Distributed Optimization for Predictive Control of a Distillation Column with Output and Control-Input Constraints ⋆

Helton F. Scherer * **Eduardo Camponogara** * **Agustinho Plucenio** *

* *Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, SC 88040-900 Brazil (e-mails: scherer@das.ufsc.br, camponog@das.ufsc.br, plucenio@das.ufsc.br)*

Abstract: A distributed predictive control framework based on a state-space model with constraints on the output and control-input is proposed. By a benchmark process, the performance of this framework is analyzed and compared with centralized control strategies in a regulation problem of a distillation column.

Keywords: Model predictive control, distributed optimization, distillation column, convex optimization, interior-point method.

1. INTRODUCTION

Advanced techniques for multivariable control like model predictive control (MPC) have become widespread in the industry, but they are still complex, time-consuming to set-up, and consequently expensive. Further, the centralized approach may not be suitable to the operation of large dynamic networks, either by the communication difficulty between sensors and the central unit, or by the computational limitation to solve optimization problems. Some petrochemical plants are examples of large systems composed by distributed, however coupled subsystems.

An alternative is distributed predictive control (Camponogara et al., 2002), which breaks the static optimization problem into smaller sub-problems to be solved by a network of control/optimization agents. It aims to solve the sub-problems in the most simple form while the final performance is preserved or even improved.

Many studies about distributed formulations are being developed. Mercangöz and Doyle III (2007) propose a distributed formulation that ensures self-sufficient state estimation in each node. Motee and Jadbabaie (2006) present a study of receding horizon control applied to physically decoupled systems with input and state constraints, where the couplings appear through the finite horizon cost function. Li et al. (2005) and Giovanini and Balderud (2006) propose MPC strategies based on Nash optimality to decoupled sub-systems.

Besides this, many algorithms to ensure convergence of distributed problems are being proposed. Dunbar (2007) presents distributed algorithms for dynamically coupled nonlinear systems subject to decoupled input constraints. An iterative procedure based on cooperation that ensures convergence to the global optimum for linear systems with constraints on the local controls is presented by Venkat et al. (2008). Camponogara and Talukdar (2007) present synchronous and asynchronous solutions of optimization problems, proposing a high level optimization framework and safety margins for meeting constraints. Recently, distributed predictive control was specialized to linear dynamic networks and applied to traffic light control, in which the dynamics of the sub-systems are coupled and the constraints are on the local controls (Camponogara and de Oliveira, 2009).

This paper proposes a problem decomposition and distributed algorithm for predictive control of linear networks with dynamic couplings and restrictions on output and control-input signals. Further, it reports on a comparison with existing approaches for the control of a distillation column.

The end result of this research is a distributed predictive control technique for programming control agents which can be deployed to perform regulatory control of linear dynamic networks. Each agent would be responsible for solving a problem of control action, exchanging its local sensor data and control actions with the other agents. The resulting control action is obtained after resolving conflicts with neighboring agents.

Because the algorithms embedded in the agents are much simpler than the centralized one, the distributed approach makes it simpler to modify and reconfigure the plant. Instead of modifying the complex centralized algorithm, it would suffice to add new agents and update only the nearby agents with whom the new agents would have relation. Further, maintenance would be facilitated since the distributed agents are much simpler.

2. DYNAMIC MODEL AND CONTROL PROBLEM

A linear dynamic network is obtained by interconnecting decoupled sub-systems that have local dynamics and controls. The couplings arise from the dynamic interconnections and the constraints on the network's output equations. $\mathscr{M} = \{1, \ldots, M\}$ denotes the set of sub-systems. Each sub-system m is governed by the following discrete-time linear dynamic equation:

$$\mathbf{x}_m(k+1) = A_m \mathbf{x}_m(k) + B_m \mathbf{u}_m(k) \quad (1)$$

⋆ This research has been funded in part by Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) under grant PRH-34/aciPG/ANP and by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under grant 473841/2007-0.

where $\mathbf{x}_m \in \mathbb{R}^{n_m}$ is the state, $\mathbf{u}_m \in \mathbb{R}^{p_m}$ is the control input, and A_m and B_m are matrices of appropriate dimensions. The output from sub-system m depends on the state of the sub-systems in set $I(m) \subseteq \mathscr{M}$ such that $m \in I(m)$:

$$\mathbf{y}_m(k) = \sum_{i \in I(m)} C_{m,i}\mathbf{x}_i(k) \tag{2}$$

and is subject to the output constraints:

$$\mathbf{y}_m^{\min} \leq \mathbf{y}_m(k) \leq \mathbf{y}_m^{\max} \tag{3}$$

with $C_{m,i}$ being matrices of suitable dimensions.

