Delft University of Technology

Delft Center for Systems and Control

Technical report 11-022

A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems*

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 dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems," *Proceedings of the 18th IFAC World Congress*, Milan, Italy, pp. 392–397, Aug.–Sept. 2011.

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/11_022.html`

A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems

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: We present a gradient-based dual decomposition method that is suitable for hierarchical MPC of large-scale systems. The algorithm generates a primal feasible solution within a finite number of iterations and solves the problem by applying a hierarchical conjugate gradient method in each dual iterative ascent step. The proposed scheme uses constraint tightening and a suboptimality bound to ensure stability and feasibility in a hierarchical MPC problem.

Keywords: Hierarchical optimization, Dual decomposition, Model Predictive Control, Large-scale systems

1. INTRODUCTION

Control of large-scale industrial processes and infrastructure systems requires the coordination of interacting subsystems while striving for optimal operation that enforces critical operational constraints (Rawlings and Stewart, 2008). Due to its ability to handle important process constraints explicitly, Model Predictive Control (MPC) has become the method of choice when designing control systems for such applications (Maciejowski, 2002; Camacho and Bordons, 1999; Rawlings and Mayne, 2009). MPC relies on solving finite-time optimal control problems repeatedly online, which may become prohibitive for large-scale systems due to the problem size or communication constraints. Recent efforts have been focusing on how to decompose the underlying optimization problem in order to arrive at a distributed or hierarchical control system that can be implemented under the prescribed computational and communication limitations (Scattolini, 2009; Venkat et al., 2008). One common way to decompose an MPC problem with coupled dynamics or constraints is to use dual decomposition methods (Wakasa et al., 2008; Necoara and Suykens, 2008; Doan et al., 2009), which typically lead to iterative methods (in either a distributed or hierarchical framework) that converge to feasible solutions only asymptotically. Implementing such approaches within each MPC update period can be problematic for some applications.

In this paper we propose a hierarchical optimization approach for solving large-scale MPC problems with coupling in dynamics and constraints, which guarantees primal feasible solutions even after only a finite number of iterations. This is achieved by employing a combination of a primal averaging scheme, a hierarchical implementation of the conjugate gradient method, and constraint tightening. We indicate only the most important elements and considerations required for a standard MPC stability proof, and instead focus on the details of the proposed hierarchical optimization approach. The paper is thus organized as follows. In Section 2, we describe the MPC optimization problem and its tightened version, which will be used to guarantee feasibility of the original problem even with a suboptimal primal average solution. Section 3 describes a gradient algorithm to solve the dual version of the tightened optimization problem, and a hierarchical implementation of the conjugate gradient method that aids in the solution. In Section 4, we show that the primal average solution generated by the gradient algorithm is a feasible solution of the original optimization problem, and that the cost function will be decreasing through the MPC updates. This allows it to be used as a Lyapunov function for MPC stability. A numerical example is presented in Section 5 to demonstrate the new algorithm. Section 6 concludes the paper and outlines future research.

2. PROBLEM DESCRIPTION

2.1 MPC problem

In this section, we provide a brief summary of an MPC problem for large-scale systems. The intention is to introduce necessary assumptions and notation, and provide a brief motivation for the need of a hierarchical method that guarantees a feasible, stabilizing MPC solution.

We consider a system with the following discrete-time linear time-invariant state space model:

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

An MPC problem (Rawlings and Mayne, 2009) is formulated by repeatedly optimizing a cost function that consists of a stage cost $\ell(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k, k = t, \ldots, t+N-1$ and a terminal cost $x_{t+N}^T P x_{t+N}$, in which Q, R, and P are positive definite matrices. The constraints include linear dynamical constraints, operational constraints, and a terminal constraint which are represented by linear inequalities. Eliminating the state variables, the MPC op-

timization problem can be written in the following form (Venkat et al., 2007):

$$f_t^* = \min_{\mathbf{u}} \quad f(\mathbf{u}, x_t) \tag{2}$$
$$\text{s.t.} \quad g(\mathbf{u}, x_t) \leq 0 \tag{3}$$

where $\mathbf{u} = [u_t^T, \ldots, u_{t+N-1}^T]^T \in \mathbb{R}^{n_\mathbf{u}}$, x_t is the measured state at time step t, f and $g = [g_1, \ldots, g_m]^T$ are convex functions. We denote the optimal cost function value of (2)–(3) by f_t^*.

Remark 2.1. For large-scale systems, problem (2)–(3) is very high-dimensional, and it typically requires excessive communication and computations in order to be solved by a centralized controller. Thus, we intend to solve this problem with multiple local controllers that are organized in a hierarchical architecture to reduce the communication and computational tasks of each controller. The solution obtained from the hierarchical controllers may not be the optimal one, however it still has to be feasible and guarantee closed-loop MPC stability.

