subsystem $\mathcal{S}_i$ has its own input vector $u_k^i \in \mathbf{R}^{m_i}$ and output vector $y_k^i \in \mathbf{R}^{p_i}$,

$$u_k = [(u_k^1)^{\mathrm{T}}, (u_k^2)^{\mathrm{T}}, \cdots, (u_k^M)^{\mathrm{T}}]^{\mathrm{T}},$$
$$y_k = [(y_k^1)^{\mathrm{T}}, (y_k^2)^{\mathrm{T}}, \cdots, (y_k^M)^{\mathrm{T}}]^{\mathrm{T}},$$

with $\sum_{i=1}^M m_i = m$ and $\sum_{i=1}^M p_i = p$. Thus, we can write

$$\begin{bmatrix} x_{k+1}^1 \\ \vdots \\ x_{k+1}^M \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{bmatrix} \begin{bmatrix} x_k^1 \\ \vdots \\ x_k^M \end{bmatrix} + \begin{bmatrix} B_{11} & \cdots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \cdots & B_{MM} \end{bmatrix} \begin{bmatrix} u_k^1 \\ \vdots \\ u_k^M \end{bmatrix},$$

$$\begin{bmatrix} y_k^1 \\ \vdots \\ y_k^M \end{bmatrix} = \begin{bmatrix} C_{11} & \cdots & C_{1M} \\ \vdots & \ddots & \vdots \\ C_{M1} & \cdots & C_{MM} \end{bmatrix} \begin{bmatrix} x_k^1 \\ \vdots \\ x_k^M \end{bmatrix} + \begin{bmatrix} D_{11} & \cdots & D_{1M} \\ \vdots & \ddots & \vdots \\ D_{M1} & \cdots & D_{MM} \end{bmatrix} \begin{bmatrix} u_k^1 \\ \vdots \\ u_k^M \end{bmatrix} + \begin{bmatrix} g_k^1 \\ \vdots \\ g_k^M \end{bmatrix}$$

The dynamics for subsystem $\mathcal{S}_i$ can be expressed as

$$\begin{aligned} x_{k+1}^i &= A_{ii}x_k^i + B_{ii}u_k^i + \sum_{j\in\mathcal{N}_i}(A_{ij}x_k^j + B_{ij}u_k^j), \quad x_0^i = \bar{x}_i, \\ y_k^i &= C_{ii}x_k^i + D_{ii}u_k^i + \sum_{j\in\mathcal{N}_i}(C_{ij}x_k^j + D_{ij}u_k^j) + g_k^i. \end{aligned}$$

Let $L_{ij} = \begin{bmatrix} A_{ij} & B_{ij} \\ C_{ij} & D_{ij} \end{bmatrix}$ denote the interconnection matrices, and let $l_{ij} = \|L_{ij}\|$ denote the interconnection gain. The set of neighbors of subsystem $\mathcal{S}_i$, which have direct influence on subsystem $\mathcal{S}_i$, is defined by the set

$$\mathcal{N}_i = \{j \in \{1, \cdots, M\} | L_{ij} \neq 0\}, \quad i = 1, \cdots, M.$$

Consider the distributed model predictive control (MPC) setup,

$$\min \sum_{i=1}^{M}\left(\sum_{k=0}^{N-1}\|y_k^i\|_2^2 + \|x_N^i\|_{2,Q_i}^2\right), \tag{13}$$

$$\text{s.t.} \quad x_{k+1}^i = A_{ii}x_k^i + B_{ii}u_k^i + \sum_{j\in\mathcal{N}_i}(A_{ij}x_k^j + B_{ij}u_k^j), \tag{13.1}$$

$$y_k^i = C_{ii}x_k^i + D_{ii}u_k^i + \sum_{j\in\mathcal{N}_i}(C_{ij}x_k^j + D_{ij}u_k^j) + g_k^i, \tag{13.2}$$