The regulation problem for the overall system subject to output and control-input constraints is:

$$\min \frac{1}{2}\sum_{m=1}^{M}\sum_{k=0}^{\infty}\left[\mathbf{y}_m(k+1)'Q_m\mathbf{y}_m(k+1)+\mathbf{u}_m(k)'R_m\mathbf{u}_m(k)\right] \tag{4a}$$

$$\text{S.to : For } m = 1,\ldots,M,\ k = 0,\ldots,\infty:$$

$$\mathbf{x}_m(k+1) = A_m\mathbf{x}_m(k) + B_m\mathbf{u}_m(k) \tag{4b}$$

$$\mathbf{y}_m(k+1) = \sum_{i \in I(m)} C_{m,i}\mathbf{x}_i(k+1) \tag{4c}$$

$$\mathbf{u}_m^{\min} \leq \mathbf{u}_m(k) \leq \mathbf{u}_m^{\max} \tag{4d}$$

$$\mathbf{y}_m^{\min} \leq \mathbf{y}_m(k+1) \leq \mathbf{y}_m^{\max} \tag{4e}$$

where Q_m are symmetric positive semi-definite and R_m are symmetric positive definite matrices.

Model predictive control solves an optimization problem that approximates the regulation problem for a finite-time horizon. Given the state $\mathbf{x}(k) = (\mathbf{x}_1,\ldots,\mathbf{x}_M)(k)$ of the system at time k, the MPC regulation problem is defined as:

$$P: \min f = \frac{1}{2}\sum_{m=1}^{M}\sum_{j=0}^{T-1}\left[\hat{\mathbf{y}}_m(k+j+1|k)'Q_m\hat{\mathbf{y}}_m(k+j+1|k) + \hat{\mathbf{u}}_m(k+j|k)'R_m\hat{\mathbf{u}}_m(k+j|k)\right] \tag{5a}$$

$$\text{S.to : For } m = 1,\ldots,M,\ j = 0,\ldots,T-1:$$

$$\hat{\mathbf{x}}_m(k+j+1|k) = A_m\hat{\mathbf{x}}_m(k+j|k) + B_m\hat{\mathbf{u}}_m(k+j|k) \tag{5b}$$

$$\hat{\mathbf{y}}_m(k+j+1|k) = \sum_{i \in I(m)} C_{m,i}\hat{\mathbf{x}}_i(k+j+1|k) \tag{5c}$$

$$\mathbf{u}_m^{\min} \leq \hat{\mathbf{u}}_m(k+j|k) \leq \mathbf{u}_m^{\max} \tag{5d}$$

$$\mathbf{y}_m^{\min} \leq \hat{\mathbf{y}}_m(k+j+1|k) \leq \mathbf{y}_m^{\max} \tag{5e}$$

$$\hat{\mathbf{x}}_m(k|k) = \mathbf{x}_m(k) \tag{5f}$$

where $\hat{\mathbf{u}}_m(k+j|k)$ is the prediction for the control input to sub-system m at time $(k+j)$ as predicted at time k, and similarly $\hat{\mathbf{y}}_m$ and $\hat{\mathbf{x}}_m$ are output and state predictions respectively. The variable T is the length of the prediction and control horizons, that have the same length to make the developments simpler.

The term "$|k$" is dropped from all variables for the sake of simplification. Before continuing with the MPC formulation, some terminology will be introduced to simplify the representation. First, it is possible to obtain the state of sub-system m at time $(k+t)$ by using the initial state and the past controls. The future states and outputs are represented as:

$$\hat{\mathbf{x}}_m(k+t) = A_m^t\mathbf{x}_m(k) + \sum_{j=1}^{t} A_m^{j-1}B_m\hat{\mathbf{u}}_m(k+t-j)$$

$$\hat{\mathbf{y}}_m(k+t) = \sum_{i \in I(m)} C_{m,i}\left(A_i^t\mathbf{x}_i(k) + \sum_{j=1}^{t} A_i^{j-1}B_i\hat{\mathbf{u}}_i(k+t-j)\right)$$

By defining the vectors $\hat{\mathbf{x}}_m$, $\hat{\mathbf{u}}_m$, and $\hat{\mathbf{y}}_m$ to represent the predictions over the entire horizon of the states, controls, and outputs, respectively, and the matrices $\overline{CA}_{m,i}$ and $\overline{CB}_{m,i}$ for the dynamics:

$$\hat{\mathbf{x}}_m = \begin{bmatrix}\hat{\mathbf{x}}_m(k+1)\\ \vdots \\ \hat{\mathbf{x}}_m(k+T)\end{bmatrix}, \quad \hat{\mathbf{u}}_m = \begin{bmatrix}\hat{\mathbf{u}}_m(k)\\ \vdots \\ \hat{\mathbf{u}}_m(k+T-1)\end{bmatrix}$$

$$\hat{\mathbf{y}}_m = \begin{bmatrix}\hat{\mathbf{y}}_m(k+1)\\ \vdots \\ \hat{\mathbf{y}}_m(k+T)\end{bmatrix}, \quad \overline{CA}_{m,i} = \begin{bmatrix}C_{m,i}A_i\\ C_{m,i}A_i^2\\ \vdots \\ C_{m,i}A_i^T\end{bmatrix}$$

$$\overline{CB}_{m,i} = \begin{bmatrix} C_{m,i}B_i & 0 & \cdots & 0 \\ C_{m,i}A_iB_i & C_{m,i}B_i & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ C_{m,i}A_i^{T-1}B_i & C_{m,i}A_i^{T-2}B_i & \cdots & C_{m,i}B_i \end{bmatrix}$$

the prediction of the outputs over the entire horizon is written in a compact form as:

$$\hat{\mathbf{y}}_m = \sum_{i \in I(m)}\left(\overline{CA}_{m,i}\mathbf{x}_i(k) + \overline{CB}_{m,i}\hat{\mathbf{u}}_i\right) \tag{6}$$