We denote the feasible solution of problem (2)–(3) at time step t as $\mathbf{u}_t$, that is chosen to be implemented but not necessarily optimal. Once it is determined, the first block u_t of the input vector $\mathbf{u}_t$ will be applied to the system.

Assumption 2.2. At each time step t, the following holds

$$f_t^* \leq f(\mathbf{u}_{t-1}, x_{t-1}) - \Delta_t \tag{4}$$

where a suitable $\Delta_t > 0$ can be computed before the calculation of $\mathbf{u}_t$.

Remark 2.3. In standard MPC literature, the preceding assumption is usually achieved by formulating the MPC problem with properly a chosen terminal cost, terminal constraint set and terminal controller. For example, choosing the terminal cost and the terminal controller by solving the Discrete-time Algebraic Riccati Equation will lead to $\Delta_t = x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1}$ (Mayne et al., 2000); or choosing the terminal controller as a decentralized stabilizing controller will allow computing Δ_t in a hierarchical way based on x_{t-1}, u_{t-1}, and the predicted state x_{t-1+N} that is computed at time step $t-1$. In these cases, Δ_t is the reduction of the cost from $f(\mathbf{u}_{t-1}, x_{t-1})$ to $f(\tilde{\mathbf{u}}_t, x_t)$ in time step t, using a feasible, stabilizing input sequence $\tilde{\mathbf{u}}_t$ constructed from $\mathbf{u}_{t-1}$.

Assumption 2.4. The Slater condition holds for problem (2)–(3) in each time step, i.e., there exists a Slater vector $\bar{\mathbf{u}}_t$ that satisfies the strict inequality constraints

$$g_j(\bar{\mathbf{u}}_t, x_t) < 0, j = 1, \ldots, m \tag{5}$$

The feasible solution $\bar{\mathbf{u}}_t$ will not be applied to the system, it will only be used to define several parameters that help solving the problem (2)–(3) in a hierarchical fashion.

Remark 2.5. In the literature of distributed and hierarchical optimization, problem (2)–(3) is often solved with dual decomposition methods in order to deal with the coupled constraints. However, such methods often result in iterations that converge to the optimal solution only asymptotically. This may also create difficulties in obtaining a corresponding primal feasible solution before convergence.

In the next section, we will present an algorithm that generates a primal feasible solution for (2)–(3) within a finite number of iteration steps. Moreover, it also provides a bound on suboptimality that could be used to show Lyapunov stability of MPC with decreasing cost function [1]. The core idea of our proposed approach is to use constraint tightening and then apply a projected gradient method for the dual problem. The algorithm is based on the results of Nedic and Ozdaglar (2009), which are used to calculate a primal feasible solution $\mathbf{u}_t$ for (2)–(3) at each time step t, and ensure a cost reduction $f(\mathbf{u}_t, x_t) < f(\mathbf{u}_{t-1}, x_{t-1})$ by making use of (4). A hierarchical implementation of the proposed approach is enabled by a suitable adaptation of the conjugate gradient method.

2.2 The tightened problem

One crucial aspect of our proposed method is that instead of solving the problem (2)–(3) directly, we will consider a tightened problem with the following constraint

$$g'(\mathbf{u}, x_t) \triangleq g(\mathbf{u}, x_t) + \mathbf{1}_m \epsilon_t \leq 0 \tag{6}$$

in which $0 < \epsilon_t < \min_{j=1,\ldots,m}\{-g_j(\bar{\mathbf{u}}_t, x_t)\}$, $g'(\mathbf{u}, x_t) = [g'_1, \ldots, g'_m]^T$, and $\mathbf{1}_m$ is the column vector with all entries equal to 1.

Using the modified constraint (6), we formulate the tightened problem

$$f_t'^* = \min_{\mathbf{u}} \quad f(\mathbf{u}, x_t) \tag{7}$$
$$\text{s.t.} \quad g'(\mathbf{u}, x_t) \leq 0 \tag{8}$$

Note that due to the dynamical and constraint couplings in the MPC formulation, problem (7)–(8) is a large-scale quadratic optimization problem

$$f(\mathbf{u}, x_t) = \frac{1}{2}\mathbf{u}^T H \mathbf{u} + b^T(x_t)\mathbf{u} \tag{9}$$
$$g'(\mathbf{u}, x_t) = C\mathbf{u} - d(x_t) + \mathbf{1}_m \epsilon_t \tag{10}$$

where H is a positive definite matrix, b and d are constant vectors depending on the initial state value, and both H and C have sparse structure resulting from the interconnection of subsystems.

