Delft University of Technology

Delft Center for Systems and Control

Technical report 10-013

An improved distributed version of Han's method for distributed MPC of canal systems*

M.D. Doan, T. Keviczky, and B. De Schutter

If you want to cite this report, please use the following reference instead:
M.D. Doan, T. Keviczky, and B. De Schutter, "An improved distributed version of Han's method for distributed MPC of canal systems," *Proceedings of the 12th IFAC Symposium on Large Scale Systems: Theory and Applications*, Villeneuve d'Ascq, France, July 2010. doi:10.3182/20100712-3-FR-2020.00026

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/10_013.html

An improved distributed version of Han's method for distributed MPC of canal systems

Minh Dang Doan * **Tamás Keviczky** * **Bart De Schutter** *

* *Delft University of Technology, Delft, The Netherlands*
(e-mail: {m.d.doan,t.keviczky,b.deschutter}@tudelft.nl)

Abstract: Recently, we have introduced a distributed version of Han's method that can be used for distributed model predictive control (DMPC) of dynamically coupled linear systems, under coupling constraints (Doan et al., 2009). Some DMPC problems of water networks can be cast into this type. In this paper, we propose an improved version of this method and apply it to a canal system. The simulation results show that the modifications lead to faster convergence of the method, thus making it more practical in control of water networks.

Keywords: distributed optimization, model predictive control, water networks, dual decomposition, decentralized and cooperative control

1. INTRODUCTION

Optimization techniques have played a fundamental role in designing automatic control systems for most part of the past half century. This dependence is even more obvious in today's wide-spread use of online optimization-based control methods, such as Model Predictive Control (MPC) (Maciejowski, 2002; Rawlings and Mayne, 2009). The ability to express important process constraints and characterize comprehensive economic objective functions has made MPC the industry standard for controlling large-scale systems ranging from chemical processes to basic infrastructure.

For control of large-scale networked systems, *centralized* MPC may be considered impractical, inflexible, and unsuitable due to information exchange requirements and computational aspects. The subsystems in the network may belong to different authorities that prevent sending all necessary information to one processing center. Moreover, the optimization problem yielded by centralized MPC can be excessively large for real-time computation. In order to deal with these limitations, *distributed model predictive control* (DMPC) has been proposed for control of such large-scale systems, by decomposing the overall system into small subsystems (Jia and Krogh, 2001; Camponogara et al., 2002; Rawlings and Stewart, 2008). The subsystems then employ distinct MPC controllers that only solve local optimization problems, use local information from neighboring subsystems, and collaborate to achieve globally attractive solutions.

Approaches to DMPC design differ from each other in the problem setup. For systems with decoupled dynamics, Dunbar and Murray (2006) proposed a DMPC scheme focusing on multiple vehicles with coupled cost functions, and utilizing predicted trajectories of the neighbors in each subsystem's optimization. A DMPC scheme with a sufficient stability test for dynamically decoupled systems was proposed by Keviczky et al. (2006), in which each subsystem optimizes also over the behaviors of its neighbors. Richards and How (2007) proposed a robust DMPC method for decoupled systems with coupled constraints, based on constraint tightening and a serial solution approach. For systems with coupled dynamics and decoupled constraints Venkat et al. (2008) proposed a distributed MPC scheme, based on a Jacobi algorithm that deals with the primal problem, using a convex combination of new and old solutions. Other research related to the DMPC field is reported by Jia and Krogh (2002); Du et al. (2001); Li et al. (2005); Camponogara and Talukdar (2007); Mercangoz and Doyle III (2007); Alessio and Bemporad (2007, 2008); Necoara et al. (2008). A recent survey on DMPC can be found in (Scattolini, 2009).

Recently, we have developed a distributed version of Han's parallel method for convex optimization (Doan et al., 2009). The method aims to define local controllers for dynamically coupled subsystems, which share coupling constraints and minimize a separable objective function. Relying on a decomposition of the dual optimization problem such that local problems have analytical solutions, the algorithm has an iterative update procedure which converges asymptotically to the global optimizer of the primal problem. At each iteration, the controllers exchange information with other "neighboring" subsystems, with which they are "connected" in terms of dynamics or constraint coupling.