$$z_k^i = E_i x_k^i + F_i u_k^i + h_k^i, \tag{13.3}$$

$$z_k^i \geq 0. \tag{13.4}$$

6.1 Separable MPC Formulation for Distributed Systems

For the distributed MPC problem (13), the vector variables are $\mathbf{x}^i = [(x_0^i)^{\mathrm{T}}, \cdots, (x_{N-1}^i)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{u}^i = [(u_0^i)^{\mathrm{T}}, \cdots, (u_{N-1}^i)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{y}^i = [(y_0^i)^{\mathrm{T}}, \cdots, (y_{N-1}^i)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{z}^i = [(z_0^i)^{\mathrm{T}}, \cdots, (z_{N-1}^i)^{\mathrm{T}}]^{\mathrm{T}}$ for $i = 1, \cdots, M$ and $\mathbf{x} = [(\mathbf{x}^1)^{\mathrm{T}}, \cdots, (\mathbf{x}^M)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{u} = [(\mathbf{u}^1)^{\mathrm{T}}, \cdots, (\mathbf{u}^M)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{y} = [(\mathbf{y}^1)^{\mathrm{T}}, \cdots, (\mathbf{y}^M)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{z} = [(\mathbf{z}^1)^{\mathrm{T}}, \cdots, (\mathbf{z}^M)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{x}_N = [(x_N^1)^{\mathrm{T}}, \cdots, (x_N^M)^{\mathrm{T}}]^{\mathrm{T}}$.

The objective function is

$$f(\mathbf{x}_N, \mathbf{y}) = \sum_{i=1}^{M} f_i(x_N^i, \mathbf{y}^i) = \sum_{i=1}^{M}\left(\sum_{k=0}^{N-1}\|y_k^i\|_2^2 + \|x_N^i\|_{2,Q_i}^2\right).$$

The constraints in (13) can be formulated into the following two constraint sets:

$$\mathcal{C}_1^i = \{(\mathbf{x}^i, x_N^i, \mathbf{u}^i, \mathbf{y}^i, \mathbf{z}^i) | (13.1), (13.2), (13.3)\},$$

and

$$\mathcal{C}_2^i = \{\mathbf{z}^i | (13.4)\}.$$

Then, we can write (13) as

$$\min_{x_N^i, \mathbf{x}^i, \mathbf{u}^i, \mathbf{y}^i, \mathbf{z}^i} \sum_{i=1}^{M}\left(f_i(x_N^i, \mathbf{y}^i) + I_{\mathcal{C}_1^i}(x_N^i, \mathbf{x}^i, \mathbf{u}^i, \mathbf{z}^i) + I_{\mathcal{C}_2^i}(\mathbf{z}^i)\right).$$

By introducing extra variables $\mathbf{x}_c^i$, $x_{c,N}^i$, $\mathbf{u}_c^i$, $\mathbf{y}_c^i$ and $\mathbf{z}_c^i$ and extra consensus constraints, we can separate the objective functions and write it equivalently as

$$\begin{aligned} \min_{x_N^i, x_{c,N}^i, \mathbf{x}^i, \mathbf{x}_c^i, \mathbf{u}^i, \mathbf{u}_c^i, \mathbf{y}^i, \mathbf{y}_c^i, \mathbf{z}^i, \mathbf{z}_c^i} & \sum_{i=1}^{M}\Big(f_i(x_N^i, \mathbf{y}^i) \\ & + I_{\mathcal{C}_1^i}(\mathbf{x}_c^i, x_{c,N}^i, \mathbf{u}_c^i, \mathbf{y}_c^i, \mathbf{z}^i) + I_{\mathcal{C}_2^i}(\mathbf{z}_c^i)\Big), \quad (14) \\ \text{s.t.} \quad & \mathbf{x}^i = \mathbf{x}_c^i, \\ & x_N^i = x_{c,N}^i, \\ & \mathbf{u}^i = \mathbf{u}_c^i, \\ & \mathbf{y}^i = \mathbf{y}_c^i, \\ & \mathbf{z}^i = \mathbf{z}_c^i. \end{aligned}$$