It is straightforward to see that $g'_j(\bar{\mathbf{u}}_t, x_t) < 0, j = 1, \ldots, m$, hence $\bar{\mathbf{u}}_t$ is also a Slater vector for problem (7)–(8). We denote $\lambda_t = \min_{j=1,\ldots,m}\{-g'_j(\bar{\mathbf{u}}_t, x_t)\}$ for later reference.

Assumption 2.6. The difference between the tightened and original optimal cost are bounded above

$$f_t'^* - f_t^* \leq \phi_t \tag{11}$$

and ϕ_t can be computed and characterized such that

$$\phi_t < \Delta_t \tag{12}$$

Remark 2.7. The tightening cost ϕ_t can be approximated when there is a bound for the gradient $\nabla_\mathbf{u} f$ in the constraint set. In such case, based on the gradient bound and given Δ_t, we can choose the constraint tightening variable ϵ_t in (6) small enough such that (12) holds.

3. A HIERARCHICAL PRIMAL FEASIBLE DUAL GRADIENT ASCENT APPROACH

In this section, we focus on the solution of the optimization problem (7)–(8), which needs to be calculated in each

[1] A complete and rigorous description of a standard MPC stability proof falls outside the scope of this paper, readers may consult (Mayne et al., 2000) for details.

MPC update step. For simplicity of exposition, the dependence of functions on x_t will be omitted.

3.1 The dual problem

We consider solving the dual problem of (7)–(8), in order to deal with the coupled constraints $g'(\mathbf{u}) \leq 0$ in a hierarchical way.

The Lagrangian of problem (7)–(8) is defined as

$$\mathcal{L}(\mathbf{u}, \mu) = f(\mathbf{u}) + \mu^T g'(\mathbf{u}) \tag{13}$$

in which $\mu \in \mathbb{R}^m_+$ is called the dual variable.

The dual function of (7)–(8),

$$q'(\mu) = \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mu) \tag{14}$$

is a concave function. Note that if f and g' are continuous functions, and $\mathcal{L}(\mathbf{u}, \mu)$ is minimized over a unique point $\mathbf{u}(\mu)$, then according to (Bertsekas, 1999, Proposition 6.1.1), q' is differentiable everywhere and

$$\nabla q'(\mu) = g'(\mathbf{u}(\mu)), \quad \forall \mu \in \mathbb{R}^m_+ \tag{15}$$

Given the assumption that Slater condition holds for (7)–(8), duality theory (Bertsekas, 1999, Chapter 5) shows that

$$q_t'^* = f_t'^* \tag{16}$$

where $q_t'^* = \max_{\mu \in \mathbb{R}^m_+} q'(\mu)$ and $f_t'^*$ is the minimum of (7)–(8).

Thus, it is possible to compute an optimal solution by solving the dual problem, which is often easier to do than the primal one. In the following section, we will present a projected gradient method for maximizing the dual function.

3.2 Projected gradient method

The projected gradient iteration for solving (7)–(8) is given by

$$\mathbf{u}^{(k)} = \arg\min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mu^{(k)}) \tag{17}$$

$$\mu^{(k+1)} = \mathcal{P}_{\mathbb{R}^m_+} \left\{ \mu^{(k)} + \alpha_t g'^{(k)} \right\} \tag{18}$$

where k stands for the iteration index, the operator $\mathcal{P}_{\mathbb{R}^m_+}$ is the projection onto the nonnegative orthant, α_t is the constant step size (used for time step t), $\mu^{(k)}$ is the dual iterate at iteration k (for the first iteration, $\mu^{(0)} = 0 \cdot \mathbf{1}_m$), and $g'^{(k)} = g'\big(\mathbf{u}^{(k)}, x_t\big)$ is the gradient of the dual function $q'(\mu)$ at iteration k.

Assumption 3.1. The gradient sequence $\{g'^{(k)}\}$ is bounded, i.e., there exists a scalar L_t such that

$$\|g'^{(k)}\|_2 \leq L_t, \forall k \geq 0 \tag{19}$$

Note that when Slater condition holds, there is a bound for $\|\mu\|_2$ on the dual optimal set (Nedic and Ozdaglar, 2009, Lemma 1). Moreover, $\mathcal{L}(\mathbf{u}, \mu)$ is a convex quadratic function of $\mathbf{u}$, hence its minimizer $\mathbf{u}(\mu)$ is affinely dependent on μ, so is $g'(\mathbf{u}(\mu))$. Then, using inequality of matrix norm, we can derive such bound for $\|g'(\mathbf{u}(\mu))\|_2$.