In this paper, we present an improved distributed version of Han's parallel algorithm for a class of convex optimization problems (Han and Lou, 1988; Doan et al., 2009) and show that it is applicable for DMPC of water networks. The improvements are illustrated in a simulation of the new DMPC scheme for a 4-reach canal system. The paper is organized as follows. The problem setup is described in Section 2. In Section 3, we summarize the original Han's method and the distributed version, followed by the new modified distributed version to speed up the convergence of the algorithm. The simulation results in Section 4 illustrate the properties of the DMPC scheme for the example

setup of the 4-reach canal. Section 5 concludes the paper and indicates some directions for future research.

2. PROBLEM SETUP

2.1 The canal system

In this paper we illustrate the application of the novel DMPC approach to the control of a system of irrigation canals. Irrigation canals are large systems, consisting of many interacting components, and spanning vast geographical areas. For the most safe and efficient operation of these canals, maintaining the levels of the water flows close to pre-specified reference values is crucial, both under normal operating conditions as well as in extreme situations. Manipulation of the water flows in irrigation canals is done using devices such as pumps and gates.

The example irrigation canal to be considered is a 4-reach canal system as illustrated in Figure 1. In this system, water flows from an upstream reservoir through the reaches, under the control of 4 gates and a pump at the end of the canal system that discharges water.

The control design is based on the master-slave control paradigm, in which the master controllers compute the flows through the gates, while each slave controller uses the local control actuators to guarantee the flow set by master controller (Schuurmans et al., 1999). We will use the new DMPC method to design the master controllers.

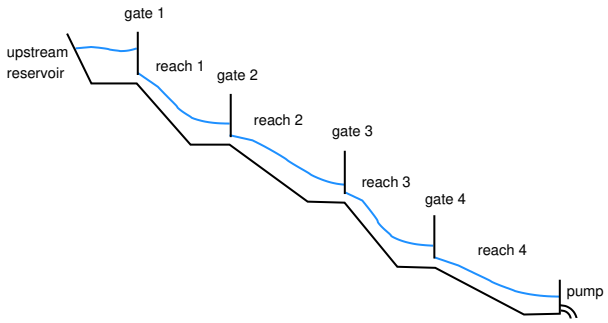

Fig. 1. The example canal system

2.2 Modeling the canal

Subsystem modeling The canal system is divided into 4 subsystems, each of which corresponds to a reach and also includes the local controller at the upstream gate of the reach. The 4^{th} subsystem has one more controller, corresponding to the pump at its downstream end.

We first use a simplified model for each subsystem as illustrated in Figure 2, and then obtain an overall model by connecting subsystem models. A subsystem is approximately modeled by a reservoir with upstream in-flow and downstream out-flow.

The discrete-time model of reach i is represented by:

$$h^i_{k+1} - h^i_k = \frac{T_\mathrm{s}}{A^i_\mathrm{s}} \left[\left(Q^i_\mathrm{in}\right)_k - \left(Q^i_\mathrm{out}\right)_k \right] \tag{1}$$

where superscript i represents the subsystem index, subscript k is for the time index, T_s is the sampling time, h is the downstream water level of the reach (zero level is set at the autonomous steady state), A_s is the water surface (volume of reservoir $= h \cdot A_\mathrm{s}$), Q_in and Q_out are in-flow and out-flow of the canal which are measured at the upstream and downstream ends, respectively. Denote the flow passing i^{th} gate by q^i, and the flow passing the pump by p^4. Due to the mass conservation law, we have $Q^i_\mathrm{out} = Q^{i+1}_\mathrm{in} = q^{i+1}$, for $i = 1, 2, 3$, and $Q^4_\mathrm{out} = p^4$.