The scaled augmented Lagrangian is

$$\begin{aligned} L_\rho = \sum_{i=1}^{M} \Big(& f_i(x_N^i, \mathbf{y}^i) + I_{\mathcal{C}_1^i}(\mathbf{x}_c^i, x_{c,N}^i, \mathbf{u}_c^i, \mathbf{y}_c^i, \mathbf{z}^i) \\ & + I_{\mathcal{C}_2^i}(\mathbf{z}_c^i) + \frac{\rho}{2}(\|\mathbf{x}^i - \mathbf{x}_c^i + \mathbf{x}_s^i\|_2^2 + \|x_N^i - x_{c,N}^i + x_{s,N}^i\|_2^2 \\ & + \|\mathbf{u}^i - \mathbf{u}_c^i + \mathbf{u}_s^i\|_2^2 + \|\mathbf{y}^i - \mathbf{y}_c^i + \mathbf{y}_s^i\|_2^2 \\ & + \|\mathbf{z}^i - \mathbf{z}_c^i + \mathbf{z}_s^i\|_2^2)\Big) + \text{constant}, \end{aligned}$$

where $(\mathbf{x}_s^i, x_{s,N}^i, \mathbf{u}_s^i, \mathbf{y}_s^i, \mathbf{z}_s^i)$ are the scaled dual variables.

6.2 ADMM-type MPC Algorithm for Distributed Systems

Next, we will present the coordinate search iterations of ADMM with respect to the local variables of each subpartition i of the system block and their corresponding constraint sets $\mathcal{C}_1^i$ and $\mathcal{C}_2^i$.

Step 1

$$\begin{aligned} (\mathbf{x}^{r+1}, & x_N^{r+1}, \mathbf{u}^{r+1}, \mathbf{y}^{r+1}, \mathbf{z}_c^{r+1}) \\ & = \operatorname{argmin}_{\mathbf{x}, x_N, \mathbf{u}, \mathbf{y}, \mathbf{z}_c} L_\rho(\mathbf{x}, x_N, \mathbf{u}, \mathbf{y}, \mathbf{z}^r, \mathbf{x}_c^r, x_{c,N}^r, \mathbf{u}_c^r, \mathbf{y}_c^r, \mathbf{z}_c), \\ & = \operatorname{argmin}_{\mathbf{x}, x_N, \mathbf{u}, \mathbf{y}, \mathbf{z}_c} \Big(f(x_N, \mathbf{y}) + I_{\mathcal{C}_2}(\mathbf{z}_c) + \frac{\rho}{2}(\|\mathbf{x} - \mathbf{x}_c^r + \mathbf{x}_s^r\|_2^2 \\ & \quad + \|x_N - x_{c,N}^r + x_{s,N}^r\|_2^2 + \|\mathbf{u} - \mathbf{u}_c^r + \mathbf{u}_s^r\|_2^2 + \|\mathbf{y} - \mathbf{y}_c^r + \mathbf{y}_s^r\|_2^2 \\ & \quad + \|\mathbf{z}^r - \mathbf{z}_c + \mathbf{z}_s^r\|_2^2)\Big), \end{aligned}$$

The solutions are

$$\begin{aligned} x_k^{i,r+1} &= x_{c,k}^{i,r} - x_{s,k}^{i,r}, \quad k = 0, 1, \cdots, N-1, \\ x_N^{i,r+1} &= (2Q_i + \rho I_{n_x^i})^{-1}\rho(x_{c,N}^r - x_{s,N}^r), \\ u_k^{i,r+1} &= u_{c,k}^{i,r} - u_{s,k}^{i,r}, \quad k = 0, 1, \cdots, N-1, \\ y_k^{i,r+1} &= (2+\rho)^{-1}\rho(y_{c,k}^{i,r} - y_{s,k}^{i,r}), \quad k = 0, 1, \cdots, N-1, \\ z_{c,k}^{i,r+1} &= \Pi_{\mathcal{C}_2^i}(z_k^{i,r} + z_{s,k}^{i,r}) \Leftrightarrow z_{c,k}^{i,r+1} = \max\{z_k^{i,r} + z_{s,k}^{i,r}, 0\}, \\ & k = 0, 1, \cdots, N-1. \end{aligned}$$

