Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Distributed Optimization for Predictive Control with Input and State Constraints: Preliminary Theory and Application to Urban Traffic Control

Eduardo Camponogara, Helton Fernando Scherer, and Leonardo Vilá Moura
Department of Automation and Systems Engineering
Federal University of Santa Catarina
Florianópolis, SC 88040-900, Brazil

***Abstract*—Distributed model predictive control (DMPC) advocates the distribution of sensing and decision making to operate large, geographically distributed systems such as the power grid and traffic networks. This paper presents a distributed optimization framework for DMPC of linear dynamic networks with constraints on each network node. A linear dynamic network can be thought of as a directed graph, whose nodes have local dynamics that depend on the local and upstream control signals and are subject to constraints on state and control variables. The distributed algorithm is based on interior-point methods and can be shown to converge to a globally optimal solution. Some theoretical results are stated and a preliminary application to green-time control in urban traffic networks is described.**

***Index Terms*—distributed MPC; distributed optimization; interior-point method; urban traffic control.**

I. Introduction

Model predictive control (MPC) has become the leading technology for the operation of complex dynamic systems. The appeal stems from its ability to handle constraints in a systematic manner, the ease of reconfiguration, and the potential to reach optimal solutions [1]. Other control alternatives typically handle constraints in an *ad hoc* manner or are limited in the class to which they are suitable. In essence, MPC converts a dynamic optimization problem in which variables are time dependent into a series of static optimization problems to be solved with standard optimization algorithms.

The centralization of computations and the communication between sensors and the central unit are chief obstacles to the use of MPC in geographically distributed systems, such as the power grid and traffic networks. Instead, distributed model predictive control (DMPC) is an evolving technology that advocates the distribution of sensing and control while preserving the same features of standard MPC, namely the use of prediction models, explicit handling of constraints, and the use of optimization algorithms [2], [3]. The works on DMPC can be roughly divided in control-based and optimization-based frameworks.

Control-based frameworks are mostly concerned with stability and control performance. The solution of the underlying optimization problem is not typically addressed. [4], [5] develop procedures for stabilizing a system of dynamically decoupled sub-systems that mutually influence a performance criterion. [6] is also concerned with stabilization of dynamically decoupled sub-systems and uses principles of Nash optimality. [7] develops algorithms for cooperation between distributed controllers that ensure closed-loop stability.

Optimization-based frameworks are somewhat concerned with the solution of the optimization problems. [8] proposes a distributed algorithm that applies Lagrangian duality to handle coupling constraints among agents. [9] develops a cooperation-based iterative procedure that ensures convergence to the global optimum for linear systems and with constraints on the local controls. [10] presents synchronous and asynchronous solutions of optimization problems, proposing a high level optimization framework and safety margins for meeting constraints. More recently, [11] focuses on linear dynamic networks with local control-input constraints, which allowed the design of a distributed feasible direction method that can reach the global optimum.

This paper presents initial results on an extension of our preceding work [11] that accounts for constraints on state variables besides the constraints on control inputs. The structure of the problem changes significantly as the agents become coupled through the constraint structure. To that end, a distributed interior-point method is outlined and some theoretical results are only stated. The paper also gives some numerical results of the application of this DMPC framework to the green-time control of urban traffic networks

II. Dynamic Model, Control, and Optimization

A linear dynamic network consists of M sub-systems that have local linear dynamics that depend on the controls of other sub-subsystems [11]. Each sub-system m has a local control vector $\mathbf{u}_m \in \mathbb{R}^{p_m}$ and a local state $\mathbf{x}_m \in \mathbb{R}^{n_m}$ which is influenced by the local variables and the controls of neighboring sub-systems. The notion of neighborhood is modeled by a coupling graph $G = (V, E)$, with $V = \{1, \ldots, M\}$ being the vertex set and $E \subseteq V \times V$ being the arc set, according to which sub-system i directly influences the state of sub-system j if $(i, j) \in E$. Assuming discrete time dynamics, the state equation for sub-system m is:

$$\mathbf{x}_m(t+1) = A_m \mathbf{x}_m(t) + \sum_{i \in I(m)} B_{mi} \mathbf{u}_i(t) \qquad (1)$$

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

where $I(m) = \{m\} \cup \{i : (i,m) \in E\}$ is the set of *input neighbors* of sub-system m which encompasses all sub-systems that have a direct influence on sub-system m. Notice that sub-system m influences itself. Each sub-system m is subject to local constraints on the inputs,

$$D_m \mathbf{u}_m(t) \leq \mathbf{d}_m \tag{2a}$$

and state constraints,

$$\mathbf{x}_m^{\min} \leq \mathbf{x}_m(t) \leq \mathbf{x}_m^{\max} \tag{2b}$$

This paper considers only the regulation problem, but this is not a limitation because often the problem of following a trajectory can be recast as a regulation problem. Given the current state $\mathbf{x}(t) = (\mathbf{x}_1, \ldots, \mathbf{x}_M)(t)$ of the network at time t, the MPC optimization problem P for regulating the network with a quadratic cost is formulated as:

$$\min \ f = \frac{1}{2} \sum_{m=1}^{M} \phi_m \tag{3a}$$

where the local objective of sub-system m is defined as:

$$\phi_m = \sum_{j=0}^{T-1} [\hat{\mathbf{x}}_m(t+j+1|t)' Q_m \hat{\mathbf{x}}_m(t+j+1|t) + \hat{\mathbf{u}}_m(t+j|t)' R_m \hat{\mathbf{u}}_m(t+j|t)]$$

and subject to:

$$\hat{\mathbf{x}}_m(t+j+1|t) = A_m \hat{\mathbf{x}}_m(t+j|t) + \sum_{i \in I(m)} B_{mi} \hat{\mathbf{u}}_i(t+j|t) \tag{3b}$$

$$D_m \hat{\mathbf{u}}_m(t+j|t) \leq \mathbf{d}_m \tag{3c}$$

$$\mathbf{x}_m^{\min} \leq \hat{\mathbf{x}}_m(t+j+1|t) \leq \mathbf{x}_m^{\max} \tag{3d}$$

$$\hat{\mathbf{x}}_m(t|t) = \mathbf{x}_m(t) \tag{3e}$$

for $m = 1, \ldots, M$, $j = 0, \ldots, T-1$, where $\hat{\mathbf{u}}_m(t+j|t)$ is the prediction calculated at time t for the control input to sub-system m at time $t+j$, and similarly $\hat{\mathbf{x}}_m$ is the state prediction. T is the length of the prediction and control horizon. Q_m is symmetric positive semi-definite and R_m is symmetric positive definite.

The green time (split) control of an urban traffic network can be cast as an MPC regulation problem on a linear dynamic network. Consider the traffic network depicted in Fig. 1. Each junction defines a sub-system whose state variables are the numbers of vehicles on the roads leading to the junction and whose control signals correspond to the green time for each phase. At junction 3, the state of the sub-system is $\mathbf{x}_3 = (x_6, x_7)$ and the local control vector is $\mathbf{u}_3 = (u_6, u_7)$ with the green time for each queue, assuming there is a phase for each queue. Vehicles accumulate in road 6 (road 7) as the queues at junction 1 (junction 4) are discharged. The vehicles depart from road 6 (road 7) depending on the green time u_6 (u_7). Assuming that vehicles flow through each junction at the saturation rate [12], the state of any sub-system m evolves dynamically as a function of its current state $\mathbf{x}_m$, the control signals at the upstream sub-systems $I(m) - \{m\}$, and the local control signals $\mathbf{u}_m$. For sub-system 3, the dynamics is modeled by matrices A_3, $B_{3,3}$, $B_{3,1}$, and $B_{3,4}$, which yields $\mathbf{x}_3(t+1) = A_3 \mathbf{x}_3(t) + B_{3,3} \mathbf{u}_3(t) + B_{3,1} \mathbf{u}_1(t) + B_{3,4} \mathbf{u}_4(t)$. The coupling graph of the traffic network appears in Fig. 2.

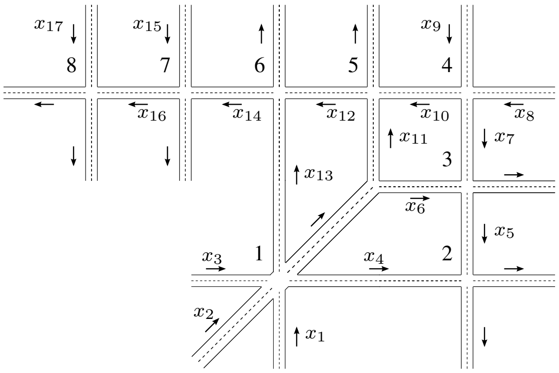


Fig. 1. Traffic network.

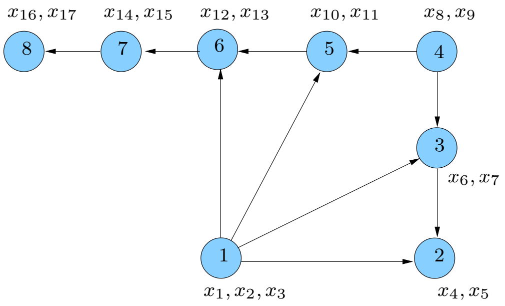


Fig. 2. Coupling graph for the traffic network.

A. Compact Formulation

A compact but equivalent form of P is presented to simplify the design of algorithms. From now on, the term "$|t$" is dropped from all variables for the sake of simplification. Notice that the state of sub-system m at time $(t+\tau)$ is calculated from the initial state and the past control signals as:

$$\hat{\mathbf{x}}_m(t+\tau) = A_m^{\tau} \mathbf{x}_m(t) + \sum_{j=1}^{\tau} \sum_{i \in I(m)} A_m^{j-1} B_{mi} \hat{\mathbf{u}}_{mi}(t+\tau-j)$$