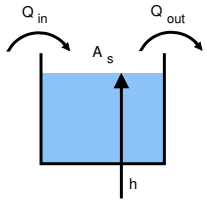

Fig. 2. Model of a reach

In order to derive local dynamics, we choose input and state vectors of subsystem i as

$$x^i_k = h^i_k$$
$$u^i_k = \begin{cases} q^i_k \ , & i = 1, 2, 3 \\ \begin{bmatrix} q^i_k \\ p^i_k \end{bmatrix} , & i = 4 \end{cases}$$

The dynamics of each subsystem can be represented by a discrete-time, linear time-invariant model of the form:

$$x^i_{k+1} = \sum_{j=1,\cdots,4} A^{ij} x^j_k + B^{ij} u^j_k, \tag{2}$$

with the state-space matrices:

$$A^{ii} = 1 \ , \quad i = 1, \cdots, 4; \quad A^{ij} = 0 \ , \quad i \neq j$$
$$B^{ii} = T_\mathrm{s}/A^i_\mathrm{s} \ , \quad i = 1, 2, 3; \quad B^{44} = \begin{bmatrix} T_\mathrm{s}/A^4_\mathrm{s} & -T_\mathrm{s}/A^4_\mathrm{s} \end{bmatrix}$$
$$B^{i(i+1)} = -T_\mathrm{s}/A^i_\mathrm{s} \ , \quad i = 1, 2; \quad B^{34} = \begin{bmatrix} -T_\mathrm{s}/A^4_\mathrm{s} & 0 \end{bmatrix}$$
$$B^{ij} = 0, \quad j \notin \{i, i+1\}$$

Centralized MPC problem The centralized MPC problem makes use of a quadratic cost function:

$$J = \sum_{i=1}^{4} \sum_{k=0}^{N-1} \left(\left(u^i_k\right)^T R_i u^i_k + \left(x^i_{k+1}\right)^T Q_i x^i_{k+1} \right) \tag{3}$$

in which N is the prediction horizon, and $\{Q_i, R_i\}_{i=1,\cdots,4}$ are given positive definite weights. It is easy to verify that this cost function can be rewritten as $J = \mathbf{x}^T H \mathbf{x}$ where H is a block-diagonal, positive definite matrix.

The constraints of the optimization problem include dynamical constraints (i.e. the model equations), initial state constraint, terminal constraint $x^i_N = 0, i = 1, \cdots, 4$, and local state and input constraints:

$$|u^i_k| \leq u^i_\mathrm{max}, \qquad |x^i_k| \leq x^i_\mathrm{max}$$

Following the method of Doan et al. (2009), the optimization problem to be solved by the centralized MPC controller at each sampling interval can be represented in a compact form as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T H \mathbf{x} \\ \text{s.t.} \quad & a_l^T \mathbf{x} = b_l, \quad l = 1, \ldots, n_\mathrm{eq} \\ & a_l^T \mathbf{x} \leq b_l, \quad l = n_\mathrm{eq} + 1, \ldots, s \end{aligned} \tag{4}$$

with $s = n_\mathrm{eq} + n_\mathrm{ineq}$, where n_eq and n_ineq are the number of scalar equality and inequality constraints, respectively.

3. DISTRIBUTED MODEL PREDICTIVE CONTROL METHOD

The optimization problem (4) will be solved by a distributed algorithm that is based on Han's parallel method for convex programs (Han and Lou, 1988). In the following we will give a summary of Han's method for convex quadratic programs, and then describe the distributed version of Han's method for quadratic programs in the form (4). Then we will proceed by describing the modified distributed version, which is the main contribution in this paper.

3.1 Han's parallel method for convex quadratic programs

The original Han's method considers general convex optimization problems where the constraint is an intersection of many convex sets. The algorithm is based on Fenchel's duality to perform a dual decomposition, and iteratively projects the dual variables onto local constraint sets. The sum of dual variables can be shown to converge to the minimizer of the dual problem (Han and Lou, 1988). A simplified version of Han's method for the quadratic optimization problem (4) is summarized in Algorithm 1.