The step size α_t for the dual update (18) is computed by:

$$\alpha_t = \frac{2(\Delta_t - \phi_t)}{L_t^2} \tag{20}$$

where Δ_t and ϕ_t are provided by (4) and (11), and L_t the norm bound for $g'^{(k)}$. This step size is chosen to facilitate showing the decreasing property of $f(\mathbf{u}_t, x_t)$ in Section 4.

The projected gradient iteration (17)–(18) is performed for $k = 1, \ldots, \bar{k}_t$, with $\bar{k}_t \in \mathbb{Z}_+$ defined a priori as

$$\bar{k}_t \geq \frac{1}{\alpha_t \epsilon_t} \left(\frac{3}{\lambda_t} f(\bar{\mathbf{u}}_t) + \frac{\alpha_t L_t^2}{2\lambda_t} + \alpha_t L_t \right) \tag{21}$$

The projected gradient method presented above generates a feasible solution for problem (2)–(3) by averaging the primal iterates:

$$\hat{\mathbf{u}}^{(\bar{k}_t)} = \frac{1}{\bar{k}_t} \sum_{l=0}^{\bar{k}_t} \mathbf{u}^{(l)} \tag{22}$$

This property will be shown in Section 4.

Remark 3.2. In order to implement the algorithm in a hierarchical fashion, we need hierarchical or distributed methods to solve problem (17) and perform (18). Even though (17) is indeed an unconstrained quadratic optimization problem, we will not use its analytical solution due to the computational burden when inverting the Hessian matrix. Instead, we will employ a conjugate gradient method (Bertsekas and Tsitsiklis, 1989, Chapter 2) and use a hierarchical implementation to find the solution of (17). The dual update (18) will also be done locally by letting each constraint be updated by a local controller.

In the next subsection, we describe a decomposition of the large-scale system, and present the hierarchical conjugate gradient method using the decomposition structure.

3.3 Subsystem decomposition

Recall that the functions f and g' have particular structure as described in (9)–(10). It is straightforward to verify (Venkat et al., 2007) that if matrices A and B are sparse (meaning that the large-scale system consists of subsystems with neighboring interactions), then H and C will also be sparse, and b and d will have a structured dependence on x_t. Now, consider a subsystem decomposition:

- Each subsystem $i = 1, \ldots, M$ has an associated decision variable $\mathbf{u}^i$ with the same dimension as $\mathbf{u}$, but containing only the variables corresponding to subsystem i in its nonzero entries [2]. We define $\mathbf{u}^i$ as
$$\mathbf{u}^i = \mathfrak{I}^i \mathbf{u} \tag{23}$$
where $\mathfrak{I}^i \in \mathbb{R}^{n_\mathbf{u} \times n_\mathbf{u}}$ is a diagonal matrix with zeros and ones on its diagonal. Matrices $\mathfrak{I}^i$ (and thus the subsystem decomposition) are chosen such that there is no overlap between the subsystems' variables.
- For each subsystem i, there is a neighborhood $\mathcal{N}^i$ that contains i itself and any other subsystem j that is coupled with i either via the objective function f (i.e., there is at least one term involving both variables of i and $j \in \mathcal{N}^i$ in f), or via the constraint function g' (i.e., there is at least one constraint that involves some variables of i and $j \in \mathcal{N}^i$).

In order to distribute the dual update (18), we will let each subsystem be in charge of updating a subset of dual

[2] Typically $\mathbf{u}^i$ contains the control inputs of subsystem i over the prediction horizon of the MPC problem.

variables, denoted by $\mathcal{D}^i$. There are different methods for partitioning of dual variables, among them one simple partitioning algorithm is the following: if the maximum absolute value of entries in a row r of C corresponds to a variable of subsystem i, then $r \in \mathcal{D}^i$. Note that each dual variable is updated by one and only one subsystem.

Since $g'(\mathbf{u}) = C\mathbf{u} - d$, we see that in order to perform update (18) for dual variables within $\mathcal{D}^i$, subsystem i will only need to communicate with subsystems $j \in \mathcal{N}^i$ during iteration k to get the necessary entries of $\mathbf{u}^{(k)}$.