Step 2

$$\begin{aligned} & (x_{c,N}^{i,r+1}, \mathbf{y}_c^{i,r+1}, \mathbf{z}^{i,r+1}, \mathbf{x}_c^{i,r+1}, \mathbf{u}_c^{i,r+1}) \\ & = \operatorname{argmin}_{x_{c,N}^i \mathbf{y}_c^i \mathbf{z}^i \mathbf{x}_c^i \mathbf{u}_c^i} L_\rho(\mathbf{x}_c^i, \mathbf{x}^{i,r+1}, x_N^{i,r+1}, \mathbf{u}_c^i, \mathbf{u}^{i,r+1}, \mathbf{y}^{i,r+1}, \\ & \qquad \mathbf{z}^i, \mathbf{x}_c^i, \mathbf{y}_c^i, \mathbf{z}_c^{i,r+1}), \\ & = \operatorname{argmin}_{x_{c,N}^i, \mathbf{y}_c^i, \mathbf{z}^i, \mathbf{x}_c^i, \mathbf{u}_c^i} I_{\mathcal{C}_1^i}(\mathbf{x}_c^i, x_{c,N}^i, \mathbf{u}_c^i, \mathbf{y}_c^i, \mathbf{z}^i) \\ & + \frac{\rho}{2}(\|x_N^{i,r+1} - x_{c,N}^i + x_{s,N}^{i,r}\|_2^2 + \|\mathbf{x}^{i,r+1} - \mathbf{x}_c^i + \mathbf{x}_s^{i,r}\|_2^2 + \|\mathbf{u}^i - \mathbf{u}_c^{i,r+1} + \mathbf{u}_s^{i,r}\|_2^2 \\ & + \|\mathbf{y}^{i,r+1} - \mathbf{y}_c^i + \mathbf{y}_s^{i,r}\|_2^2 + \|\mathbf{z}^i - \mathbf{z}_c^{i,r+1} + \mathbf{z}_s^{i,r}\|_2^2). \end{aligned} \tag{15}$$

Denote

$$(\mathbf{x}_p^{i,r}, x_{p,N}^{i,r}, \mathbf{u}_p^{i,r}, \mathbf{y}_p^{i,r}, \mathbf{z}_p^{i,r}) = (\mathbf{x}^{i,r+1} + \mathbf{x}_s^{i,r}, x_N^{i,r+1} + x_{s,N}^{i,r}, \mathbf{u}^{i,r+1} + \mathbf{u}_s^{i,r}, \mathbf{y}^{i,r+1} + \mathbf{y}_s^{i,r}, \mathbf{z}_c^{i,r+1} - \mathbf{z}_s^{i,r}),$$

the optimization problem (15) becomes an Euclidean projection onto the set $\mathcal{C}_1^i$, i.e.

$$(\mathbf{x}_c^{i,r+1}, x_{c,N}^{i,r+1}, \mathbf{u}_c^{i,r+1}, \mathbf{y}_c^{i,r+1}, \mathbf{z}^{i,r+1}) = \Pi_{\mathcal{C}_1^i}(\mathbf{x}_p^{i,r}, x_{p,N}^{i,r}, \mathbf{u}_p^{i,r}, \mathbf{y}_p^{i,r}, \mathbf{z}_p^{i,r}),$$

which boils down to the following optimization problem

$$\min_{x_{c,N}^i, \mathbf{y}_c^i, \mathbf{z}^i, \mathbf{x}_c^i, \mathbf{u}_c^i} \|(\mathbf{x}_c^{i,r+1}, x_{c,N}^{i,r+1}, \mathbf{u}_c^{i,r+1}, \mathbf{y}_c^{i,r+1}, \mathbf{z}^{i,r+1}) - (\mathbf{x}_p^{i,r}, x_{p,N}^{i,r}, \mathbf{u}_p^{i,r}, \mathbf{y}_p^{i,r}, \mathbf{z}_p^{i,r})\|_2^2, \quad (16)$$