Defining the matrices $\bar{Q}_m$ and $\bar{R}_m$ with proper dimensions:

$$\bar{Q}_m = \begin{bmatrix} Q_m & 0 & \cdots & 0 \\ 0 & Q_m & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & Q_m \end{bmatrix}, \quad \bar{R}_m = \begin{bmatrix} R_m & 0 & \cdots & 0 \\ 0 & R_m & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & R_m \end{bmatrix}$$

and the vectors $\hat{\mathbf{u}}_m^{\min}$, $\hat{\mathbf{u}}_m^{\max}$, $\hat{\mathbf{y}}_m^{\min}$, and $\hat{\mathbf{y}}_m^{\max}$:

$$\hat{\mathbf{u}}_m^{\min} = \begin{bmatrix}\mathbf{u}_m^{\min}\\ \vdots \\ \mathbf{u}_m^{\min}\end{bmatrix}, \quad \hat{\mathbf{u}}_m^{\max} = \begin{bmatrix}\mathbf{u}_m^{\max}\\ \vdots \\ \mathbf{u}_m^{\max}\end{bmatrix}$$

$$\hat{\mathbf{y}}_m^{\min} = \begin{bmatrix}\mathbf{y}_m^{\min}\\ \vdots \\ \mathbf{y}_m^{\min}\end{bmatrix}, \quad \hat{\mathbf{y}}_m^{\max} = \begin{bmatrix}\mathbf{y}_m^{\max}\\ \vdots \\ \mathbf{y}_m^{\max}\end{bmatrix}$$

problem P is expressed as:

$$P: \min \frac{1}{2}\sum_{m=1}^{M}\left(\hat{\mathbf{y}}_m'\bar{Q}_m\hat{\mathbf{y}}_m + \hat{\mathbf{u}}_m'\bar{R}_m\hat{\mathbf{u}}_m\right) \tag{7a}$$

$$\text{S.to : For } m = 1,\ldots,M:$$

$$\hat{\mathbf{y}}_m = \sum_{i \in I(m)}\left(\overline{CA}_{m,i}\mathbf{x}_i(k) + \overline{CB}_{m,i}\hat{\mathbf{u}}_i\right) \tag{7b}$$

$$\hat{\mathbf{u}}_m^{\min} \leq \hat{\mathbf{u}}_m \leq \hat{\mathbf{u}}_m^{\max} \tag{7c}$$

$$\hat{\mathbf{y}}_m^{\min} \leq \hat{\mathbf{y}}_m \leq \hat{\mathbf{y}}_m^{\max} \tag{7d}$$

The next step is to represent problem P using only the current state and the control predictions. First, let us define f_m as the portion of problem P for a specific m and replace (6) in the objective function:

$$
\begin{aligned}
f_m =& \frac{1}{2}\left(\sum_{i\in I(m)}\left(\overline{CA}_{m,i}\mathbf{x}_i(k)+\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right)\right)'\bar{Q}_m \\
&\cdot\left(\sum_{i\in I(m)}\left(\overline{CA}_{m,i}\mathbf{x}_i(k)+\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right)\right)+\frac{1}{2}\hat{\mathbf{u}}_m'\bar{R}_m\hat{\mathbf{u}}_m \\
=& \frac{1}{2}\left(\sum_{i\in I(m)}\overline{CA}_{m,i}\mathbf{x}_i(k)\right)'\bar{Q}_m\left(\sum_{i\in I(m)}\overline{CA}_{m,i}\mathbf{x}_i(k)\right) \\
&+\left(\sum_{i\in I(m)}\overline{CA}_{m,i}\mathbf{x}_i(k)\right)'\bar{Q}_m\left(\sum_{i\in I(m)}\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right) \\
&+\frac{1}{2}\left(\sum_{i\in I(m)}\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right)'\bar{Q}_m\left(\sum_{i\in I(m)}\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right) \\
&+\frac{1}{2}\hat{\mathbf{u}}_m'\bar{R}_m\hat{\mathbf{u}}_m
\end{aligned}
\tag{8}
$$