Let $\bar{\mathbf{x}}_m = (\hat{\mathbf{x}}_m(t+1), \ldots, \hat{\mathbf{x}}_m(t+T))$ and $\bar{\mathbf{u}}_m = (\hat{\mathbf{u}}_m(t), \ldots, \hat{\mathbf{u}}_m(t+T-1))$ be vectors collecting the control and state variables over the horizon. Following the guidelines from [11], suitable matrices $\bar{A}_m$ and $\bar{B}_{mi}$ can be defined so that the vector of state predictions is expressed in a compact form:

$$\bar{\mathbf{x}}_m = \bar{A}_m \mathbf{x}_m(t) + \sum_{i \in I(m)} \bar{B}_{mi} \bar{\mathbf{u}}_i \tag{4}$$

Let I_n be the identity matrix of dimension n. Let $\bar{Q}_m = I_T \otimes Q_m$ and $\bar{R}_m = I_T \otimes R_m$ be matrices defined in terms of the

Kronecker product $\otimes$. Further, define the following terms in the objective of sub-system m:

$$\mathbf{g}_{mi}(t) = \bar{B}'_{mi}\bar{Q}_m\bar{A}_m\mathbf{x}_m(t),\ i \in I(m) \tag{5a}$$

$$H_{mij} = \bar{B}'_{mi}\bar{Q}_m\bar{B}_{mj},\ i,j \in I(m),\ i \neq m \text{ or } j \neq m \tag{5b}$$

$$H_{mmm} = \bar{B}'_{mm}\bar{Q}_m\bar{B}_{mm} + \bar{R}_m \tag{5c}$$

$$c(t) = \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{x}_m(t)'\bar{A}'_m\bar{Q}_m\bar{A}_m\mathbf{x}_m(t) \tag{5d}$$

where $\mathcal{M} = \{1,\ldots,M\}$. By redefining the objective of sub-system m using the definitions above, discarding eq. (3b), and collecting the constraints (3c) and (3d) using $\bar{\mathbf{u}}_m$, P becomes:

$$\begin{aligned} \min\ f(\bar{\mathbf{u}}) = \frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{i\in I(m)}\sum_{j\in I(m)} \bar{\mathbf{u}}'_i H_{mij}\bar{\mathbf{u}}_j \\ + \sum_{m\in\mathcal{M}}\sum_{i\in I(m)} \mathbf{g}_{mi}(t)'\bar{\mathbf{u}}_i + c(t) \end{aligned} \tag{6a}$$

subject to the constraints below for all $m \in \mathcal{M}$:

$$\bar{D}_m\bar{\mathbf{u}}_m \leq \bar{\mathbf{d}}_m \tag{6b}$$

$$\bar{\mathbf{x}}_m^{\min} \leq \bar{A}_m\mathbf{x}_m(t) + \sum_{i\in I(m)} \bar{B}_{mi}\bar{\mathbf{u}}_i \leq \bar{\mathbf{x}}_m^{\max} \tag{6c}$$

where vector $\bar{\mathbf{u}} = (\bar{\mathbf{u}}_1,\ldots,\bar{\mathbf{u}}_M)$ collects all of the control variables over the horizon, $\bar{D}_m = I_T \otimes D_m$, $\bar{\mathbf{d}}_m = (\mathbf{d}_m,\ldots,\mathbf{d}_m)$ is a vector of consistent dimension, and $\bar{\mathbf{x}}_m^{\max}$ ($\bar{\mathbf{x}}_m^{\min}$) is defined analogously. While in our preceding work the agents are coupled through the quadratic terms in the objective (6a) and have only local constraints [11], in this paper the linear dynamic network is subject to constraints on the states which interconnect the agents through the constraints (6c).

B. Problem Decomposition

This paper focuses on the break up of problem P into a network $\{P_m\}$ of coupled sub-problems to be solved by a network of agents, each one located at a sub-system and communicating with neighboring agents. The problem decomposition is said to be *perfect* if each sub-problem P_m contains all the objective terms and constraints of P that depend directly on $\hat{\mathbf{u}}_m$. The agents coupled with agent m depend on the following sets:

- $\bar{I}(m) = \{i : m \in I(i),\ i \neq m\}$ is set of *output neighbors* of sub-system m, which contains any sub-system i whose state $\mathbf{x}_i$ is dependent on $\mathbf{u}_m$;
- $C(m) = \{(i,j) \in I(m) \times I(m) : i = m \text{ or } j = m\}$ contains the sub-system pairs of objective terms that are affected by $\mathbf{u}_m$;
- $C(m,q) = \{(i,j) \in I(q) \times I(q) : i = m \text{ or } j = m\}$ has the sub-system pairs of objective terms in ϕ_q, $q \in \bar{I}(m)$, affected by $\mathbf{u}_m$.