$$\text{s.t.} \quad x_{c,k+1}^{i,r+1} = A_{ii} x_{c,k}^{i,r+1} + B_{ii} u_{c,k}^{i,r+1} + \sum_{j<i, j\in\mathcal{N}_i} (A_{ij} x_{c,k}^{j,r+1} + B_{ij} u_{c,k}^{j,r+1}), + \sum_{j>i, j\in\mathcal{N}_i} (A_{ij} x_{c,k}^{j,r} + B_{ij} u_{c,k}^{j,r}), \quad k = 0, \cdots, N-1,$$

$$y_{c,k}^{i,r+1} = C_{ii} x_{c,k}^{i,r+1} + \sum_{j<i, j\in\mathcal{N}_i} C_{ij} x_{c,k}^{j,r+1} + \sum_{j>i, j\in\mathcal{N}_i} C_{ij} x_{c,k}^{j,r} + D_{ii} u_{c,k}^{i,r+1} + \sum_{j<i, j\in\mathcal{N}_i} D_{ij} u_{c,k}^{j,r+1} + \sum_{j>i, j\in\mathcal{N}_i} D_{ij} u_{c,k}^{j,r} + g_k^i, \quad k = 0, \cdots, N-1,$$

$$z_k^{i,r+1} = E_i x_{c,k}^{i,r+1} + F_i u_{c,k}^{i,r+1} + h_k^i, \quad k = 0, \cdots, N-1,$$

which can be solved by Riccati recursion Rao et al. [1998].

Step 3 Update the scaled dual variables

$$(\mathbf{x}_s^{r+1}, x_{s,N}^{r+1}, \mathbf{u}_s^{r+1}, \mathbf{y}_s^{r+1}, \mathbf{z}_s^{r+1}) = (\mathbf{x}_s^r, x_{s,N}^r, \mathbf{u}_s^r, \mathbf{y}_s^r, \mathbf{z}_s^r) + ((\mathbf{x}^{r+1}, x_N^{r+1}, \mathbf{u}^{r+1}, \mathbf{y}^{r+1}, \mathbf{z}^{r+1}) - (\mathbf{y}_c^{r+1}, x_{c,N}^{r+1}, \mathbf{y}_c^{r+1}, \mathbf{y}_c^{r+1}, \mathbf{z}_c^{r+1})).$$

7. ON CONVERGENCE RATE

Theorem 1. The sequence of the above ADMM iterates $\{(\mathbf{x}^r, x_N^r, \mathbf{u}^r, \mathbf{y}^r, \mathbf{z}^r, \mathbf{x}_s^r, x_{s,N}^r, \mathbf{u}_s^r, \mathbf{y}_s^r, \mathbf{z}_s^r)\}$ for solving the distributed MPC problem (14) converges R-linearly to an optimal solution, provided the step size ρ is sufficiently small. Moreover, the sequence of function values $f(\mathbf{x}_N^r, \mathbf{y}^r)$ also converges R-linearly.

Proof. The convex problem (14) satisfies:

- The primal and dual optimum can be attained and unique (Strong duality holds).
- $\theta_i(\nu_i)$'s in (6) correspond to either $f_i(x_N^i, \mathbf{y}^i) + I_{\mathcal{C}_2^i}(\mathbf{z}_c^i)$ or $I_{\mathcal{C}_1^i}(\mathbf{x}_c^i, x_{c,N}^i, \mathbf{u}_c^i, \mathbf{y}_c^i, \mathbf{z}^i)$ in (14), where f_i's are strictly convex and continuously differentiable.
- $\mathcal{C}_1^i$ and $\mathcal{C}_2^i$ are polyhedral sets.
- Each sub matrix L_i's in (6) equal I or $-I$ in (14).