Defining vectors $\mathbf{g}_{m,i,j}$, matrices $H_{m,i,j}$, and a constant c_m to represent the terms of f_m:

$$\mathbf{g}_{m,i,j} = \overline{CB}_{m,i}'\bar{Q}_m\overline{CA}_{m,j}\mathbf{x}_j(k) \text{ for } i,j\in I(m) \tag{9a}$$

$$H_{m,m,m} = \overline{CB}_{m,m}'\bar{Q}_m\overline{CB}_{m,m}+\bar{R}_m \tag{9b}$$

$$H_{m,i,j} = \overline{CB}_{m,i}'\bar{Q}_m\overline{CB}_{m,j} \quad \text{for } i,j\in I(m), i\neq m \text{ or } j\neq m \tag{9c}$$

$$c_m = \frac{1}{2}\sum_{i\in I(m)}\sum_{j\in I(m)}\mathbf{x}_i(k)'\overline{CA}_{m,i}'\bar{Q}_m\overline{CA}_{m,j}\mathbf{x}_j(k) \tag{9d}$$

it is possible to redefine problem P as:

$$P: \min\frac{1}{2}\sum_{m=1}^{M}\sum_{i\in I(m)}\sum_{j\in I(m)}\left[\hat{\mathbf{u}}_i'H_{m,i,j}\hat{\mathbf{u}}_j+\mathbf{g}_{m,i,j}'\hat{\mathbf{u}}_i\right]+\sum_{m=1}^{M}c_m \tag{10a}$$

S.to : For $m=1,\ldots,M$:

$$\hat{\mathbf{u}}_m^{\min}\leq\hat{\mathbf{u}}_m\leq\hat{\mathbf{u}}_m^{\max} \tag{10b}$$

$$\hat{\mathbf{y}}_m^{\min}\leq\sum_{i\in I(m)}\left(\overline{CA}_{m,i}\mathbf{x}_i(k)+\overline{CB}_{m,i}\hat{\mathbf{u}}_i\right)\leq\hat{\mathbf{y}}_m^{\max} \tag{10c}$$

This quadratic programming formulation will be used to solve the problem of control calculation in the centralized approach.

2.1 Logarithmic Barrier Method

The logarithmic barrier method is an *interior-point* method for solving convex optimization problems with inequality constraints (Boyd and Vandenberghe, 2004),

$$\text{minimize } f(\mathbf{x}) \tag{11a}$$

$$\text{subject to } A\mathbf{x}\leq\mathbf{b}, \tag{11b}$$

It is assumed that the problem is solvable, *i.e.*, an optimal solution $\mathbf{x}^\star$ exists, and the constraints delimit a closed set. A *barrier function* is any function $B(\mathbf{x}):\Re^n\to\Re$ that satisfies

- $B(\mathbf{x})\geq 0$ for all $\mathbf{x}$ that satisfy $A\mathbf{x}<\mathbf{b}$, and
- $B(\mathbf{x})\to\infty$ as $\mathbf{x}$ approaches the boundary of $\{\mathbf{x}|A\mathbf{x}\leq\mathbf{b}\}$

Being $\mathbf{a}_i'$ the i-th row of A, the idea of the method is to treat the constraints using a logarithmic barrier function as follows:

$$\phi(\mathbf{x}) = -\sum_{i=1}^{m}\log\left(b_i-\mathbf{a}_i'\mathbf{x}\right) \tag{12}$$

where the domain of ϕ is $\mathbf{dom}\ \phi=\{\mathbf{x}|A\mathbf{x}<\mathbf{b}\}$.

With (12), problem (11) can be approximated by:

$$P(\varepsilon):\ \min\ g(\mathbf{x}) = f(\mathbf{x})+\varepsilon\phi(\mathbf{x}) \tag{13}$$

where $\varepsilon>0$ is a parameter that sets the accuracy of the approximation. As ε decreases, more accurate the approximation $P(\varepsilon)$ becomes, whose optimal solution is $\mathbf{x}(\varepsilon)$. The optimal solution is reached by solving (13) for a decreasing sequence of $\varepsilon\to 0^+$, *i.e.*, $\lim_{\varepsilon\to 0^+}\mathbf{x}(\varepsilon)=\mathbf{x}^\star$. The pseudo-code of the barrier method for solving problem (11) appears in Algorithm 1.

Newton's method can be used to compute the optimal solution to $P(\varepsilon)$ using the gradient and the Hessian of $\phi(\mathbf{x})$ and $f(\mathbf{x})$. Algorithm 2 shows how to use Newton's method with a backtracking line search to choose the step size of each iteration. The gradient and the Hessian of the logarithmic barrier function ϕ are given by:

$$\nabla\phi(\mathbf{x}) = \sum_{i=1}^{M}\frac{1}{(b_i-\mathbf{a}_i'\mathbf{x})}\mathbf{a}_i,\ \nabla^2\phi(\mathbf{x}) = \sum_{i=1}^{M}\frac{1}{(b_i-\mathbf{a}_i'\mathbf{x})^2}\mathbf{a}_i\mathbf{a}_i'$$

Algorithm 1: Barrier method

given: strictly feasible $\mathbf{x}$, $\varepsilon:=\varepsilon^0$, $0<\mu<1$, tolerance $e>0$
repeat
  compute $\mathbf{x}(\varepsilon)$ by minimizing $g(\mathbf{x})$, starting at $\mathbf{x}$;
  update $\mathbf{x}:=\mathbf{x}(\varepsilon)$; $\varepsilon:=\mu\varepsilon$;
until $\varepsilon\leq e$;