Algorithm 1. **Han's method for convex programs**

Choose parameter α big enough [1] . For $p = 1, 2, \ldots$:

1) For $l = 1, \ldots, s$, find $\mathbf{z}_l^{(p)}$ that solves

$$\min_{\mathbf{z}} \quad \frac{1}{2}\|\mathbf{z} + \alpha \mathbf{y}_l^{(p-1)} - \mathbf{x}^{(p-1)}\|_2^2$$
$$\text{s.t.} \quad a_l^T \mathbf{z} = b_l \quad \text{or} \quad a_l^T \mathbf{z} \leq b_l$$

2) Assign $\mathbf{y}_l^{(p)} = \mathbf{y}_l^{(p-1)} + (1/\alpha)\left(\mathbf{z}_l^{(p)} - \mathbf{x}^{(p-1)}\right)$
3) Set $\mathbf{y}^{(p)} = \mathbf{y}_1^{(p)} + \cdots + \mathbf{y}_s^{(p)}$
4) Compute: $\mathbf{x}^{(p)} = H^{-1}\mathbf{y}^{(p)}$

In this representation, each vector $\mathbf{y}_l$ is a dual variable corresponding to l^{th} constraint. For problem (4), Han's method was proved to converge to the global optimum if the cost function is strongly convex, or equivalently if H is positive definite (Han and Lou, 1988). An interesting property of this method is that the number of parallel processes is equal to the number of constraints (as opposed to other dual decomposition methods where the number of parallel processes often equals the number of variables).

3.2 Distributed version of Han's method

Han's algorithm involves calculation of the global variables, therefore a global coordination method is required. A distributed version of Han's method was proposed by Doan et al. (2009), and it makes use of the explicit solutions in Step 1 of Algorithm 1, and exploits the structure of (4) to decompose the computations, hence avoiding global communications.

[1] Han and Lou (1988) recommended $\alpha = \alpha_0 \triangleq s/\rho$, where s is the number of constraints and ρ is one half of the smallest eigenvalue of H.

The main idea behind the distributed version of Han's method is illustrated in Figures 3 and 4, with a simple system consisting of 4 subsystems and the coupling matrix that shows how subsystems are coupled via their variables (boxes on the same row illustrate the variables that are coupled in one constraint). In Han's method using global variables, a subsystem has to communicate with all other subsystems in order to compute the updates of the global variables. For the distributed version of Han's method, each subsystem only communicates with the other subsystems of which the variables are necessary for computing the updates of its local variables.

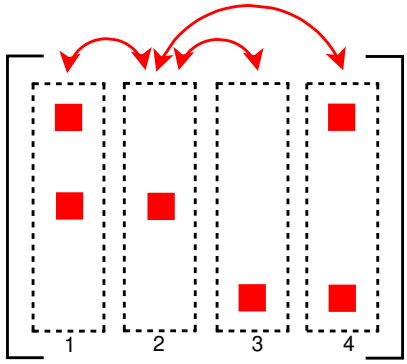

Fig. 3. Communication links of the 2nd subsystem in the *centralized* coordination version of Han's algorithm for an example 4-subsystem problem. An update for a global variable requires the 2nd subsystem to communicate with all the others.

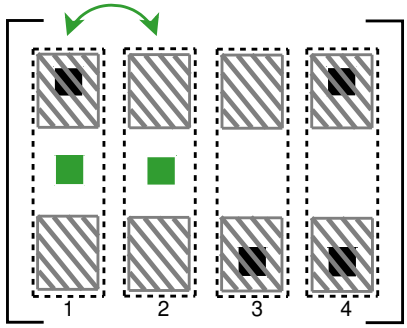

Fig. 4. Communication link of the 2nd subsystem in the *distributed* coordination version of Han's algorithm for an example 4-subsystem problem. The 2nd subsystem only cares about its local variable, therefore it does not need to communicate with the others that do not couple with it.

The distributed version of Han's method was proved to achieve the same convergence property as the original method of Han (Doan et al., 2009).

3.3 Modifications of Han's method to speed up convergence

A disadvantage of Han's method (and its distributed version) is the slow convergence rate, due to the fact that it is essentially a projection method to solve the dual problem of (4). Therefore, we need to modify the method to achieve better convergence rate.