For the traffic network depicted in Fig. 1, $I(1) = \{1\}$, $\bar{I}(1) = \{2,3,5,6\}$, $C(1) = \{(1,1)\}$, and $C(1,3) = \{(1,3),(1,4),(1,1),(3,1),(4,1)\}$. Notice that an agent m can affect the state of systems other than $I(m) \cup \bar{I}(m)$. Although sub-system $4 \notin I(1) \cup \bar{I}(1)$, sub-system 1 is coupled to sub-system 4 via sub-system 3. The interdependence between sub-systems is formalized with the notion of neighborhood. From the point of view of an agent m, the variables of a network are divided in three sets:

- *local variables:* the variables in vector $\mathbf{u}_m$;
- *neighborhood variables:* all the variables in vector $\mathbf{y}_m = (\mathbf{u}_i : i \in N(m))$ where $N(m) = I(m) \cup \{i : (i,j) \in C(m,k), k \in \bar{I}(m)\} - \{m\}$ is the neighborhood of agent m. The neighborhood of agent m consists of the sub-systems other than m that are affected by the decision $\mathbf{u}_m$ or whose decisions affect $\mathbf{x}_m$. Notice that $\bar{I}(m) \subseteq N(m)$.
- *remote variables:* all of the other variables which consist of vector $\mathbf{z}_m = (\mathbf{u}_i : i \notin N(m) \cup \{m\})$.

Thus $\mathbf{u} = (\mathbf{u}_m, \mathbf{y}_m, \mathbf{z}_m)$ whichever agent m. For a perfect problem decomposition, problem P_m must account for all the dependencies with the neighbors of agent m. This is achieved if P_m is obtained from P by i) discarding from the objective f the terms not involving $\bar{\mathbf{u}}_m$ and ii) dropping the constraints not associated with agent m. Formally, agent m's local problem $P_m(\bar{\mathbf{y}}_m)$ or simply P_m is:

$$\min\quad f_m = \frac{1}{2}\bar{\mathbf{u}}'_m H_m\bar{\mathbf{u}}_m + \mathbf{g}_m(t)'\bar{\mathbf{u}}_m \tag{7a}$$

$$\text{s.to}:$$

$$\bar{D}_m\bar{\mathbf{u}}_m \leq \bar{\mathbf{d}}_m \tag{7b}$$

$$\bar{\mathbf{x}}_m^{\min} \leq \bar{A}_m\mathbf{x}_m(t) + \sum_{i\in I(m)} \bar{B}_{mi}\bar{\mathbf{u}}_i \leq \bar{\mathbf{x}}_m^{\max} \tag{7c}$$

where H_m is a suitable matrix and $\mathbf{g}_m(t)$ is a suitable vector. A procedure to obtain H_m and $\mathbf{g}_m(t)$ from H_{ijl} and $\mathbf{g}_{ij}(t)$ appears in [11]. Hereafter, we assume a perfect problem decomposition which evidently ensures that:

$$f(\bar{\mathbf{u}}) = f_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) + \bar{f}_m(\bar{\mathbf{y}}_m, \bar{\mathbf{z}}_m) + c(t)$$

for any agent m with $\bar{f}_m$ being a suitable function. To simplify the design of algorithms, we recast P in the following form:

$$P:\quad \min\quad f(\bar{\mathbf{u}}) \tag{8a}$$

$$\text{s.to}: h_i(\bar{\mathbf{u}}) \leq 0,\ i = 1,\ldots,p \tag{8b}$$

with $h_i : \mathbb{R}^n \to \mathbb{R}$ representing all the inequalities, $n = T\sum_{m\in\mathcal{M}} p_m$ being the dimension of $\bar{\mathbf{u}}$, and p being the number of constraints. Further, redefine $P_m(\bar{\mathbf{y}}_m)$ as:

$$P_m(\bar{\mathbf{y}}_m):\quad \min\quad f_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) \tag{9a}$$

$$\text{s.to}: h_i(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) \leq 0,\ i \in h(m) \tag{9b}$$

with $h(m) = \{i : h_i \text{ is a function of } \bar{\mathbf{u}}_m\}$. Notice that h_i does not depend on $\bar{\mathbf{z}}_m$ for all $i \in h(m)$. Let $\Omega = \{\bar{\mathbf{u}} \in \mathbb{R}^n : h_i(\bar{\mathbf{u}}) \leq 0, i = 1,\ldots,p\}$ denote the feasible space of P.

Remark 1: Problem (8a)-(8b) has the following properties:

1) P is convex since f is a convex function and Ω is a convex set;
2) f is a strictly convex and twice continuously differentiable function; and
3) $h_1,\ldots,h_p$ are linear and hence differentiable.

Assumption 1: The following is assumed on P:

1) The interior of the feasible set is nonempty, that is, $\bar{\Omega} = \{\bar{\mathbf{u}} \in \mathbb{R}^n : h_i(\bar{\mathbf{u}}) < 0, i = 1,\ldots,p\} \neq \emptyset$;
2) there exists an optimal solution $\bar{\mathbf{u}}^\star$ to P.