Algorithm 2: Newton's method

given: a starting point $\mathbf{x}\in\mathbf{dom}\ g$, tolerance $e>0$
repeat
  compute the Newton step: $\Delta\mathbf{x}_{nt}:=-\nabla^2 g(\mathbf{x})^{-1}\nabla g(\mathbf{x})$;
  choose step size
    given: a descent direction $\Delta\mathbf{x}_{nt}$, $\alpha\in(0,0.5)$, $\beta\in(0,1)$
    $t:=1$;
    while $g(\mathbf{x}+t\Delta\mathbf{x}_{nt})>g(\mathbf{x})+\alpha t\nabla g(\mathbf{x})'\Delta\mathbf{x}_{nt}$ **do**
      $t:=\beta t$;
  end
  update: $\mathbf{x}:=\mathbf{x}+t\Delta\mathbf{x}_{nt}$;
  compute the decrement: $\lambda^2:=\nabla g(\mathbf{x})'\nabla^2 g(\mathbf{x})^{-1}\nabla g(\mathbf{x})$;
until $\lambda^2/2\leq e$;

By approximating problem P given in (10) to the equivalent unconstrained form $P(\varepsilon)$, where the constraints on outputs and controls are put together in $\phi(\hat{\mathbf{u}})$, unconstrained minimization algorithms like those described in this section can be used to solve the problem.

3. DISTRIBUTED OPTIMIZATION AND CONTROL

This paper focuses now on the distributed solution of P, discussing how to perform a decomposition of the problem into a network of coupled sub-problems P_m that will be solved by a network of distributed agents (Camponogara and Talukdar, 2005, 2007), and the use of a distributed iterative algorithm to solve these sub-problems.

Each agent will compute a control vector $\hat{\mathbf{u}}_m$. So, for a perfect decomposition, each agent m must have all the information on problem P that depends on $\hat{\mathbf{u}}_m$. Before giving the decomposition, let us define some special sets:

- $O(m)=\{i: m\in I(i), i\neq m\}$ to represent the set of *output neighbors* of m;

- $C(m) = \{(i,j) \in I(m) \times I(m) : i = m \text{ or } j = m\}$ for the sub-system pairs of quadratic terms in the cost function of sub-system m that depend on $\hat{\mathbf{u}}_m$;
- $O(m,k) = \{(i,j) \in I(k) \times I(k) : i = m \text{ or } j = m\}$ for the pairs of quadratic terms in the cost function of sub-system k, $k \in O(m)$, that depend on $\hat{\mathbf{u}}_m$;
- $N(m) = (I(m) \cup O(m) \cup \{i : (i,j) \in O(m,k), k \in O(m)\}) - \{m\}$, which defines the *neighborhood* of agent m, including input and output *neighbors*;
- $\hat{\omega}_m = (\hat{\mathbf{u}}_i : i \in N(m))$, for the set of control signals of the neighbors of agent m;
- $\hat{\mathbf{z}}_m = (\hat{\mathbf{u}}_i : i \in \mathscr{M} - N(m) \cup \{m\})$, for the set of all control signals that are not in $\hat{\omega}_m$ and $\hat{\mathbf{u}}_m$.

The problem P from the view of agent m is defined as:

$$\begin{aligned} P_m: \ \min \ & \frac{1}{2} \sum_{(i,j) \in C(m)} \hat{\mathbf{u}}_i' H_{mij} \hat{\mathbf{u}}_j + \sum_{i \in I(m)} \mathbf{g}_{mmi} \hat{\mathbf{u}}_m + c_m \\ & + \frac{1}{2} \sum_{k \in O(m)} \sum_{(i,j) \in O(m,k)} \hat{\mathbf{u}}_i' H_{kij} \hat{\mathbf{u}}_j \qquad (14a) \\ & + \sum_{k \in O(m)} \sum_{(m,j) \in O(m,k)} \mathbf{g}_{kmj} \hat{\mathbf{u}}_m \\ \text{S.to}: \ & \hat{\mathbf{u}}_m^{\min} \le \hat{\mathbf{u}}_m \le \hat{\mathbf{u}}_m^{\max} \qquad (14b) \\ & \hat{\mathbf{y}}_i^{\min} \le \sum_{j \in I(i)} \left(\overline{CA}_{i,j} \mathbf{x}_j(k) + \overline{CB}_{i,j} \hat{\mathbf{u}}_j \right) \le \hat{\mathbf{y}}_i^{\max}, \qquad (14c) \\ & \text{for all } i \in O(m) \cup \{m\} \end{aligned}$$

