Delft University of Technology

Delft Center for Systems and Control

Technical report 13-001

A hierarchical MPC approach with guaranteed feasibility for dynamically coupled linear 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, "A hierarchical MPC approach with guaranteed feasibility for dynamically coupled linear systems," in *Distributed Model Predictive Control Made Easy* (R.R. Negenborn and J.M. Maestre, eds.), vol. 69 of *Intelligent Systems, Control and Automation: Science and Engineering*, Dordrecht, The Netherlands: Springer, ISBN 978-94-007-7005-8, pp. 393–406, 2014.

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.deschutter.info/abs/13_001.html`

A hierarchical MPC approach with guaranteed feasibility for dynamically coupled linear systems

Minh Dang Doan, Tamás Keviczky and Bart De Schutter

Abstract In this chapter we describe an iterative two-layer hierarchical approach to MPC of large-scale linear systems subject to coupled linear constraints. The algorithm uses constraint tightening and applies a primal-dual iterative averaging procedure to provide feasible solutions in every sampling step. This helps overcome typical practical issues related to the asymptotic convergence of dual decomposition based distributed MPC approaches. Bounds on constraint violation and level of suboptimality are provided. The method can be applied to large-scale MPC problems that are feasible in the first sampling step and for which the Slater condition holds (i.e., there exists a solution that strictly satisfies the inequality constraints). Using this method, the controller can generate feasible solutions of the MPC problem even when the dual solution does not reach optimality, and closed-loop stability is also ensured using bounded suboptimality.

1 Introduction

When there are couplings among linear subsystems in a distributed MPC problem, dual decomposition is often used in order to divide the computational tasks among the subsystems. A typical requirement of the dual decomposition-based methods is that the dual problem needs to be solved exactly in order to obtain a primal feasible solution [1]. However, iterative approaches based on dual decomposition often only converge asymptotically to the optimum, which may not be practical when implementing these approaches in a real-time environment. In this chapter, we apply a dual decomposition technique to the class of MPC problems for linear systems with coupled dynamics and coupled linear constraints, and propose a novel method that is motivated by the use of constraint tightening in robust MPC [5]. This method

M. D. Doan, T. Keviczky, B. De Schutter
Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, The Netherlands, e-mail: m.d.doan@tudelft.nl, t.keviczky@tudelft.nl, b.deschutter@tudelft.nl

allows terminating the iterations for the dual problem before reaching convergence while still guaranteeing a feasible primal solution to be found. Moreover, the algorithm also generates a decreasing cost function, leading to closed-loop stability. In summary, the proposed framework guarantees primal feasible solutions and MPC stability using a finite number of iterations with bounded suboptimality.

2 Boundary conditions

Our approach aims at large-scale interconnected systems with constrained discrete-time linear time-invariant dynamics where some of the individual constraints can also be defined over more than one subsystem (coupling constraints). This class includes a wide range of applications, e.g., water and power distribution networks, urban traffic networks, industrial processes, arrays of mechanical actuators, and climate control systems, among others [8, 9, 6, 12, 7]. Mathematically, let M be the number of subsystems ($\mathscr{N} = \{1,\dots,M\}$), and let the dynamical model of each subsystem be represented in the following form:

$$x_{k+1}^i = \sum_{j=1}^{M} \mathbf{A}^{ij} x_k^j + \mathbf{B}^{ij} u_k^j, \quad i \in \mathscr{N} \tag{1}$$

The corresponding centralized state-space model is written in a compact form:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \tag{2}$$

where $x_k = [(x_k^1)^T (x_k^2)^T \dots (x_k^M)^T]^T, u_k = [(u_k^1)^T (u_k^2)^T \dots (u_k^M)^T]^T$, $\mathbf{A} = [\mathbf{A}_{ij}]_{i,j\in\mathscr{N}}$ and $\mathbf{B} = [\mathbf{B}_{ij}]_{i,j\in\mathscr{N}}$.

We formulate an MPC problem using a terminal penalty and a terminal constraint set. At a particular time step t the MPC optimization problem is defined as follows:

$$\min_{\mathbf{u},\mathbf{x}} \quad \sum_{k=t}^{t+N_\mathrm{p}-1} \left(x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k \right) + x_{t+N_\mathrm{p}}^T \mathbf{P} x_{t+N_\mathrm{p}} \tag{3}$$

$$\text{s.t.} \quad x_{k+1}^i = \sum_{j\in\mathscr{N}^i} \mathbf{A}^{ij} x_k^j + \mathbf{B}^{ij} u_k^j, \quad i \in \mathscr{N}, \quad k = t,\dots,t+N_\mathrm{p}-1 \tag{4}$$