In this paper, we present 2 modifications of the distributed version of Han's method:

- Scaling of the step sizes related to dual variables by using different α_l values for the update of each dual variable l instead of the same α for all dual variables.
- Use of nonzero initial guesses, which allows taking the current MPC solution as the start for the next sample step.

We will use the same notations as in Doan et al. (2009, Section VI), which are briefly summarized below:

- L_i: the set of indices of constraints that subsystem i is responsible for updating their dual variables throughout the algorithm.
- $\mathcal{N}^i$: the *neighborhood* of subsystem i, consisting of i itself and other subsystems that have direct dynamical or constraint couplings with subsystem i.
- $L_{\mathcal{N}^i}$: the set of indices of constraints within responsibility of all subsystems in $\mathcal{N}^i$.
- $\mathbf{x}^{(p)|i}$: the *self image* of the global variable vector $x^{(p)}$ made by subsystem i; this vector has the same size as $\mathbf{x}^{(p)}$, containing all variables of subsystem i at the right positions, and zeros for the other entries.
- $\mathbf{x}^{(p)|\mathcal{N}^i}$: the *neighborhood image* of $x^{(p)}$ made by subsystem i, using variables of all subsystems inside $\mathcal{N}^i$ at the right positions, and zeros for the other entries.
- $\mathbf{x}^{(p)|\mathcal{N}^i}_{\text{assumed}}$: the *assumed neighborhood image* $x^{(p)}$ made by subsystem i. The difference between $\mathbf{x}^{(p)|\mathcal{N}^i}_{\text{assumed}}$ and $\mathbf{x}^{(p)|\mathcal{N}^i}$ is that only the values of variables belonging to subsystem i are correct, while for the variables of other neighboring subsystems $j \in \{\mathcal{N}^i \setminus i\}$, the values could be different from the real ones.
- $\mathfrak{I}^i$: index matrix of subsystem i; it is the mask for the global variable $\mathbf{x}$ such that only variables of subsystem i are kept, i.e. $\mathbf{x}^{(p)|i} = \mathfrak{I}^i\mathbf{x}^{(p)}$.

We present the improved distributed version of Han's method in the following algorithm:

Algorithm 2. **Improved distributed algorithm for the MPC optimization problem**

Pre-computed parameters: Each subsystem i computes and stores the following parameters throughout the control scheme:

- For each $l \in L_i$: $\alpha_l = \big(k_\alpha\big)_l \alpha_0$, where k_α is the scaling vector. α_l acts as local step size regarding l^{th} dual variable, and therefore k_α should be chosen such that the convergence rates of all s dual variables are improved. The method to choose k_α will be discussed in this section.
- For each $l \in L_i$: $\bar{c}_l = \frac{-1}{a_l^T a_l} H^{-1} a_l$. We can see that $\bar{c}_l$ can be computed locally by a local controller with *a priori* knowledge of the parameter a_l and the weighting blocks on the diagonal of H that correspond to the non-zero elements of a_l.

MPC step:

At the beginning of the MPC step, the current states of all subsystems are measured. The sequences of predicted states and inputs generated in the previous MPC step are shifted forward one step, then we add zero states and zero inputs to the end of the shifted sequences. The new sequences are then used as the initial guess for solving the optimization problem in the current MPC step. The initial guess for each subsystem can be defined locally. For subsystem i, denote the initial guess as $\mathbf{x}^{(0)|i}$. At the first MPC step, we have $\mathbf{x}^{(0)|i} = 0, \forall i$.

The idea of using previously predicted states and inputs for initialization is a popular technique in MPC (Rawlings and Mayne, 2009). Especially with Han's method, whose convergence rate is slow, an initial guess that is close to the optimal solution will be very helpful to reduce the number of iterations.

The current state is plugged into the MPC problem, then we get an optimization problem of the form (4). This problem will be solved in a distributed way by the following iterative procedure.