It is possible to simplify the representation of problem P_m by grouping some terms as follows:

$$H_m = H_{mmm} + \sum_{k \in O(m)} H_{kmm} \qquad (15a)$$

$$\begin{aligned} \mathbf{g}_m = \ & \frac{1}{2} \sum_{(i,m) \in C(m): i \neq m} \left(H'_{mim} + H_{mmi} \right) \hat{\mathbf{u}}_i + \sum_{i \in I(m)} \mathbf{g}_{mmi} \\ & + \frac{1}{2} \sum_{k \in O(m)} \sum_{(i,m) \in O(m,k): i \neq m} \left(H'_{kim} + H_{kmi} \right) \hat{\mathbf{u}}_i \qquad (15b) \\ & + \sum_{k \in O(m)} \sum_{(m,j) \in O(m,k)} \mathbf{g}_{kmj} \hat{\mathbf{u}}_m \end{aligned}$$

Using the terms defined above, problem P_m is represented as:

$$\begin{aligned} P_m(\hat{\omega}_m): \ \min f_m(\hat{\mathbf{u}}_m) \ = \ & \frac{1}{2} \hat{\mathbf{u}}_m' H_m \hat{\mathbf{u}}_m + \mathbf{g}_m' \hat{\mathbf{u}}_m + c_m \qquad (16a) \\ \text{S.to}: \ & \hat{\mathbf{u}}_m^{\min} \le \hat{\mathbf{u}}_m \le \hat{\mathbf{u}}_m^{\max} \qquad (16b) \\ & \hat{\mathbf{y}}_i^{\min} \le \sum_{j \in I(i)} \left(\overline{CA}_{i,j} \mathbf{x}_j(k) + \overline{CB}_{i,j} \hat{\mathbf{u}}_j \right) \le \hat{\mathbf{y}}_i^{\max}, \qquad (16c) \\ & \text{for all } i \in O(m) \cup \{m\} \end{aligned}$$

This quadratic form will be used by each agent m to compute the control signal $\hat{\mathbf{u}}_m$ in the distributed approach.

Some properties about the decomposition are:

- $P_m(\hat{\omega}_m)$ consists of problem P with all the objective terms and constraints that depend on $\hat{\mathbf{u}}_m$;
- each sub-problem $P_m(\hat{\omega}_m)$ is convex.

3.1 Distributed Algorithm

This section describes briefly the distributed algorithm for the agent network to reach a solution. Let $P_m(\varepsilon)$ be the centering problem for $P_m(\hat{\omega}_m)$ with a given ε:

$$P_m(\varepsilon): \ \min \ f_m(\hat{\mathbf{u}}_m) + \varepsilon \phi_m(\hat{\mathbf{u}}_m) \qquad (17)$$

where $\phi_m(\hat{\mathbf{u}}_m)$ is the logarithmic barrier function of the constraints given in (16). It is important to note that the problem to be solved by each agent is much simpler than the one used in the centralized formulation. So, the distributed solution must encompass a sequence of steps before the optimal control sequence is reached (de Oliveira and Camponogara, 2008).

First, define the vector $\hat{\mathbf{u}}^k = \left(\hat{\mathbf{u}}_1^k, \ldots, \hat{\mathbf{u}}_M^k\right)$ with the set of all control variables of P at iteration k. For a given ε, all agents have to *negotiate* to find a solution for each $P_m(\varepsilon)$ in the network. And this process is repeated for a decreasing sequence of $\varepsilon \to 0^+$. The convergence to a stationary solution is ensured by respecting two assumptions:

(1) *Synchronous Work:* if agent m revises its decisions at iteration k, then:
   (a) agent m uses $\hat{\omega}_m^k$ to produce an approximate solution of $P_m(\varepsilon)$;
   (b) all the neighbors of agent m keep their decisions at iteration k: $\hat{\mathbf{u}}_i^{k+1} = \hat{\mathbf{u}}_i^k$ for all $i \in N(m)$.

(2) *Continuous Work:* if $\hat{\mathbf{u}}^k$ is not a stationary point for problems $P_i(\varepsilon)$, $i \in \mathscr{M}$, then at least one agent m for which $\hat{\mathbf{u}}_m^k$ is not a stationary point for $P_m(\varepsilon)$ produces a new iterate $\hat{\mathbf{u}}_m^{k+1}$.

Condition (a) of Assumption 1 and Assumption 2 hold by arranging the agents to iterate repeatedly in a sequence $\langle S_1, \ldots, S_r \rangle$, where $S_i \subseteq \mathscr{M}$, $\cup_{i=1}^r S_i = \mathscr{M}$, and all distinct pairs $m, n \in S_i$ are non-neighbors for all i. The pseudo-code of the distributed barrier method for solving the problem network $\{P_m(\varepsilon)\}$ is given in Algorithm 3.