$$x_k \in \mathscr{X}, \quad k = t+1,\dots,t+N_\mathrm{p}-1 \tag{5}$$

$$x_{t+N_\mathrm{p}} \in \mathscr{X}_\mathrm{f} \subset \mathscr{X} \tag{6}$$

$$u_k \in \mathscr{U}, \quad k = t,\dots,t+N_\mathrm{p}-1 \tag{7}$$

$$u_k^i \in \Omega_i, \quad i \in \mathscr{N}, \quad k = t,\dots,t+N_\mathrm{p}-1 \tag{8}$$

$$x_t = x(t) \in \mathscr{X} \tag{9}$$

where $\mathbf{u} = [u_t^T, \ldots, u_{t+N_\mathrm{p}-1}^T]^T$, $\mathbf{x} = [x_{t+1}^T, \ldots, x_{t+N_\mathrm{p}}^T]^T$, the matrices $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{R}$ are block-diagonal and positive definite, the constraint sets $\mathscr{U}$, $\mathscr{X}$, and $\mathscr{X}_\mathrm{f}$ are convex polytopes and have nonempty interiors, and each local constraint set Ω_i is a hyperbox. Each subsystem i is assigned a neighborhood, denoted by $\mathscr{N}^i$, containing subsystems that have direct dynamical interactions with subsystem i, including itself. Note that the coupled constraints involving control inputs are represented by the set $\mathscr{U}$ in (7), which does not incorporate local constraints that are captured by Ω_i in (8). The initial state x_t is the current state at time step t. Notice that the MPC formulation (3)–(9) *does not* incorporate a terminal zero-point constraint ($x_{t+N_\mathrm{p}} = 0$). Moreover, the approach proposed in this chapter does not handle any equality constraints on the states except for the dynamical constraints (4), since the corresponding constraint sets would have an empty interior, which would prevent constraint tightening, the key idea of this approach.

As $\mathscr{U}$, $\mathscr{X}$, and $\mathscr{X}_\mathrm{f}$ are polytopes, the constraints (5) and (6) are represented by linear inequalities. Moreover, the state vector $\mathbf{x}$ is affinely dependent on $\mathbf{u}$. Hence, we can eliminate the state variables $x_{t+1}, \ldots, x_{t+N_\mathrm{p}}$ and transform the constraints (4), (5), and (6) into linear inequalities on the input variable $\mathbf{u}$. Eliminating the state variables in (3)–(9) leads to an optimization problem in the following form:

$$f_t^* = \min_{\mathbf{u}} \quad f(\mathbf{u}, x_t) \tag{10}$$

$$\text{s.t.} \quad g(\mathbf{u}, x_t) \leq 0 \tag{11}$$

$$\mathbf{u} \in \tilde{\Omega} \tag{12}$$

where f is a convex quadratic function, $g = [g_1, \ldots, g_m]^T$ with $g_i, i = 1, \ldots, m$ affine functions, and $\tilde{\Omega} = \Omega \times \cdots \times \Omega$ (N_p times) where $\Omega = \Omega_1 \times \cdots \times \Omega_M$, is a hyperbox. Note that $f(\mathbf{u}, x_t) > 0, \forall \mathbf{u} \neq 0, x_t \neq 0$, since $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{R}$ are positive definite.

In addition, our approach also makes use of the following assumptions:

Assumption 2.1 *There exists a block-diagonal feedback gain* $\mathbf{K}$ *such that the matrix* $\mathbf{A} + \mathbf{BK}$ *is Schur*[1] *(i.e.,* $\mathbf{K}$ *yields a decentralized stabilizing control law for the unconstrained aggregate system).*

Assumption 2.2 *The terminal constraint set* $\mathscr{X}_\mathrm{f}$ *is strictly positively invariant for the closed-loop system with* $x_{k+1} = (\mathbf{A} + \mathbf{BK})x_k$*, i.e., if* $x \in \mathrm{int}(\mathscr{X}_\mathrm{f})$ *then* $(\mathbf{A} + \mathbf{BK})x \in \mathrm{int}(\mathscr{X}_\mathrm{f})$*. In addition, for any state in* $\mathscr{X}_\mathrm{f}$*, the control input generated by the terminal controller should satisfy the input constraints, i.e.,* $-\mathbf{K}x \in \mathscr{U} \cap \Omega$, $\forall x \in \mathscr{X}_\mathrm{f}$.

Assumption 2.3 *The Slater condition holds for the constraint* (11)*, i.e., there exists a feasible vector that satisfies* (11) *with strict inequality constraints [1]. It is also assumed that prior to computing the control input to be implemented in each time step* t*, a Slater vector* $\bar{\mathbf{u}}_t$ *is available, such that*

$$\begin{cases} g_j(\bar{\mathbf{u}}_t, x_t) < 0, j = 1, \ldots, m \\ \bar{\mathbf{u}}_t \in \tilde{\Omega} \end{cases} \tag{13}$$

[1] A matrix is Schur if all of its eigenvalues are inside the unit circle.