III. Distributed Optimization

The constraints on state variables create difficulties to the distributed solution of $\{P_m\}$. Standard methods based on gradient projection and descent get stuck at fixed points that do not coincide with an optimal solution to P. That is, there are nonoptimal points $\bar{\mathbf{u}}$ at which no single agent m can improve its objective f_m by solving problem $P_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m)$ given the current values of the neighboring variables $\bar{\mathbf{y}}_m$. This section outlines an interior-point method that approximates problem (8a)–(8b) with an unconstrained formulation to which gradient descent can be applied [13], [14]. The approximation accounts for the constraints in the objective using a barrier function[1]. One such function is the *logarithmic barrier:*

$$\phi(\bar{\mathbf{u}}) = -\sum_{i=1}^{p} \log(-h_i(\bar{\mathbf{u}})) \tag{10}$$

whose domain is $\operatorname{dom} \phi = \bar{\Omega}$. The formulation that represents the inequalities in the objective is known as *centering problem*:

$$P(\epsilon) : \min_{\bar{\mathbf{u}} \in \mathbb{R}^n} \theta(\bar{\mathbf{u}}) = f(\bar{\mathbf{u}}) + \epsilon \phi(\bar{\mathbf{u}}) \tag{11}$$

where $\epsilon > 0$ regulates the accuracy of the approximation. The logarithmic barrier $\phi(\bar{\mathbf{u}})$ tends to infinity if any $h_i(\bar{\mathbf{u}}) \to 0$. Function $\theta(\bar{\mathbf{u}})$ is strictly convex because f is strictly convex and $-\epsilon \log(u)$ is convex and nonincreasing in u. Further, $\theta(\bar{\mathbf{u}})$ is twice continuously differentiable. This means that a solution to P of arbitrary accuracy is found by solving $P(\epsilon)$ with a sufficiently small ϵ using standard unconstrained optimization algorithms. This strategy, however, is not efficient because the difficulty of solving $P(\epsilon)$ increases rapidly as ϵ decreases. Instead, the barrier method solves a sequence of $P(\epsilon)$ of decreasing ϵ with a hot start at the previous solution.

The solution $\bar{\mathbf{u}}(\epsilon)$ to $P(\epsilon)$ is unique for any $\epsilon > 0$ because $\theta(\bar{\mathbf{u}})$ is strictly convex. Any such solution is called a *central point*. The set $\mathcal{P} = \{\bar{\mathbf{u}}(\epsilon) : \epsilon > 0\}$ is called *central path.*

A. Centralized Barrier Method

The Lagrangian of P is $\mathcal{L}(\bar{\mathbf{u}}, \boldsymbol{\lambda}) = f(\bar{\mathbf{u}}) + \sum_{i=1}^{p} \boldsymbol{\lambda}_i h_i(\bar{\mathbf{u}})$. Given any Lagrangian vector $\boldsymbol{\lambda} \geq 0$, $\min\{\mathcal{L}(\bar{\mathbf{u}}, \boldsymbol{\lambda}) : \bar{\mathbf{u}}\} = d(\boldsymbol{\lambda})$ induces a lower bound on the optimal objective $f^\star$ of P. First-order optimality conditions applied to the centering problem $P(\epsilon)$ can be used to show that a central point $\bar{\mathbf{u}}(\epsilon)$ defines a dual feasible point. First, associate the following Lagrangian multipliers with $\bar{\mathbf{u}}(\epsilon)$:

$$\boldsymbol{\lambda}_i(\epsilon) = -\frac{\epsilon}{h_i(\bar{\mathbf{u}}(\epsilon))}, \ i = 1, \ldots, p$$

Clearly, $\boldsymbol{\lambda}_i(\epsilon) > 0$ since $h_i(\bar{\mathbf{u}}(\epsilon)) < 0$. Because the pair $(\boldsymbol{\lambda}(\epsilon), \bar{\mathbf{u}}(\epsilon))$ is dual feasible,

$$d(\boldsymbol{\lambda}(\epsilon)) = \min\{\mathcal{L}(\bar{\mathbf{u}}, \boldsymbol{\lambda}(\epsilon)) : \bar{\mathbf{u}}\} =$$

$$f(\bar{\mathbf{u}}(\epsilon)) + \epsilon \sum_{i=1}^{p} -\frac{1}{h_i(\bar{\mathbf{u}}(\epsilon))} h_i(\bar{\mathbf{u}}(\epsilon)) = f(\bar{\mathbf{u}}(\epsilon)) - \epsilon p$$

which yields a lower bound on the optimal objective:

$$f(\bar{\mathbf{u}}(\epsilon)) - f^\star \leq \epsilon p$$

demonstrating that $\bar{\mathbf{u}}(\epsilon)$ converges to $\bar{\mathbf{u}}^\star$ as ϵ tends to 0.

Instead of directly solving the centering problem $P(\epsilon)$ with $\epsilon \leq \gamma/p$ for a desired accuracy γ, the barrier method solves a sequence of centering problems for decreasing values of ϵ until $\epsilon \leq \gamma/p$ as outlined below.

Algorithm 1: Barrier Method

```
input: a strictly feasible $\bar{\mathbf{u}}$, initial $\epsilon$, decrease rate $\mu < 1$,
       and tolerance $\gamma$
initialize: $k := -1$ and $\epsilon^{(0)} := \epsilon$
repeat
    $k := k + 1$
    centering step: obtain $\bar{\mathbf{u}}^{(k)}$ by solving $P(\epsilon^{(k)})$ with
                    initial solution $\bar{\mathbf{u}}$
    if $\epsilon^{(k)} > \gamma/p$ then
        $\bar{\mathbf{u}} := \bar{\mathbf{u}}^{(k)}$ and $\epsilon^{(k+1)} := \mu\epsilon^{(k)}$
until $\epsilon^{(k)} \leq \gamma/p$ ;
output: $\bar{\mathbf{u}}^{(k)}$
```

The barrier method is a general framework to solve problems with inequality constrains (and possibly linear equations) by solving a sequence of unconstrained problems. The decrease rate μ, the choice of initial solution, and initial parameter ϵ play a part in the performance of the method.