Algorithm 3: Distributed barrier method

```
given: strictly feasible û⁰ = (û₁⁰, ..., û_M⁰), ε = ε⁰ > 0,
       0 < μ < 1, tolerance e > 0, and a sequence ⟨S₁, ..., S_r⟩
repeat
    Define the initial group of decoupled agents, i := 1;
    Define a flag for the stationary test, δ := false;
    repeat
        each agent m ∈ S_i receives ω̂_m^k and computes û_m^{k+1} by
        solving P_m(ε) starting at û_m^k;
        each agent j computes û_j^{k+1} = û_j^k for j ∉ S_i;
        û^{k+1} := (û₁^{k+1}, ..., û_M^{k+1});
        if û^{k+1} is a stationary point for P_i(ε), ∀i ∈ M then
            δ := true;
        else
            k := k + 1;
            i := (i mod r) + 1;
    until δ = true ;
    ε := με;
until ε < e ;
```

4. COMPUTATIONAL ANALYSIS

4.1 Distillation Column

This section presents the application of model predictive control to a benchmark problem, comparing the performance of the centralized and distributed approaches. The model of the heavy oil fractionator utilized is referred in the literature as the Shell Oil's heavy oil fractionator (Prett and Morari, 1987; Camacho and Bordons, 2004). It relates the controlled variables y_1, y_2, and y_3 that correspond to the top endpoint composition, side end composition, and bottom reflux temperature, respectively, with the manipulated variables u_1, u_2, and u_3, corresponding to top draw, side draw, and bottom reflux duties. The discrete model is obtained with a sampling time of 4 minutes, and it is possible to obtain a state space representation in the form:

$$\begin{bmatrix} \mathbf{x}_{11}(k+1) \\ \mathbf{x}_{21}(k+1) \\ \mathbf{x}_{31}(k+1) \\ \mathbf{x}_{12}(k+1) \\ \vdots \\ \mathbf{x}_{33}(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & & & \\ & A_{21} & \emptyset & \\ & \emptyset & \ddots & \\ & & & A_{33} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{11}(k) \\ \mathbf{x}_{21}(k) \\ \vdots \\ \mathbf{x}_{33}(k) \end{bmatrix}$$

$$+ \begin{bmatrix} B_{11} & 0 & 0 \\ B_{21} & \vdots & \vdots \\ B_{31} & 0 & \\ 0 & B_{12} & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & B_{33} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \\ y_3(k+1) \end{bmatrix} = \begin{bmatrix} C_{11} & 0 & 0 & C_{12} & \cdots & 0 \\ 0 & C_{21} & 0 & 0 & \ddots & \vdots \\ 0 & 0 & C_{31} & 0 & \cdots & C_{33} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{11}(k) \\ \mathbf{x}_{21}(k) \\ \vdots \\ \mathbf{x}_{33}(k) \end{bmatrix}$$

where each group $\mathbf{x}_{ij}$, A_{ij}, B_{ij}, and C_{ij} represents the couplings among the outputs and controls of the transfer function located in line i and column j of the transfer function matrix (Camacho and Bordons, 2004).

This formulation makes the distillation column a special case of the theory developed in the previous sections. For this special case, because the column is fully coupled, the set $I(m)$ is equal to $\{1,2,3\}$ for any m, and the *neighborhood* is equal to the *output* for each agent m. Other sets are given in Table 1.

Table 1. Sets used in the problem decomposition

m	$O(m)$	$C(m)$	$O(m,k)$
1	{2,3}	{(1,1),(1,2),(2,1),(1,3),(3,1)}	$O(1,2) = O(1,3) = C(1)$
2	{1,3}	{(1,2),(2,1),(2,2),(2,3),(3,2)}	$O(2,1) = O(2,3) = C(2)$
3	{1,2}	{(1,3),(3,1),(2,3),(3,2),(3,3)}	$O(3,1) = O(3,2) = C(3)$

Three different algorithms were used to solve the problem for the purpose of comparison:

- *Centralized Quadratic Programming* (cent_{QP}): the solution of P is obtained using a specific solver in Matlab® for problems in the quadratic form with constraints;
- *Centralized Barrier* (cent_{Br}): the solution of P is reached using the logarithmic barrier method and the centralized formulation;

Table 2. Numerical results

	dist_{Br}		cent_{Br}		cent_{QP}	
T	time	objective	time	objective	time	objective
1	0.0332	5.6188	0.0133	5.6188	0.0099	5.6188
2	0.0386	11.5680	0.0200	11.5680	0.0096	11.5680
5	0.0553	40.5596	0.0243	40.5596	0.0096	40.5596
10	0.0857	73.1869	0.0423	73.1868	0.0109	73.1868
15	0.1417	85.9801	0.0670	85.9797	0.0157	85.9797
20	0.2527	91.2292	0.1211	91.2284	0.0197	91.2284
25	0.3880	93.7129	0.2004	93.7115	0.0361	93.7115
30	0.6258	95.0680	0.2657	95.0660	0.0302	95.0660

- *Distributed Barrier* (dist_{Br}): each problem P_m is solved by a different agent and the constraints are treated with the logarithmic barrier method.

The criteria of convergence in Newton's method, which is used in cent_{Br} and dist_{Br} to solve centering problems, is $\lambda^2/2 \leq 10^{-5}$. The convergence criterion for the logarithmic barrier method is $\varepsilon \leq 10^{-4}$, while the stationary test is satisfied with $e \leq 10^{-4}$. The weights on control action and output deviation were set equal because their adjustment is not the focus of this work, but it is clear that the choice of weights affects directly the compromise between performance and robustness.