Assumption 2.4 *At each time step t, the following inequality holds:*

$$f(\mathbf{u}_{t-1}, x_{t-1}) > f(\bar{\mathbf{u}}_t, x_t) \tag{14}$$

For later reference, we define $\Delta_t > 0$ which can be computed before time step t as follows:

$$\Delta_t = f(\mathbf{u}_{t-1}, x_{t-1}) - f(\bar{\mathbf{u}}_t, x_t) \tag{15}$$

Assumption 2.5 *For each $x_t \in \mathscr{X}$, the Euclidean norm of $g(\mathbf{u}, x_t)$ is bounded:*

$$\exists L_t : L_t \geq \|g(\mathbf{u}, x_t)\|_2, \forall \mathbf{u} \in \tilde{\Omega} \tag{16}$$

Assumptions 2.1 and 2.2 are almost standard in MPC, except that in Assumption 2.2 we require *strict positive invariance* instead of the milder condition of *positive invariance* that is often employed in MPC literature. Assumptions 2.3–2.5 will help us to provide bounds on the constraint violation and the suboptimality of the solution. These assumptions can be satisfied if the search domain is bounded, i.e., there is a lower bound and an upper bound for every control input, which is usually the case in MPC. Assumption 2.4 implies that the decreasing property of the cost function can be obtained directly by using $\bar{\mathbf{u}}_t$ as the solution at time step t, i.e., with the choice $\mathbf{u}_t = \bar{\mathbf{u}}_t$. However, this choice will likely deteriorate the performance of the MPC controller due to the increasing conservativeness of $\bar{\mathbf{u}}_t$ as t increases. Therefore, our approach will make use of $\bar{\mathbf{u}}_t$ only as a starting point, based on which a new solution is computed at every time step t. Moreover, we have developed a method in [4, 2] to find $\bar{\mathbf{u}}_t$ in every time step $t \geq 1$, which makes Assumption 2.4 easy to fulfill. A summary of this method is given later in Remark 1.

The communication architecture is assumed to be hierarchical, i.e., there is a coordinator for computing common parameters and performing limited communication with all agents. Each agent only communicates with the coordinator and a small number of other agents (belonging to set $\mathscr{N}^i$) that are considered as its *neighbors*.

3 Description of the approach

Our approach aims at solving a tightened version of problem (10)–(12) in a hierarchical manner by using a dual decomposition technique. We first construct the tightened problem and its dual problem, and then describe the algorithm.

3.1 Initialization

Prior to applying our algorithm to solve the MPC optimization problem at each time step t, there is an initialization which includes two main tasks: formulating the

tightened problem to be solved, and determining a sufficient number of iterations to be performed in that time step.

3.1.1 Formulation of the tightened problem

At the beginning of each MPC step t, a new Slater vector $\bar{\mathbf{u}}_t$ (cf. Assumption 2.3) should be determined. Based on this $\bar{\mathbf{u}}_t$, a value c_t is computed such that:

$$0 < c_t < \min_{j=1,\dots,m} \{-g_j(\bar{\mathbf{u}}_t, x_t)\} \tag{17}$$

Then we construct the tightened problem:

$$f_t'^* = \min_{\mathbf{u}} \quad f(\mathbf{u}, x_t) \tag{18}$$

$$\text{s.t.} \quad g'(\mathbf{u}, x_t) \leq 0 \tag{19}$$

$$\mathbf{u} \in \tilde{\Omega} \tag{20}$$

with the tightened constraint:

$$g'(\mathbf{u}, x_t) \triangleq g(\mathbf{u}, x_t) + \mathbf{1}_m c_t \tag{21}$$

where $g'(\mathbf{u}, x_t) = [g_1', \dots, g_m']^T$, and $\mathbf{1}_m$ is a vector of ones with m elements.

We denote the dual variable as $\mu \in \mathbb{R}_+^m$ and define the Lagrangian function:

$$\mathscr{L}(\mathbf{u}, \mu, x_t) \triangleq f(\mathbf{u}, x_t) + \mu^T g'(\mathbf{u}, x_t) \tag{22}$$

A bound L_t on the 2-norm of the constraints g should be given (cf. Assumption 2.5). Using (16) and the triangle inequality of the 2-norm, we will get $L_t' = L_t + c_t$ as a norm bound for g', i.e., $L_t' \geq \|g'(\mathbf{u}, x_t)\|_2, \forall \mathbf{u} \in \tilde{\Omega}$.