Distributed iterative procedure to solve the optimization problem:

Initialize with $p = 0$. Each subsystem i communicates with the neighbors $j \in \mathcal{N}^i$ to get $\mathbf{x}^{(0)|j}$, then constructs $\mathbf{x}^{(0)|\mathcal{N}^i} = \sum_{j\in\mathcal{N}^i} \mathbf{x}^{(0)|j}$. Subsystem i computes its local dual variable $\mathbf{y}^{(0)|\mathcal{N}^i} = H\mathbf{x}^{(0)|\mathcal{N}^i}$, and then computes initial intermediate variables:

$$\gamma_l^{(0)} = \max\{a_l^T(\mathbf{x}^{(0)|\mathcal{N}^i} - \mathbf{y}^{(0)|\mathcal{N}^i}) - b_l, 0\}, \quad l \in L_i$$

Next, for $p = 1, 2, \ldots,$ the following steps are executed:

1) **Communications to get the updated main variables**
Each controller i communicates with its neighbors $j \in \mathcal{N}^i$ to get updated values of their variables, contained in $\mathbf{x}^{(p-1)|j}$. Vice versa, i also sends its updated variables in $\mathbf{x}^{(p-1)|i}$ to its neighbors as requested.
After getting information from the neighbors, controller i constructs the *neighborhood image* $\mathbf{x}^{(p-1)|\mathcal{N}^i}$ as:
$$\mathbf{x}^{(p)|\mathcal{N}^i} = \sum_{j\in\mathcal{N}^i} \mathbf{x}^{(p)|j}$$
2) **Update intermediate variables γ_l in parallel**
In this step, the local controllers update γ_l corresponding to each constraint l under their responsibility. More specifically, each local controller i updates γ_l for each $l \in L_i$ in the following manner:
   - If constraint l is an equality constraint ($l \in \{1, \ldots, n_{\text{eq}}\}$), then $\gamma_l^{(p)} = a_l^T\mathbf{x}^{(p-1)|\mathcal{N}^i} + \gamma_l^{(p-1)} - b_l$.
   - If constraint l is an inequality constraint ($l \in \{n_{\text{eq}}+1, \ldots, s\}$), then $\gamma_l^{(p)} = \max\{a_l^T\mathbf{x}^{(p-1)|\mathcal{N}^i} + \gamma_l^{(p-1)} - b_l, 0\}$.
3) **Communications to get the updated intermediate variables**
Each local controller i communicates with its neighbors to get updated $\gamma_l^{(p)}$ values that the neighbors just computed in Step 2).
4) **Update main variables in parallel**
Local controller i uses all $\gamma_l^{(p)}$ values that it has (by communications and those computed by itself) to compute an *assumed neighborhood image* of $\mathbf{x}$:

$$\mathbf{x}_{\text{assumed}}^{(p)|\mathcal{N}^i} = \sum_{l \in L_{\mathcal{N}^i}} \frac{1}{\alpha_l} \gamma_l^{(p)} \bar{c}_l \tag{5}$$

Then controller i selects the values of its variables in $\mathbf{x}_{\text{assumed}}^{(p)|\mathcal{N}^i}$ to construct the new *self image*:

$$\mathbf{x}^{(p)|i} = \mathfrak{I}^i \mathbf{x}_{\text{assumed}}^{(p)|\mathcal{N}^i} \tag{6}$$

which contains $u_0^{i,(p)}, \ldots, u_{N-1}^{i,(p)}, x_1^{i,(p)}, \ldots, x_N^{i,(p)}$.

After updating their variables, each local controller checks the local termination criteria. When all local controllers have converged [2], the algorithm stops and the local control actions are implemented, otherwise the controllers proceed to Step 1) to start a new iteration.

Implement MPC input:

When the iterative procedure finishes, each subsystem applies the first input $u_0^{i,(p)}$, then waits for the next state measurement to start a new MPC step.

Method to choose the scaling vector:

In the modified version of distributed Han's method, a good choice of the scaling vector helps to dramatically improve the convergence speed. We have observed that the convergence rate of some dual variables under the responsibility of a subsystem i will affect the convergence rate of dual variables under the responsibility of its neighbors in $\mathcal{N}^i$. Therefore the choice of scaling vector should focus on improving the convergence rate of "slower convergent" dual variables. In our simulation, we rely on the Hessian to find the scaling vector. Specifically, for a subsystem i whose variables have the average weight $\bar{h}_i$ (e.g. average of entries related to i's states and inputs in the diagonal of the Hessian), we choose the scale factor $\left(k_\alpha\right)_l = 1/\bar{h}_i$, with all $l \in L_i$. We also multiply the scaling vector k_α with a factor $\theta < 0$ for enlarging the step sizes of all dual variables; this θ is tuned in the first MPC step.