3.4 Hierarchical conjugate gradient method

The algorithm we propose to use for solving (17) is an adaptation of the conjugate gradient method as described in (Bertsekas and Tsitsiklis, 1989, Chapter 2). Hereby we summarize the main steps and underlying ideas of this iterative method:

- The algorithm starts at some $\mathbf{u}(0)$ and select $s(0) = -\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(0), \mu^{(k)}) = -(H\mathbf{u}(0) + b(x_t) + C^T\mu^{(k)})$.
- The iteration has the form:
$$\mathbf{u}(p+1) = \mathbf{u}(p) + \gamma(p)s(p),\ p = 0, 1, \dots \tag{24}$$
with p the iteration index, $s(p)$ the direction of update at iteration p, and $\gamma(p)$ an optimal scalar step size.
- The algorithm stops if $\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p), \mu^{(k)}) = 0$. Otherwise, update $s(p)$ by
$$s(p) = -\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p), \mu^{(k)}) + \beta(p)s(p-1) \tag{25}$$
where $\beta(p)$ is generated by
$$\beta(p) = \frac{\nabla^T_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p), \mu^{(k)})\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p), \mu^{(k)})}{\nabla^T_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p-1), \mu^{(k)})\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p-1), \mu^{(k)})} \tag{26}$$
- We update $\gamma(p)$ by
$$\gamma(p) = -\frac{s(p)^T\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}(p), \mu^{(k)})}{s(p)^T H s(p)} \tag{27}$$
- One feature of this iteration method is the conjugate property of $s(p)$, such that
$$s(p)^T H s(r) = 0, \forall r \neq p \tag{28}$$
- The algorithm terminates after at most $n_{\mathbf{u}}$ steps[3], where $n_{\mathbf{u}}$ is the size of $\mathbf{u}$.

For application in hierarchical MPC, it is required that the communications and computations of (24), (25), (26), (27), (18) and (22) can be done in a hierarchical setting. We propose to use a hierarchical optimization method, in which a coordinator communicates with all local controllers, and each local controller can also communicate with others in its neighborhood. In summary, we propose a nested iterative algorithm in which the outer loop is the projected gradient method for the dual problem (17)–(18), and the inner loop is the hierarchical conjugate gradient method for solving (17).

Algorithm 1. **Hierarchical Primal Feasible Dual Gradient (H-PF-DG) method**

(1) Input: $\bar{\mathbf{u}}_t, \delta_t, L_t$. The coordinator computes $\epsilon_t, \lambda_t, \alpha_t$, and $\bar{k}_t$, then sends ϵ_t and $\bar{k}_t$ to all local controllers.

(2) Set $k = 0$. Choose $\mu^{(0)} = 0 \cdot \mathbf{1}_m$, $\mathbf{u}^{(0)} = 0 \cdot \mathbf{1}_{n_{\mathbf{u}}}$, then each local controller has ${\mathbf{u}^{(0)}}^i = 0 \cdot \mathbf{1}_{n_{\mathbf{u}}}$.

(3) Solve (17) at step k by the following iterative process:

(a) Set $p = 0$. Initialize each local controller $i \in \{1, \dots, M\}$ with $\mathbf{u}(0)^i = {\mathbf{u}^{(k)}}^i$.

(b) Each local controller $i \in \{1, \dots, M\}$ communicates with $j \in \mathcal{N}^i$ to get $\mathbf{u}(p)^j$, then computes
$$\nabla\mathcal{L}(\mathbf{u}(p))^i = \mathfrak{I}^i\big(H\sum_{j\in\mathcal{N}^i}\mathbf{u}^j(p) + b + C^T\mu^{(k)}\big)$$

(c) Each local controller $i \in \{1, \dots, M\}$ computes $\nabla\mathcal{L}^T(\mathbf{u}(p))^i\nabla\mathcal{L}(\mathbf{u}(p))^i$, and then sends the result to the coordinator.

(d) The coordinator makes the sum:
$$\nabla\mathcal{L}^T(\mathbf{u}(p))\nabla\mathcal{L}(\mathbf{u}(p)) = \sum_{i=1}^{M}\nabla\mathcal{L}^T(\mathbf{u}(p))^i\nabla\mathcal{L}(\mathbf{u}(p))^i$$
Note that steps 3(b), 3(c), and 3(d) are aimed at updating $\nabla\mathcal{L}^T(\mathbf{u}(p))\nabla\mathcal{L}(\mathbf{u}(p))$, which appears in (26), by computing its subsystem components.

(e) The coordinator checks whether $\nabla\mathcal{L}(\mathbf{u}(p)) = 0$. If so, then it announces "stop" and each controller takes ${\mathbf{u}^{(k)}}^i = \mathbf{u}(p)^i$. Go to step (4). If $\nabla\mathcal{L}(\mathbf{u}(p)) \neq 0$, the coordinator computes $\beta(p)$ by (26), and sends $\beta(p)$ to all local controllers.

(f) Each local controller $i \in \{1, \dots, M\}$ computes $s(p)^i = -\nabla\mathcal{L}(\mathbf{u}(p))^i + \beta(p)s(p-1)^i$ if $p > 0$, or $s(p)^i = -\nabla\mathcal{L}(\mathbf{u}(p))^i$ if $p = 0$, and then communicates with $j \in \mathcal{N}^i$ to get $s(p)^j$. The purpose of this step is to use local implementation for computing (25).