Using the Slater vector $\bar{\mathbf{u}}_t$, we then compute Δ_t using (15). After that we choose two parameters $\alpha_t > 0$ and $\varepsilon_t > 0$ such that $\alpha_t L_t'^2/2 + \varepsilon_t \leq \Delta_t$. Later α_t will be used as the step size of the algorithm for maximizing the dual function, and ε_t will be the desired suboptimality of the algorithm for minimizing the Lagrangian function.

Remark 1. In [4, 2], we describe a method to update L_t and to construct a new Slater vector $\bar{\mathbf{u}}_t$ for each time step $t \geq 1$. In summary, L_t is updated by taking into account the change of the initial state from x_{t-1} to x_t; and the new $\bar{\mathbf{u}}_t$ is constructed by shifting the previous solution $\mathbf{u}_{t-1}$ one-step ahead and adding the terminal state and terminal input, which are feasible due to Assumption 2.2. For the first MPC step, i.e., $t = 0$, these values can be computed offline as follows.

In order to determine L_0, we formulate a maximization problem to find the maximum 2-norm of $g(\mathbf{u}, x_0)$, in which $\mathbf{u} \in \tilde{\Omega}$ and $x_0 \in \mathscr{X}$. This optimization problem has a convex cost function and a convex constraint set, and hence the maximum will occur at one of the vertices of the constraint set. As a consequence, finding the

solution of this maximization problem is straightforward by evaluating the 2-norm of g at all the vertices. The maximal function value is then assigned to L_0.

For finding $\bar{\mathbf{u}}_0$, we can start with a guess for c_t, then tighten the constraints using (21), and use a centralized solver to find a feasible point of the constraints $g'(\mathbf{u}, x_t) \leq 0$. If this problem is feasible, then the feasible input sequence obtained can be set as $\bar{\mathbf{u}}_0$, and we may increase c_t to find a better Slater vector. If this problem is infeasible, then we reduce c_t until a feasible solution is obtained, which can be used as $\bar{\mathbf{u}}_0$.

3.1.2 Determination of the number of iterations

The problem (18)–(20) will be solved by a nested iteration. In the outer loop, the dual function is maximized using a projected gradient method in combination with an averaging scheme that provides bounds for the constraint violation and the cost function. In the inner loop, a hierarchical optimization algorithm is used to provide - with a desired precision - an approximate solution to the minimization of the Lagrangian function. The outer loop is executed for $\tilde{k}_t$ iterations, each of which includes $\tilde{p}_t$ iterations of the inner loop.

The algorithm is simple and is presented in Section 3.2. Before starting the iterative algorithm, we let a coordinator determine $\tilde{k}_t$ and $\tilde{p}_t$.

As described in [4, 2], we can determine a sufficient minimum number of outer iterations $\tilde{k}_t$ as follows:

$$\tilde{k}_t \geq \frac{1}{\alpha_t c_t}\left(\frac{3}{\gamma_t} f(\bar{\mathbf{u}}_t, x_t) + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t'\right) \tag{23}$$

where $\gamma_t = \min_{j=1,\ldots,m}\{-g_j'(\bar{\mathbf{u}}_t, x_t)\} = \min_{j=1,\ldots,m}\{-g_j(\bar{\mathbf{u}}_t, x_t)\} - c_t$. The number of inner iterations $\tilde{p}_t$ can be computed by:

$$\tilde{p}_t \geq \log_\phi \frac{\varepsilon_t}{\Lambda M \max_i D_i} \tag{24}$$

where Λ is the Lipschitz constant of the Lagrangian function $\tilde{\mathscr{L}}$ over the set $\tilde{\Omega}$, D_i is the diameter of the set $\tilde{\Omega}_i$ with respect to the Euclidean norm, i.e., $\|\mathbf{u}_1^i - \mathbf{u}_2^i\|_2 \leq D_i, \forall \mathbf{u}_1^i, \mathbf{u}_2^i \in \tilde{\Omega}_i$, with $\tilde{\Omega}_i \triangleq \Omega_i \times \cdots \times \Omega_i$ (N_p times). The value $\phi \in [0,1)$ is the contraction modulus of the Jacobi algorithm, i.e., the decay rate of the distance from a current iterate to the optimizer, and can be computed by [4]:

$$\phi = \max_i \left\{ \max \left\{ 2\gamma\big(\lambda_{\max}(\mathbf{H}_{ii}) + \sum_{j\neq i} \bar{\sigma}(\mathbf{H}_{ij})\big) - 1, \right.\right. \\ \left.\left. 1 - 2\gamma\big(\lambda_{\min}(\mathbf{H}_{ii}) - \sum_{j\neq i} \bar{\sigma}(\mathbf{H}_{ij})\big) \right\} \right\} \tag{25}$$