A relevant issue is how to solve the centering problem $P(\epsilon)$. Two algorithms are gradient descent and Newton's method using either exact or backtracking line search. Gradient descent is simple because it needs only the gradient of θ and ensures global convergence, but its theoretical convergence rate is only linear. Newton's method with line search also ensures global convergence and its local convergence rate is quadratic, but it is more complex since it needs the Hessian. Both algorithms typically use a backtracking line search, except in situations where the line search can be solved analytically.

B. Distributed Gradient Descent

Here, a distributed gradient descent method is proposed to solve $\{P_m(\epsilon)\}$ instead of $P(\epsilon)$. The agents will produce a series of iterates $\bar{\mathbf{u}}^{(k)}$ that converges to the solution $\bar{\mathbf{u}}(\epsilon)$. Given the neighborhood variables $\bar{\mathbf{y}}_m$, the centering sub-problem associated with agent m is:

$$P_m(\epsilon, \bar{\mathbf{y}}_m) : \min_{\bar{\mathbf{u}}_m} \theta_m(\bar{\mathbf{u}}_m) = f_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) + \epsilon \phi_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m)$$

where $\phi_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) = -\sum_{i \in h(m)} \log(-h_i(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m))$ is the logarithmic barrier depending on $\bar{\mathbf{u}}_m$. Given $(\bar{\mathbf{u}}_m^{(k)}, \bar{\mathbf{y}}_m^{(k)})$, agent m produces the next iterate $\bar{\mathbf{u}}_m^{(k+1)}$ as follows:

$$\bar{\mathbf{u}}_m^{(k+1)} = \bar{\mathbf{u}}_m^{(k)} - s_m^{(k)} \nabla \theta_m(\bar{\mathbf{u}}_m^{(k)})$$

where $s_m^{(k)} > 0$ is a step in the descent direction $-\nabla \theta_m(\bar{\mathbf{u}}_m^{(k)})$. The Newton's direction is another option. Here, the focus is on the gradient descent direction for the sake of simplicity.

[1] A real-valued function within the interior of the feasible space that tends to infinity as the solution is drawn towards the boundary of any constraint.

Backtracking is an inexact line search that finds a step yielding sufficient decrease in the objective. It depends on two parameters, $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$. The step $s_m^{(k)} = \beta^t$ where $t \geq 0$ is the smallest integer such that:

$$\theta_m(\bar{\mathbf{u}}_m^{(k)} - \beta^t \nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})) \leq \theta_m(\bar{\mathbf{u}}_m^{(k)}) - \alpha\beta^t \|\nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})\|^2$$

For small β^t, backtracking eventually terminates since $\theta_m(\bar{\mathbf{u}}_m^{(k)} - \beta^t \nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})) \approx \theta_m(\bar{\mathbf{u}}_m^{(k)}) - \beta^t \|\nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})\|^2 < \theta_m(\bar{\mathbf{u}}_m^{(k)}) - \alpha\beta^t \|\nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})\|^2$ with $\alpha < 1$.

To ensure convergence of the sequence of iterates $\bar{\mathbf{u}}^{(k)}$ to a stationary point[2] $\bar{\mathbf{u}}^\star$ to $P(\epsilon)$, and thereby an optimum, some assumptions regarding the way the agents work and the procedures they use to solve $P_m(\epsilon, \bar{\mathbf{y}}_m)$ are established below.

Assumption 2: (Synchronous Work) If agent m updates its variables in iteration k, then:

1) agent m uses $\bar{\mathbf{y}}_m = \bar{\mathbf{y}}_m^{(k)}$ to obtain an approximate solution $\bar{\mathbf{u}}_m^{(k+1)}$ to $P_m(\epsilon, \bar{\mathbf{y}}_m)$ using the descent method;
2) $\bar{\mathbf{u}}_m^{(k)}$ is not an optimal solution to $P_m(\epsilon, \bar{\mathbf{y}}_m)$; and
3) each neighbor of agent m does not iterate, meaning that $\bar{\mathbf{u}}_i^{(k+1)} = \bar{\mathbf{u}}_i^{(k)}$ for all $i \in N(m)$.

Assumption 3: (Maximum Work) If $\bar{\mathbf{u}}^{(k)}$ is not optimal to $P(\epsilon)$, then agent $m(k) = \arg_{i \in \mathcal{M}} \max\{\|\nabla\theta_i(\bar{\mathbf{u}}_i^{(k)})\|\}$ performs a backtracking line search, starting at $\bar{\mathbf{u}}_{m(k)}^{(k)}$, to produce a solution $\bar{\mathbf{u}}_{m(k)}^{(k+1)}$ for iteration k.