(g) Each local controller $i \in \{1, \dots, M\}$ computes $[s(p)^i]^T\nabla\mathcal{L}(\mathbf{u}(p))^i$ and $[s(p)^i]^T H\sum_{j\in\mathcal{N}^i}s(p)^j$, and then sends these results to the coordinator.

(h) The coordinator makes the sums:
$$s(p)^T\nabla\mathcal{L}(\mathbf{u}(p)) = \sum_{i=1}^{M}[s(p)^i]^T\nabla\mathcal{L}(\mathbf{u}(p))^i$$
$$s(p)^T s(p) = \sum_{i=1}^{M}\left\{[s(p)^i]^T H\sum_{j\in\mathcal{N}^i}s(p)^j\right\}$$
and computes $\gamma(p)$ according to (27), then sends $\gamma(p)$ to all local controllers.

(i) Each local controller $i \in \{1, \dots, M\}$ updates $\mathbf{u}(p+1)^i = \mathbf{u}(p)^i + \gamma(p)s(p)^i$.

(j) Set $p = p + 1$, go to step 3(b).

(4) Each local controller $i \in \{1, \dots, M\}$ communicates with $j \in \mathcal{N}^i$ to get ${\mathbf{u}^{(k)}}^j$.

(5) Each local controller $i \in \{1, \dots, M\}$ updates the dual variables in the set $\mathcal{D}^i$ by:
$$\mu^{(k+1)}(l) = \mathcal{P}_{\mathbb{R}_+}\Big\{\mu^{(k)}(l) + \alpha_t\Big(\sum_{v=1}^{n_{\mathbf{u}}}C(l,v)\mathbf{u}^{(k)}(v) - d(l)\Big)\Big\}, \forall l \in \mathcal{D}^i$$
where all $C(l, v)$ that are nonzero correspond to variables v of the subsystems $j \in \mathcal{N}^i$, and therefore the knowledge of ${\mathbf{u}^{(k)}}^j, j \in \mathcal{N}^i$ is enough for this computation.

(6) Set $k = k + 1$. If $k \leq \bar{k}_t$, go to step (3).

[3] The number of iterations can be significantly reduced with proper preconditioning.

(7) Each local controller $i \in \{1,\dots,M\}$ computes $\hat{\mathbf{u}}^i = \frac{1}{\bar{k}_t}\sum_{k=0}^{\bar{k}_t} \mathbf{u}^{(k)^i}$. The corresponding global vector $\hat{\mathbf{u}}^{(\bar{k}_t)} = \sum_{i=1}^{M} \hat{\mathbf{u}}^i$ will be a feasible solution of (2)–(3).

Remark 3.3. Algorithm 1 needs a coordinator to compute and deliver common variables to the local controllers. However, most of the computations are carried out by local controllers. Each local controller needs to exchange information with its neighbors and the coordinator. Regarding communications, major communications between subsystems are in the order of $2\bar{k}_t \times n_{\mathbf{u}} \times \sum_{i=1}^{M} |\mathcal{N}^i|$ messages, while the communications between the coordinator and all local controllers are in the order of $4\bar{k}_t \times n_{\mathbf{u}} \times M$ messages.

In the next section, we will show that $\hat{\mathbf{u}}^{(\bar{k}_t)}$ generated by Algorithm 1 is a feasible solution of (2)–(3), and ensures cost reduction for the MPC problem.

4. PROPERTIES OF THE H-PF-DG ALGORITHM

Denoting the primal average sequence by $\hat{\mathbf{u}}^{(k)} = \frac{1}{k}\sum_{l=0}^{k} \mathbf{u}^{(l)}$ where $\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)}$ are generated by (17), we have the following result (Nedic and Ozdaglar, 2009) for $k \geq 1$:

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(k)}\right) \right]^+ \right\|_2 \leq \frac{1}{k\alpha_t}\left(\frac{3}{\lambda_t}[f(\bar{\mathbf{u}}_t) - q_t'^*] + \frac{\alpha_t L_t^2}{2\lambda_t} + \alpha_t L_t \right) \tag{29}$$

$$f\left(\hat{\mathbf{u}}^{(k)}\right) \leq f_t'^* + \frac{\left\|\mu^{(0)}\right\|_2^2}{2k\alpha_t} + \frac{\alpha_t L_t^2}{2} \tag{30}$$

where the notation g'^+ indicates the constraint violation, i.e., $g'^+ = \max\{g', 0 \cdot \mathbf{1}_m\}$. Using the constraint violation bound (29) and the cost upper bound (30) for the tightened problem (7)–(8), we will show that $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of (2)–(3), and $f(\mathbf{u}_t, x_t) \leq f(\mathbf{u}_{t-1}, x_{t-1})$.