where $\mathbf{H}_{ij}$ with $i, j \in \{1, \ldots, M\}$ denotes the submatrix of $\mathbf{H}$ (the Hessian of the Lagrangian function), containing entries at rows belonging to subsystem i and

columns belonging to subsystem j. The notation $\lambda_{\max}(\mathbf{H}_{ii})$ and $\lambda_{\min}(\mathbf{H}_{ii})$ stands for the maximum and minimum eigenvalue of $\mathbf{H}_{ii}$, respectively, while $\bar{\sigma}(\mathbf{H}_{ij})$ denotes the maximum singular value of $\mathbf{H}_{ij}$. Note that depending on the matrices $\mathbf{H}_{ij}, i, j \in \{1,\dots,M\}$, the Jacobi algorithm in the inner loop may or may not converge. When ϕ computed by (25) falls into $[0,1)$, then it means the Jacobi algorithm converges and can be used. Otherwise, another algorithm should be used in the inner loop instead of the Jacobi algorithm; in this case a hierarchical conjugate gradient method can be employed, see [3] for more details.

3.2 The proposed algorithm

In the following, we present our proposed algorithm for solving the MPC problem at each time step t. Note that prior to applying the algorithm, we need to determine the Lagrangian and the parameters ε_t, α_t as described in the previous section.

Algorithm 1 ***Hierarchical Primal Feasible method with Dual Approximate Gradient (HPF-DAG)***

1. ***Outer loop:*** *Set* $\mu^{(0)} = \mathbf{0}_m$. *For* $k = 0,\dots,\tilde{k}_t$, *find* $\mathbf{u}^{(k)}, \mu^{(k+1)}$ *such that:*

$$\tilde{\mathscr{L}}(\mathbf{u}^{(k)}, \mu^{(k)}, x_t) \leq \min_{\mathbf{u}\in\tilde{\Omega}} \tilde{\mathscr{L}}(\mathbf{u}, \mu^{(k)}, x_t) + \varepsilon_t \tag{26}$$

$$\mu^{(k+1)} = \mathscr{P}_{\mathbb{R}^m_+}\left\{\mu^{(k)} + \alpha_t d^{(k)}\right\} \tag{27}$$

where $\mathscr{P}_{\mathbb{R}^m_+}$ *denotes the projection onto the nonnegative orthant,* $d^{(k)} = g'\left(\mathbf{u}^{(k)}, x_t\right)$ *is an* ε_t*-subgradient of the dual function at* $\mu^{(k)}$.

Inner loop:

- *Solve problem* (26) *in a distributed way with a Jacobi algorithm. For* $p = 0,\dots,\tilde{p}_t$, *every subsystem i computes:*

$$\mathbf{u}^i(p+1) = \arg\min_{\mathbf{u}^i\in\tilde{\Omega}_i} \tilde{\mathscr{L}}(\mathbf{u}^1(p),\dots,\mathbf{u}^{i-1}(p), \mathbf{u}^i, \mathbf{u}^{i+1}(p),\dots,\mathbf{u}^M(p), \mu^{(k)}) \tag{28}$$

  where $\tilde{\Omega}_i$ *is the local constraint set for control variables of subsystem i.*
- *Define* $\mathbf{u}^{(k)} \triangleq [\mathbf{u}^1(\tilde{p}_t)^T,\dots,\mathbf{u}^M(\tilde{p}_t)^T]^T$, *which is guaranteed to satisfy* (26).

2. *Compute* $\hat{\mathbf{u}}^{(\tilde{k}_t)} = \frac{1}{\tilde{k}_t}\sum_{l=0}^{\tilde{k}_t} \mathbf{u}^{(l)}$, *and take* $\mathbf{u}_t = \hat{\mathbf{u}}^{(\tilde{k}_t)}$ *as the solution of* (10)–(12).

Note that the computations (27) and (28) are performed by the subsystems, with synchronization provided by the coordinator. While solving the local problem (28), each subsystem needs to communicate with its neighbors to get the latest update for constructing the proper cost function in the current inner loop iteration.

4 Availability of theoretical results

In this section, we recall the theoretical guarantees of our method in the form of two propositions.