Then the proof follows from Theorem 3.1 and Corollary 3.1 in Hong and Luo [2013]. □

8. ILLUSTRATIVE EXAMPLE

Let's consider a DMPC problem that is similar to Giselsson and Rantzer [2013] which is randomly generated. The dynamical system consists three subsystems and each subsystem has five states and one input. As the problem is a regulation problem, the output of the ith subsystem

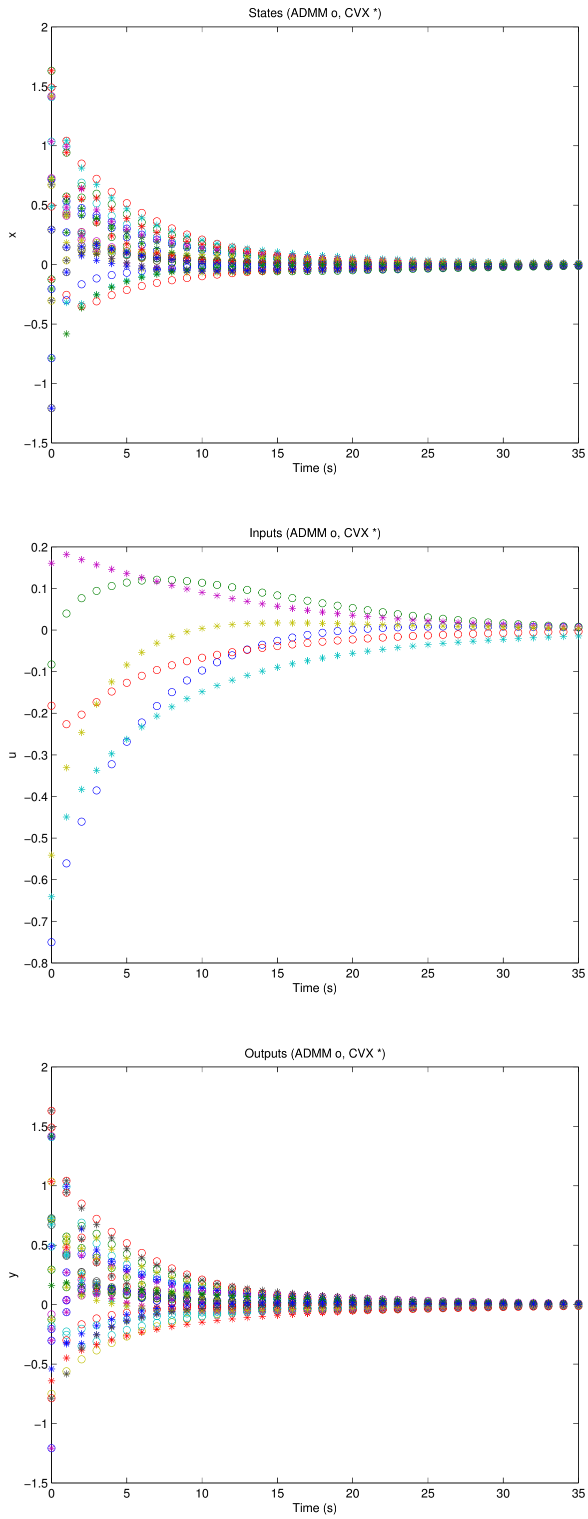

Fig. 1. Trajectories evolution from initial conditions.

is chosen to be $y_k^i = \begin{pmatrix} I_5 \\ 0 \end{pmatrix} x_k^i + \begin{pmatrix} 0 \\ I_1 \end{pmatrix} u_k^i$. Both state and input constraints are set within the range $[-2, 2]$. The control horizon is chosen to $N = 5$. All the computations and simulations are performed in MATLAB and CVX. In Fig. 1, the performance of the Distributed ADMM MPC algorithm described in Section 6 is compared with the centralized MPC result from CVX. From the simulations we find that the states evolution of the distributed ADMM MPC developed in this paper, which are marked as circles, can work as well as the centralized MPC computed with CVX, which are marked as stars. Fig. 2 shows the linear convergence of the objective function in the iterates.