Ten different feasible start points were chosen at random in the experiments and the initial ε was defined as 10^3. The previous solution was not used as an initial approximation for reoptimization to induce worst-case scenarios. The analyses considered each type of algorithm, different lengths of prediction horizon (T), and rate of decrease $\mu \in \{0.05,\ 0.1,\ 0.3,\ 0.5\}$ for the interior-point methods.

4.2 Numerical Results

The objective function given in (7) is used for comparison as the cost of the computational experiments. Table 2 has the results of the accumulated cost obtained with the ten start points and the four different values for μ. The cost difference between dist_{Br} and the centralized approach is less than 3×10^{-3}, proving that solving the set of distributed problems, $\{P_m\}$, and solving the centralized problem, P, is equivalent. The gap in cost is due to the acceptance range of the convergence criteria.

Table 2 also contains the results about the time spent in the experiments, comparing cent_{QP}, cent_{Br} and dist_{Br}. A computer with an AMD Turion™ 64x2 1.60 GHz processor and 2048 MB of memory was used to perform the experiments. Time is given in seconds and represents the mean time spent to compute the control actions of forty different experiments with each algorithm. As expected, the greater the value of T, more time is necessary to reach the solution, but the time used was always less than one second, which is a very good result, considering real applications.

The iterations between agents to exchange information were also counted. Fig. 1 depicts the average number of iterations for the experiments with varying prediction horizon and decrease rate of the barrier method.

The convergence criterion can be relaxed to minimize the number of iterations between agents. In this case, due to the strong couplings among the variables, the information exchange is considerable. More scattered models will be used in future experiments, which are more suitable for a distributed approach.

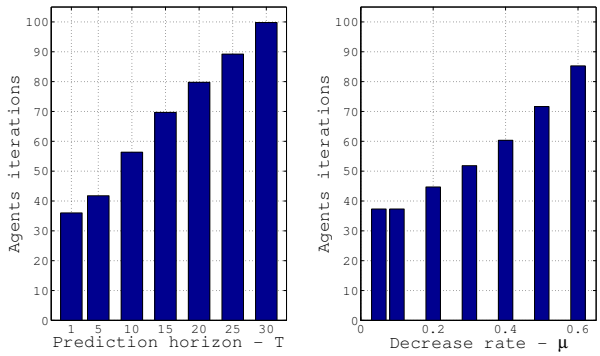

Fig. 1. Algorithm iterations varying T and μ, respectively.

Ending the analyses, it can be said that a network of distributed agents can solve the set of sub-problems $\{P_m\}$, rather than having a single agent solve P in centralized MPC, without incurring great loss of performance. Other adjustments can be made in the algorithm, such as the limit on the number of iterations, to guarantee the fulfillment of the deadlines, but all modifications have a compromise between speed and quality.

5. CONCLUSION

This work presented a distributed MPC framework with constraints on output and control-input signals. The methodology of problem decomposition was outlined and the barrier method was used to deal with the constraints, replacing the constrained minimization problem by a sequence of unconstrained minimization problems. Centralized MPC algorithms and the distributed MPC algorithm were applied to solve a regulation problem in a distillation column model.

The performance of distributed MPC in the distillation column scenario was comparable to the performance obtained with centralized MPC. The computational cost necessary to solve each agent problem was less than the centralized case. This advantage might allow the use of the distributed algorithm in machines with less computational resources.

It is worth emphasizing that distributed predictive control is more appropriate for multivariable problems where the couplings are scattered, which does not happen with the model of the distillation column.

Future works will focus on the implementation of reference tracking and state observers. As well as the whole study of how to introduce these extensions in a distributed algorithm. Another goal is to look into other scenarios that can be more appropriate for the application of distributed algorithms.

REFERENCES

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Camacho, E.F. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag, London.

Camponogara, E. and de Oliveira, L.B. (2009). Distributed optimization for model predictive control of linear dynamic networks. Accepted by *IEEE Transactions on Systems, Man, and Cybernetics – Part A*.

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

Camponogara, E. and Talukdar, S.N. (2005). Designing communication networks to decompose network control problems. *INFORMS Journal on Computing*, 17(2), 207–223.

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

de Oliveira, L.B. and Camponogara, E. (2008). Multi-agent model predictive control for split signaling in urban traffic networks. In *Proceedings of the 5th Workshop on Agents in Traffic and Transportation*, 21–28. Estoril, Portugal.

Dunbar, W.B. (2007). Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 57(7), 1249–1263.

Giovanini, L. and Balderud, J. (2006). Game approach to distributed model predictive control. In *Proceedings of the International Control Conference*. Glasgow, UK.

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

Mercangöz, 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.

Motee, N. and Jadbabaie, A. (2006). Distributed receding horizon control of spatially invariant systems. In *Proceedings of the American Control Conference*, 731–736.

Prett, D.M. and Morari, M. (1987). *Shell Process Control Workshop*. Butterworths, Boston.

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