4.1 Primal feasible solution

Proposition 4.1. Let Assumption 2.4 hold and Algorithm 1 be executed. The primal average $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of (2)–(3).

Proof: Applying the result in (29) leads to

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) \right]^+ \right\|_2 \leq \frac{1}{\bar{k}_t\alpha_t}\left(\frac{3}{\lambda_t}\left[f(\bar{\mathbf{u}}_t) - q_t'^*\right] + \frac{\alpha_t L_t^2}{2\lambda_t} + \alpha_t L_t \right)$$

Moreover, $q_t'^* = f_t'^* \geq 0$ because $f(\mathbf{u}) \geq 0, \forall \mathbf{u}$ due to the use of a quadratic stage cost in the MPC setting. We have

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) \right]^+ \right\|_2 \leq \frac{1}{\bar{k}_t\alpha_t}\left(\frac{3}{\lambda_t} f(\bar{\mathbf{u}}_t) + \frac{\alpha_t L_t^2}{2\lambda_t} + \alpha_t L_t \right) \tag{31}$$

Combining (31) and (21), and noticing that $\bar{k}_t$ and ϵ_t are both positive lead to

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) \right]^+ \right\|_2 \leq \epsilon_t \tag{32}$$

$$\Rightarrow g_j'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) \leq \epsilon_t, \quad j = 1,\dots,m \tag{33}$$

$$\Rightarrow g_j\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) < 0, \quad j = 1,\dots,m \tag{34}$$

in which the last inequality is due to (6). This means that $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of problem (2)–(3). □

4.2 Decreasing cost function

Let us recall that the optimization problem formulation is motivated by an MPC problem for which Assumption 2.2 holds. The following proposition shows that the cost function of the MPC problem is a decreasing function.

Proposition 4.2. Let $\mathbf{u}_{t-1}$ and x_{t-1} be given and satisfy (4). Considering $\mathbf{u}_t = \hat{\mathbf{u}}^{(\bar{k}_t)}$ generated by Algorithm 1, the following inequality holds

$$f(\mathbf{u}_t, x_t) \leq f(\mathbf{u}_{t-1}, x_{t-1}) \tag{35}$$

Proof: Using (30), (20), and noting that $\mu^{(0)} = 0$ lead to

$$f(\mathbf{u}_t, x_t) \triangleq f\left(\hat{\mathbf{u}}^{(\bar{k}_t)}\right) \leq f_t'^* + \frac{\alpha_t L_t^2}{2} \tag{36}$$

Due to (20), we then have

$$f(\mathbf{u}_t, x_t) \leq f_t'^* - \phi_t + \Delta_t \tag{37}$$

Combining (37), (11), and (4) results in the cost decrease property: $f(\mathbf{u}_t, x_t) \leq f(\mathbf{u}_{t-1}, x_{t-1})$. □

In summary, the algorithm H-PF-DG is able to generate a feasible solution for the MPC optimization problem within each time step t, which is used to show that the cost function is decreasing if Assumption 2.2 holds. In order to employ this algorithm for hierarchical MPC, one needs to make sure that a feasible prediction $\bar{\mathbf{u}}_t$ of the input sequence, and a cost reduction δ_t are available before each time step t.

In the next section, we illustrate Algorithm 1 and its properties in a numerical optimization example.

5. NUMERICAL EXAMPLE

In this section, Algorithm 1 is applied to an optimization problem of the form

$$\min_{\mathbf{u}} \quad \frac{1}{2}\mathbf{u}^T H \mathbf{u} + b^T \mathbf{u} \tag{38}$$

$$\text{s.t.} \quad C\mathbf{u} - d \leq 0 \tag{39}$$

where

$$H = \begin{bmatrix} 4 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \; b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; C = \left[\, C_1^T \middle| C_2^T \middle| C_3^T \middle| C_4^T \, \right]^T,$$

$$d = [1, 1, 2, 2, 4, -1, -3, 4, 0, -2, -1, 4]^T$$

with

$$C_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad C_4 = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

The coupling structure in H and C leads to the following subsystem decomposition:

$$\mathcal{N}^1 = \{1, 2\}, \qquad \mathcal{D}^1 = \{1, 2, 3\},$$
$$\mathcal{N}^2 = \{2, 1, 3\}, \qquad \mathcal{D}^2 = \{4, 5, 6\},$$
$$\mathcal{N}^3 = \{3, 2, 4\}, \qquad \mathcal{D}^3 = \{7, 8, 9\},$$
$$\mathcal{N}^4 = \{4, 3\}, \qquad \mathcal{D}^4 = \{10, 11, 12\}$$

where $\mathcal{N}^i$ is the subsystem neighborhood and $\mathcal{D}^i$ is the subset of dual variables associated with each subsystem i.