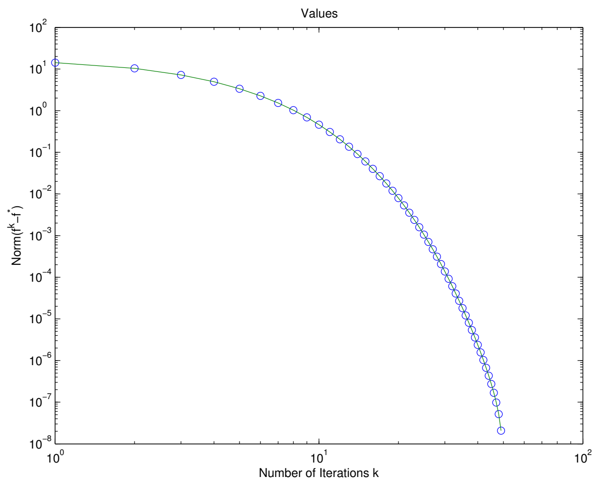

Fig. 2. Norm$(f^k - f^*)$ versus iteration for 1st sampl. time.

9. CONCLUSIONS

This paper derived an ADMM-type algorithm for solving large-scale MPC problem. The update of the state variables is in sequential and the estimate error can also be caused by communication delays and dropouts, which indicate the potential to develop more efficient communication schemes for distributed ADMM algorithms in the future.

REFERENCES

M. Annergren, A. Hansson and B. Wahlberg. An ADMM algorithm for solving l_1 regularized MPC. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)* , Maui, USA, 2012.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Second Edition, 1999.

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

E. Camponogara, D. Jia, B.H. Krogh and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, vol. 22, issue. 1, pp. 44–52, 2002.

P. D. Christofides, R. Scattolini, D. Muñoz de la Peña and J. Liu. Distributed model predictive control: a tutorial review and future research directions. *Computers and Chemical Engineering*, no. 51, pp. 21–41, 2013.

F. Farokhi, I. Shames and K. H. Johansson. Distributed MPC via dual decomposition and alternating direction method of multipliers. In J. M. Maestre and R. R. Negenborn, Eds., *Distributed MPC Made Easy*, Springer-Verlag, 2012.

P. Giselsson and A. Rantzer. Distributed model predictive control with sub optimality and stability guarantees. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)* , Atlanta, USA, 2010.

P. Giselsson. Execution time certification for gradient-based optimization in model predictive control. In *Proceedings of the IEEE Conference on Decision and Control (CDC)* , Maui, USA, 2012.

P. Giselsson. Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control. In *Proceedings of the 2013 American Control Conference*, Washington, D. C., USA, 2013.

P. Giselsson. A generalized distributed accelerated gradient method for DMPC with iteration complexity bounds. In *Proceedings of the 2013 American Control Conference*, Washington, D. C., USA, 2013.

P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, accepted for publication, 2013.

M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint* , March, 2013.

J. Liu, C. H. Xia, N. B. Shroff and H. D. Sherali. Distributed cross-layer optimization in wireless networks: A second-order approach. *arXiv preprint* , Feb, 2013.

E. Wei, A. Ozdaglar and A. Jadbabaie. A distributed newton method for network utility maximization– I: Algorithm. *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

C. V. Rao, S. J. Wright and J. B. Rawlings. Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 723–757, 1998.

R. Scattolini. Architectures for distributed and hierarchical model predictive control – a review. *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

S. Richter, C. N. Jones and M. Morari. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2012.

B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, vol. 59, issue 6, pp. 460–469, 2010.

T.H. Summers and J. Lygeros. Distributed model predictive consensus via the alternating direction method of multipliers In *Proceedings of the 50th Allerton Conference on Communication, Computing, and Control*, Monticello, IL, USA, October 1-5, 2012.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Submitted to *SIAM J. Optim.*, 2008.

Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.