Proposition 1. *Suppose Assumptions 2.1–2.5 hold. Construct g' as in* (21)*. Let the outer loop* (26)–(27) *with $\mu^{(0)} = \mathbf{0}_m$ be iterated for $k = 0, \ldots, \tilde{k}_t$. Then $\hat{\mathbf{u}}^{(\tilde{k}_t)}$ computed in Algorithm 1 is a feasible solution of* (10)–(12)*.*

This result is based on the bound of the constraint violation. Let g'^{+} denote the constraint violation, i.e., $g'^{+} = \max\{g', \mathbf{0}_m\}$, then the primal average sequence $\hat{\mathbf{u}}^{(k)} = \frac{1}{k}\sum_{l=0}^{k} \mathbf{u}^{(l)}$ generated at iteration $k \geq 1$ satisfies:

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(k)}, x_t\right) \right]^{+} \right\|_2 \leq \frac{1}{k\tilde{\alpha}_t} \left(\frac{3}{\gamma_t} [f(\bar{\mathbf{u}}_t, x_t) - \tilde{q}'^{*}_t] + \frac{\tilde{\alpha}_t L_t'^2}{2\gamma_t} + \tilde{\alpha} L'_t \right) \tag{29}$$

The proof for (29) and Proposition 1 can be found in [4]. Proposition 1 guarantees that the solution of the algorithm is feasible for implementation.

Let us recall that the optimization problem formulation is motivated by an MPC problem for which Assumption 2.4 holds. The following proposition shows that the cost function of the MPC problem is a decreasing function.

Proposition 2. *Suppose Assumptions 2.1–2.5 hold. Then the solution $\hat{\mathbf{u}}^{(\bar{k}_t)}$ generated by Algorithm 1 satisfies the following inequality:*

$$f(\mathbf{u}_t, x_t) < f(\mathbf{u}_{t-1}, x_{t-1}), \quad \forall t \in \mathbb{N} \backslash \{0\} \tag{30}$$

This result is made possible using the upper bound on the cost function that is associated to the primal average generated by Algorithm 1:

$$f\left(\hat{\mathbf{u}}^{(k)}, x_t\right) \leq f_t'^{*} + \frac{\left\|\mu^{(0)}\right\|_2^2}{2k\tilde{\alpha}_t} + \frac{\tilde{\alpha}_t L_t'^2}{2} + \varepsilon_t \tag{31}$$

The proof for (31) and Proposition 2 can be found in [4]. This result leads to closed-loop stability, by using the cost function $f(\mathbf{u}_t, x_t)$ as a Lyapunov function.

The idea of constraint tightening is used often to ensure robustness of MPC, such as when designing robust distributed MPC for the case of decoupled systems [5, 10]. However, to the best of our knowledge it has not been applied for the case of coupled systems. In our proposed framework, we use constraint tightening to provide a feasible solution to the coupled constraints, rather than aiming to achieve an explicit robustness property. We believe that this framework can also be extended for robust hierarchical MPC where coupled constraints are present.

5 Availability of application results

In this section, we demonstrate an application of the proposed algorithm for a system of irrigation canals, where the objective is to regulate the water flows. Irrigation canals are large-scale systems, consisting of many interacting components, and spanning vast geographical areas. For the most efficient and safe operation of these canals, maintaining the water levels close to pre-specified reference values is crucial, both under normal operating conditions as well as in extreme situations. Manipulation of the water levels and flows in irrigation canals is typically 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. More details about modeling of this system can be found in [2].

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 the master controller [11]. We now apply the proposed hierarchical MPC method to design the master controllers.

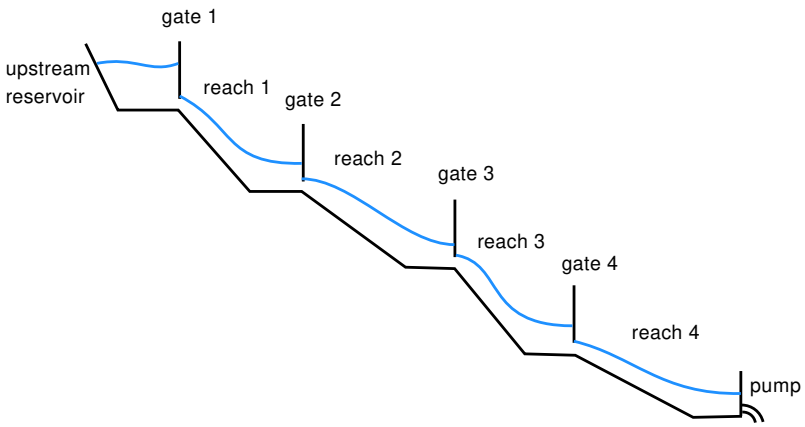

Fig. 1 Example of connected canals for water transportation

The Algorithm HPF-DAG is used in this example, and it generates at every time step t a feasible solution with respect to the physical constraints:

$$x_{\min} \leq x_{t+k} \leq x_{\max},\ k = 1,\ldots,N_{\mathrm{p}} \tag{32}$$

$$u_{\min} \leq u_{t+k} \leq u_{\max},\ k = 0,\ldots,N_{\mathrm{p}}-1 \tag{33}$$

The constraints described in (32) are then transformed into the inequality constraint $g(\mathbf{u}, x_t) \geq 0$ after the state variables are eliminated, and the constraints (33) are cast into the form $\mathbf{u} \in \tilde{\Omega}$ introduced earlier.