The choice of the scaling vector depends on the structure of the centralized optimization problem, thus we only need to choose it once in the first MPC step. Then for the next MPC steps, we can reuse the same scaling vector.

4. SIMULATION RESULTS AND DISCUSSION

DMPC methods are applied to the regulation problem of the simulated canal system of Section 2, which has a perturbed initial state. We use distributed Han's method with and without the modifications described in Section 3.3 for the same setup, and compare the results. Figure 5 shows that the distributed Han's method with modifications achieves better convergence rate, allowing the distributed optimization to converge within an acceptable number of iterations. A simulation of closed-loop MPC is performed for 20 sample steps. Figure 6 shows that the distributed solutions converge to the centralized solutions in every sample step.

[2] Checking the termination criteria in a distributed fashion requires a dedicated logic scheme, the description of which is omitted for brevity.

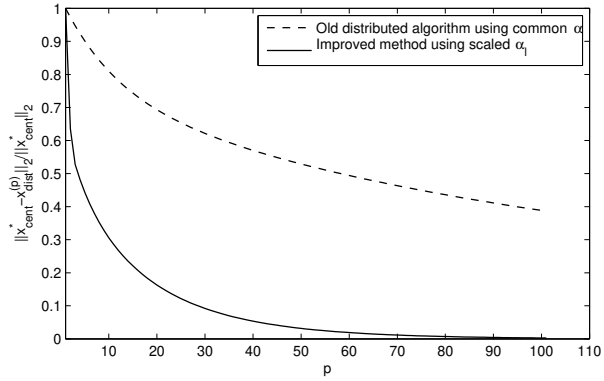

Fig. 5. Comparison of convergence rates of the former and the new distributed versions of Han's method in the first sampling time (k=1)

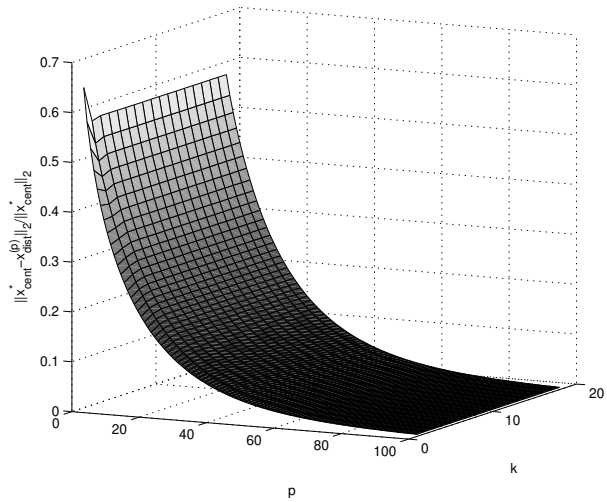

Fig. 6. Normalized norm of difference between the centralized and the distributed solutions versus the iteration step p and sample step k

Although the new scheme is verified by this simulation, there are still several theoretical issues that need to be addressed:

Firstly, there is no convergence proof for the modified distributed version of Han's method yet. We observe that in setups that are more complicated, the method to choose scaling vector proposed in this simulation does not always work well (sometimes after several sample steps, the algorithm does not converge in the next sample steps). Note that with this method we aim to solve the dual problem, therefore the primal iterate would be infeasible unless the algorithm converges.

Secondly, in the MPC formulation we keep both inputs and states as variables of the centralized optimization problem. This formulation is advantageous in distributed MPC because the Hessian will have a diagonal structure, and the *neighborhood* of each subsystem will only contain its direct neighbors (the *neighborhood* would be greatly extended if we eliminate the states in the optimization problem). However, using states as variables requires considering the dynamical equations as equality constraints of the optimization problem, and the existence of equality constraints typically requires an exact solution in order

to guarantee feasibility. In future research, we will also study MPC formulations in which all states are eliminated, so that the centralized optimization only has inequality constraints. Such formulation would allow stopping the algorithm in a finite number of steps, and the final iterate could be feasible (although it may be suboptimal).