We used the following parameters to initialize the H-PF-DG algorithm:

$$\bar{\mathbf{u}} = [0, 2, 2, -2]^T, \quad \Delta = 5, \quad \phi = 2, \quad L = 112.$$

where Δ and ϕ were chosen based on conservative approximations.

We remark that the formula of the step size given in (20) is a conservative sufficient condition to ensure Propositions 4.1 and 4.2. In our numerical example, we simulated Algorithm 1 with a larger step size ($\alpha = 0.16$) and observed that primal feasibility and cost decrease were achieved with only a fraction of the number of iterations that would suffice based on our theoretical results. The simulation takes 7.36 s with MATLAB R2008b for Linux, running on an Intel(R) Core(TM)2 Duo CPU E6550 at 2.33 GHz and 2GB RAM. The evolution of the cost function associated with the primal average sequence $\hat{\mathbf{u}}^{(k)}$ is plotted in Figure 1. In the first few iterations of the algorithm $\hat{\mathbf{u}}^{(k)}$ is infeasible, which explains why the cost is lower than the optimal value. We observed that $\hat{\mathbf{u}}^{(k)}$ satisfies the constraints for all iterations $k \geq 26$. Hence, a solution that fulfills all our requirements (i.e., primal feasible and leads to MPC cost decrement) is obtained only after 26 iterations, which is much smaller than $\bar{k} = 13155$ as suggested by the conservative lower bound in (21).

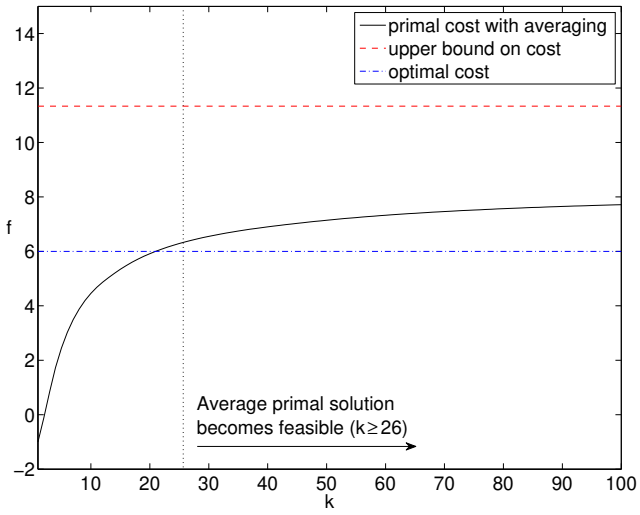

Fig. 1. Evolution of the cost function value with primal averaging.

6. CONCLUSIONS AND FUTURE WORK

We have presented a dual gradient-based hierarchical method for solving a large quadratic optimization problem. The iterative algorithm can be terminated after a finite number of iterations and provides a feasible primal solution. In connection with hierarchical MPC, this algorithm also helps to show the decreasing property of the cost function, which can then be used, with further assumptions, as a Lyapunov function for proving MPC stability. For future research, we intend to apply the method in hierarchical MPC applications, analyze the convergence speed of the algorithm and develop less conservative bounds on the number of iterations. We plan to extend the framework to a completely distributed implementation using dual decomposition methods and approximate subgradients.

ACKNOWLEDGEMENTS

Research supported by the European 7th framework STREP project "Hierarchical and distributed model predictive control (HD-MPC)", contract number INFSO-ICT-223854, and the European 7th Framework Network of Excellence "Highly-complex and networked control systems (HYCON2)".

REFERENCES

Bertsekas, D.P. (1999). *Nonlinear programming*. Athena Scientific, Belmont, MA.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and distributed computation: Numerical Methods*. Prentice-Hall, Upper Saddle River, NJ.

Camacho, E.F. and Bordons, C. (1999). *Model Predictive Control*. Springer, London.

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.

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

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

Necoara, I. and Suykens, J. (2008). Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control*, 53(11), 2674–2679.

Nedic, A. and Ozdaglar, A. (2009). Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4), 1757–1780.

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

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.

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

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.

Venkat, A., Rawlings, J., and Wright, S. (2007). Distributed model predictive control of large-scale systems. In R. Findeisen, F. Allgwer, and L. Biegler (eds.), *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, 591–605. Springer Berlin / Heidelberg.

Wakasa, Y., Arakawa, M., Tanaka, K., and Akashi, T. (2008). Decentralized model predictive control via dual decomposition. In *47th IEEE Conference on Decision and Control*, 381–386.