The closed-loop simulation is performed for 20 time steps. In each time step t, Algorithm 1 is used to generate a solution to the MPC optimization problem (3)–(9), with the number of outer loop iterations $\tilde{k}_t$ determined by (23), and the number of inner loop iterations $\tilde{p}_t$ determined by (24). Then the first control input in the solution generated by Algorithm 1 is used to simulate the system, and the routine restarts with the next time step. In every time step, we also solve the same optimization problem by a centralized solver to get the optimal solution and the optimal cost, for comparison purposes.

In Figure 2, we plot the evolutions of cost functions associated with different MPC solutions. At every time step, we compare the optimal cost, the cost associated with the initial feasible solution $\bar{\mathbf{u}}_t$, the cost generated by Algorithm 1 (i.e., associated with $\mathbf{u}_t$), and the upper bound of the resulting cost (see (31)). Although it is easy to obtain closed-loop stability with this example, the comparison of the cost nevertheless confirms that the upper bound of the cost is always respected by the result of the Algorithm HPF-DAG. Moreover, as t increases, the upper bound is closer to the optimal cost, thus the performance of the Algorithm HPF-DAG is also closer to the optimal centralized MPC performance.

In Figure 3, the number of outer loop iterations $\tilde{k}_t$ and the number of inner loop iterations $\tilde{p}_t$, are plotted for each sampling step. The corresponding CPU time required to run Algorithm 1 at each sampling step is shown in Figure 4. The CPU time is measured by implementing all the computations in one PC, running MATLAB on Windows with an Intel(R) Core(TM) i7 CPU at 2.30 GHz and with 8 GB RAM. We can see that the overall computation time for the simulation is long, since in every sampling period, there must be $\tilde{k}_t \times \tilde{p}_t \times M$ small optimization problems solved, where M is the number of subsystems (in this example $M = 4$). If the algorithm is simulated in a distributed setting, the computations would be divided between the M subsystems, thus the computation time would be much less than what is indicated by this simulation. More details of the simulation example can be found in [2].

6 Conclusions and discussion for extensions

In this chapter, we have presented a constraint tightening approach for solving MPC optimization problems involving large-scale linear systems with coupling in dynamics and constraints. This new approach provides guaranteed feasibility and stability after a finite number of iterations. The method is realized as a hierarchical algorithm, called HPF-DAG, in which the coordinator is in charge of determining the common parameters and the number of iterations, while the main computational tasks are divided up between the subsystems. We have provided the steps for implementation of this algorithm for the case of using a positive definite cost function and linear constraints. Future extensions of this approach can include the consideration of pos-

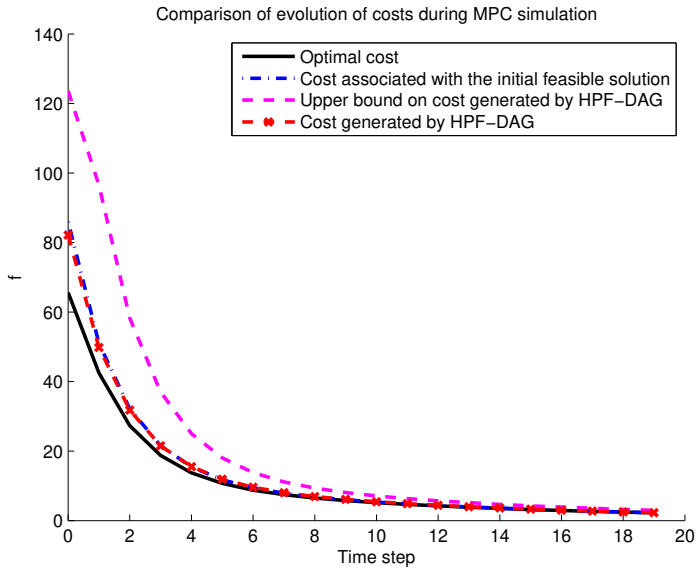

Fig. 2 Comparison of cost function evolutions in the simulation example.