Remark 2: $\theta(\bar{\mathbf{u}}) = \theta_m(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m) + \bar{\theta}_m(\bar{\mathbf{y}}_m, \bar{\mathbf{z}}_m)$ for a suitable function $\bar{\theta}_m$ and all $m \in \mathcal{M}$.

A perfect decomposition ensures the proposition below.

Proposition 1: $\bar{\mathbf{u}}$ is a stationary point to $P(\epsilon)$ if, and only if, the pair $(\bar{\mathbf{u}}_m, \bar{\mathbf{y}}_m)$ is a stationary point to $P_m(\epsilon, \bar{\mathbf{y}}_m)$ for all $m \in \mathcal{M}$.

The distributed gradient descent algorithm for solving $P(\epsilon)$ is outlined in Algorithm 2. At each step k, the agents follow a communication protocol to ensure that the agent of *maximum descent*, $m(k)$, iterates in parallel with any other set of mutually decoupled agents. The set $S(k) \subseteq \mathcal{M}$ with the agents that iterate satisfy the property that $m(k) \in S(k)$ and $i \notin N(j)$ for all $i, j \in S(k)$.

The following proposition can be shown with a rigorous convergence analysis of Algorithm 2.

Theorem 1: Under Assumptions 2 and 3, the distributed gradient descent algorithm yields a sequence of iterates $\bar{\mathbf{u}}^{(k)}$ that converges to the optimal solution $\bar{\mathbf{u}}(\epsilon)$ to the centering problem $P(\epsilon)$.

IV. Signaling Split Control

An urban traffic network (UTN) is formed by several roads interconnected by junctions where traffic lights may be installed to coordinate conflicting traffic streams. The network depicted in Fig. 1 has traffic lights in each junction.

The rising number of motor vehicles is pushing traffic systems around the globe to operate near their limits, invariably increasing congestion and the chance of grid locks that incur delays, raise the level of emissions, and reduce safety and comfort of drivers and commuters [15]. One way to reduce the side effects of congestion is the use of control measures and strategies that optimize the performance of the existing traffic infrastructure.

Algorithm 2: Distributed Gradient Descent Algorithm

```
input: a strictly feasible $\bar{\mathbf{u}}$, barrier parameter $\epsilon$,
       backtracking parameters $\alpha \in (0, 1/2)$ and
       $\beta \in (0, 1)$, and tolerance $\gamma$
initialize: $k := 0$ and $\bar{\mathbf{u}}^{(0)} := \bar{\mathbf{u}}$
while $\|\nabla\theta(\bar{\mathbf{u}}^{(k)})\| > \gamma$ do
    let $S(k)$ be a subset of non-neighboring agents
    for each agent $m \in S(k)$ in parallel do
        obtain $\bar{\mathbf{y}}_m^{(k)}$ from the neighbors $i \in N(m)$
        $s_m^{(k)} := 1$
        while $\theta_m(\bar{\mathbf{u}}_m^{(k)} - s_m^{(k)} \nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})) >$
        $\theta(\bar{\mathbf{u}}_m^{(k)}) - \alpha s_m^{(k)} \|\nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})\|^2$ do
            $s_m^{(k)} := \beta s_m^{(k)}$
        $\bar{\mathbf{u}}_m^{(k+1)} := \bar{\mathbf{u}}_m^{(k)} - s_m^{(k)} \nabla\theta_m(\bar{\mathbf{u}}_m^{(k)})$
    for each agent $m \in \mathcal{M} - S(k)$ in parallel do
        $\bar{\mathbf{u}}_m^{(k+1)} := \bar{\mathbf{u}}_m^{(k)}$
    $k := k + 1$
output: $\bar{\mathbf{u}}^{(k)}$
```

The influence of traffic lights on network flow is determined mostly by the stage specification in each junction, the cycle time, the offset between junctions, and split of green time [16]. [11] shows that the split control problem proposed in [16] can be cast as a regulation problem in a linear dynamic network with constraints on local control signals. The more general constraint structure of the model presented in this paper allows the enforcement of capacity constraints, namely a state variable x_i representing vehicles on road i can be subject to the road capacity $x_i^{\max}$.

Using the store-and-forward model illustrated in Fig. 3, it is possible to obtain a mathematical model that describes how the queue in a respective link evolves according to the initial queue and the physical characteristics of the UTN. Extending such model for all network junctions, the store-and-forward model leads to the following dynamic equation:

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{12}$$

where $\mathbf{x}(t) = (x_1, \ldots, x_{17})(t)$ is the vector with the number of vehicles in each link and $\mathbf{u}(t) = (u_1, \ldots, u_{17})(t)$ is the vector with split signals. This dynamic equation depends on the network topology, stage specification, cycle time, saturation flows, and turning rates. Rather than enforcing the green times to add up to cycle time at each junction, we impose the constraint that this sum does not exceed cycle time and the capacity constraints on the roads. This is expressed in constraint (2a) using an appropriate matrix D_m and vector $\mathbf{d}_m$ at each junction m. And by constraint (2b) with $x_m^{\min} = 0$ and $x_m^{\max}$ at the capacity of road m.

The traffic-responsive urban control (TUC) strategy [16],

[2] A vector satisfying first-order optimality conditions.