Another problem is that the proposed method is for *quadratic programs* only. Although many MPC problems for linear time-invariant systems are formulated as quadratic programs, there are other variants that use different objective functions, and nonlinear MPC would also yield more complicated optimization problems than quadratic programs. With such problems, we might not be able to implement Han's parallel method in a distributed fashion. This issue motivates the research for other decomposition methods that can handle more general problems, e.g. convex problems with linear or decoupled nonlinear constraints.

Last but not least, the MPC formulation in this paper employs the terminal constraint $x_N = 0$, which is conservative since it reduces the domain of attraction of MPC. An improvement could be made by replacing this constraint with less restrictive conditions (e.g. terminal constraint set and terminal controller). However, there is still no distributed scheme to construct the terminal constraint set and the terminal controller (and also the terminal penalty matrix that is solution of the Riccati equation), other than assuming them to be completely decoupled.

5. CONCLUSIONS

The modified distributed version of Han's method has an improved convergence rate, thus it is more suitable for DMPC of large-scale water networks. Future research will involve finding a way to construct the scaling vector of the modified distributed version of Han's method together with a theoretical proof of the convergence. We will also investigate different distributed optimization methods using dual decomposition techniques to address nonlinear MPC with more general optimization problem. Another direction is to find distributed MPC schemes for suboptimal MPC.

ACKNOWLEDGEMENTS

Research supported by the European 7th framework STREP project "Hierarchical and distributed model predictive control", contract number INFSO-ICT-223854.

REFERENCES

Alessio, A. and Bemporad, A. (2007). Decentralized model predictive control of constrained linear systems. In *European Control Conference*, 2813–2818. Kos, Greece.

Alessio, A. and Bemporad, A. (2008). Stability conditions for decentralized model predictive control under packet drop communication. In *American Control Conference*, 3577–3582.

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

Camponogara, E. and Talukdar, S. (2007). Distributed model predictive control: Synchronous and asynchronous computation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(5), 732–745.

Doan, D., Keviczky, T., Necoara, I., Diehl, M., and De Schutter, B. (2009). A distributed version of Han's method for DMPC using local communications only. *Control Engineering and Applied Informatics*, 11(3), 6–15.

Du, X., Xi, Y., and Li, S. (2001). Distributed model predictive control for large-scale systems. In *American Control Conference*, volume 4, 3142–3143.

Dunbar, W.B. and Murray, R.M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42, 549–558.

Han, S.P. and Lou, G. (1988). A parallel algorithm for a class of convex programs. *SIAM Journal on Control and Optimization*, 26(2), 345–355.

Jia, D. and Krogh, B. (2002). Min-max feedback model predictive control for distributed control with communication. In *American Control Conference*, volume 6, 4507–4512.

Jia, D. and Krogh, B. (2001). Distributed model predictive control. In *American Control Conference*, volume 4, 2767–2772.

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

Li, S., Zhang, Y., and Zhu, Q. (2005). Nash-optimization enhanced distributed model predictive control applied to the shell benchmark problem. *Information Sciences*, 170(2-4), 329–349.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, England.

Mercangoz, M. and Doyle III, F.J. (2007). Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17(3), 297–308.

Necoara, I., Doan, D., and Suykens, J. (2008). Application of the proximal center decomposition method to distributed model predictive control. In *IEEE Conference on Decision and Control*, 2900–2905.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin.

Rawlings, J.B. and Stewart, B.T. (2008). Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18(9), 839–845.

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

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19(5), 723–731.

Schuurmans, J., Hof, A., Dijkstra, S., Bosgra, O.H., and Brouwer, R. (1999). Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering*, 125(4), 189–195.

Venkat, A., Hiskens, I., Rawlings, J., and Wright, S. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6), 1192–1206.