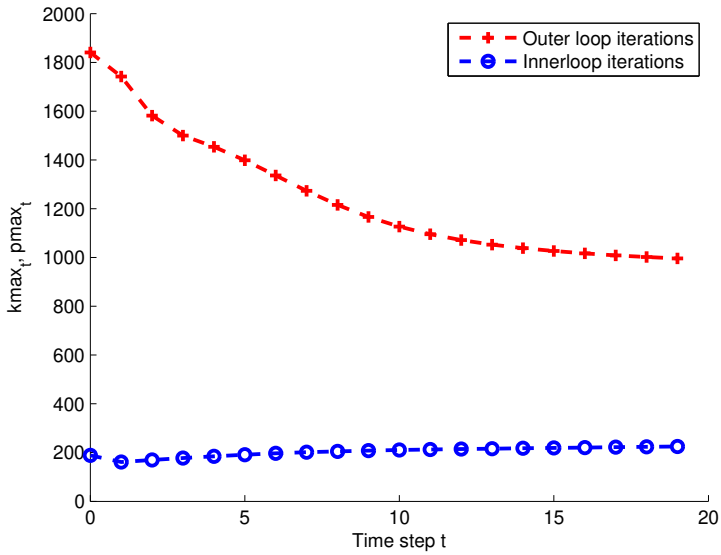

Fig. 3 Number of iterations in Algorithm HPF-DAG at each time step.

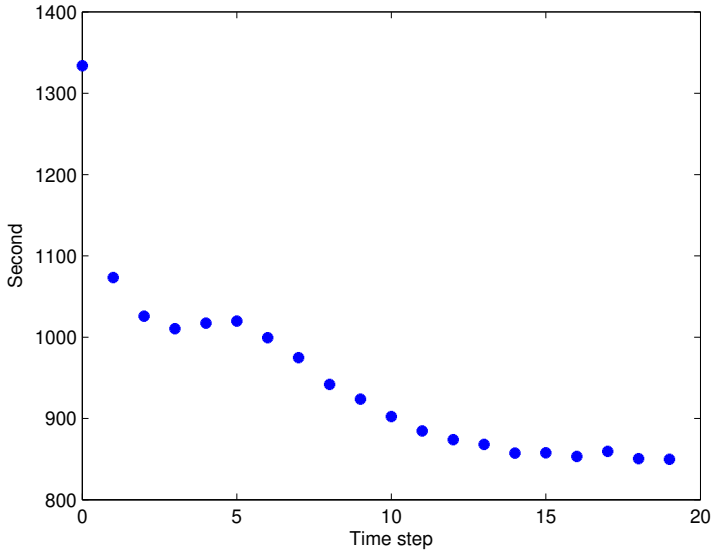

Fig. 4 Aggregate computation time over all subsystems for the simulation of Algorithm 1 at each time step.

itive semidefinite cost (i.e., some states and inputs are allowed to have zero penalty weights), and more general (nonlinear) convex constraints. These extensions would affect the convergence rate of the Jacobi algorithm used in the inner loop, hence the number of inner iterations would change, or in some cases we would need to replace the Jacobi algorithm with another, more suitable one. Other topics for extensions include the *a posteriori* choice of the solution between either the Slater vector $\bar{\mathbf{u}}_t$ or the primal average $\hat{\mathbf{u}}^{(\tilde{k}_t)}$ generated by the algorithm HPF-DAG; or determining optimal α_t, c_t, and ε_t parameters for fastest execution, while taking into account the communication effort.

Acknowledgements The work presented in this chapter has been supported by the European Union Seventh Framework STREP project *Hierarchical and Distributed Model Predictive Control (HD-MPC)* with contract number INFSO-ICT-223854.

References

1. D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.
2. M. D. Doan. *Distributed Model Predictive Controller Design Based on Distributed Optimization*. PhD thesis, Delft University of Technology, 2012.
3. M. D. Doan, T. Keviczky, and B. De Schutter. A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems. In *Proceedings of*

the 18th IFAC World Congress, pages 392–397, Milan, Italy, August–September 2011.
4. M.D. Doan, T. Keviczky, and B. De Schutter. A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints. In *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5236–5241, Orlando, Florida, December 2011.
5. Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007.
6. M. Mercangoz and F. J. Doyle III. Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17(3):297–308, March 2007.
7. P.-D. Morosan, R. Bourdais, D. Dumur, and J. Buisson. A distributed MPC strategy based on Benders' decomposition applied to multi-source multi-zone temperature regulation. *Journal of Process Control*, 21(5):729–737, 2011. Special Issue on Hierarchical and Distributed Model Predictive Control.
8. R. R. Negenborn, P. J. van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control for irrigation canals. *Networks and Heterogeneous Media*, 4(2):359–380, June 2009.
9. V. B. Peccin and E. Camponogara. Distributed model predictive control applied to urban traffic networks: Implementation, experimentation, and analysis. In *2010 IEEE Conference on Automation Science and Engineering*, pages 399–405, August 2010.
10. A.G. Richards and J.P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, September 2007.
11. J. Schuurmans, A. Hof, S. Dijkstra, O. H. Bosgra, and R. Brouwer. Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering*, 125(4):189–195, July 1999.
12. 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.