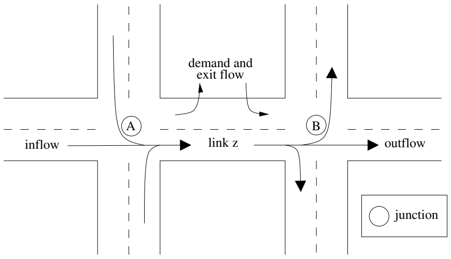


Fig. 3. Illustration of the store-and-forward model.

[12] solves a linear quadratic regulation problem (LQR) to obtain the green times. Because the LQR regulator does take into account any restrictions, this strategy subsequently solves an optimization problem to find the feasible control signals that are closest to the unconstrained LQR signals. Alternatively, some works stress that sliding-horizon control methods [17], [18], such as model predictive control (MPC), are more flexible, are easily adjustable, explicitly handle constraints, and can potentially improve network performance.

As mentioned above, the constraints that force green times to add up to cycle time were dropped because the capacity constraints would invariably become infeasible. The store-and-forward assumption that vehicles depart from a link at saturation flow can lead the number of vehicles to become negative, violating the lower bound constraint on road capacity. Thus, our strategy revises the split signals given by the DMPC framework by extending the green times until they add up to cycle time. The extension is proportional to the green time of each phase.

V. Computational Experiments

The purpose of this section is twofold. First, it aims to compare the performance of the centralized and distributed barrier methods with the performance of a standard algorithm at solving the MPC problem. Second, it provides some preliminary results from the application of the split control model proposed in the preceding section and the standard LQR strategy in a simulated scenario.

A. Numerical Analysis

The algorithms used compute the control signals were:

- *Centralized Quadratic Programming* (cent$_{QP}$): where the solution of P is obtained using a professional solver for problems in the quadratic form with constraints;
- *Centralized Barrier* (cent$_{Br}$): the solution is reached using the logarithmic barrier method (Algorithm 1);
- *Distributed Barrier* (dist$_{Br}$): the solution of P is obtained with a network of agents, whereby sub-problem P_m is solved by a different agent and the constraints are treated with the barrier method.

The traffic scenario consists of the 8 junction UTN depicted in Fig. 1. Following the store-and-forward model, we arrived at the dynamic system (12) for traffic flow and subsequently recast this system as a linear dynamic network consisting of a sub-system for each junction. Along with this model, the centralized barrier method (Algorithm 1) and the distributed barrier method (which uses Algorithm 2 for solving the centering problem) were coded in Matlab®. The algorithm QUADPROG available in Matlab played the role of cent$_{QP}$.

TABLE I
NUMERICAL RESULTS

T	dist$_{Br}$		cent$_{Br}$		cent$_{QP}$	
	time	objective	time	objective	time	objective
1	0.4110	8.4224e6	0.0511	8.4224e6	0.0390	8.4225e6
2	0.6306	8.6068e6	0.0738	8.6068e6	0.0595	8.6068e6
3	0.9054	8.6586e6	0.1838	8.6586e6	0.1254	8.6586e6
4	1.2081	8.6922e6	0.2905	8.6922e6	0.2169	8.6923e6
5	1.4704	8.7292e6	0.4524	8.7291e6	0.2431	8.7292e6
6	1.6502	8.7666e6	0.7493	8.7666e6	0.3505	8.7666e6
7	1.3686	8.7700e6	1.4777	8.7700e6	0.5182	8.7700e6
10	1.5650	8.7407e6	3.0855	8.7407e6	1.9167	8.7407e6
12	1.9610	8.7232e6	4.9732	8.7231e6	2.2237	8.7232e6
15	2.5023	8.6982e6	9.9076	8.6979e6	4.3631	8.6982e6

Several experimental conditions were set up by varying the prediction horizon T, the decrease rate μ of the barrier method, and the initial conditions (*i.e.*, the initial state $\mathbf{x}(t)$ of the network). Table I gives the results of these experiments. Time is given in seconds and the accumulated objective is the sum of the objectives over all experiments. For each horizon length and algorithm, the table shows the average of 40 experiments which were obtained by varying the initial conditions (10 random seeds) and the rate of decrease $\mu \in \{0.1, 0.2, 0.4, 0.6\}$ for the interior-point methods.

The numerical experiments elicit two remarks: first, dist$_{Br}$ was slower than cent$_{QP}$ and cent$_{Br}$ for small horizon lengths; second, the time consumption of the centralized barrier method grows faster than its distributed counterpart as the horizon length increases, which means that for a large scale network the speed of the distributed approach may be competitive with the centralized method. Fig. 4 more clearly shows the computational growth of the algorithms as horizon length increases.

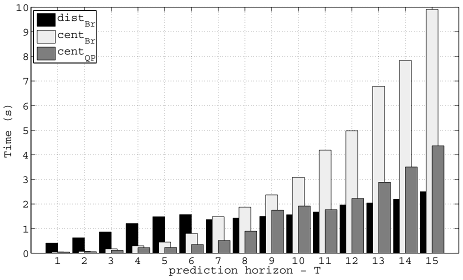


Fig. 4. Processing time.

B. Simulation Analysis

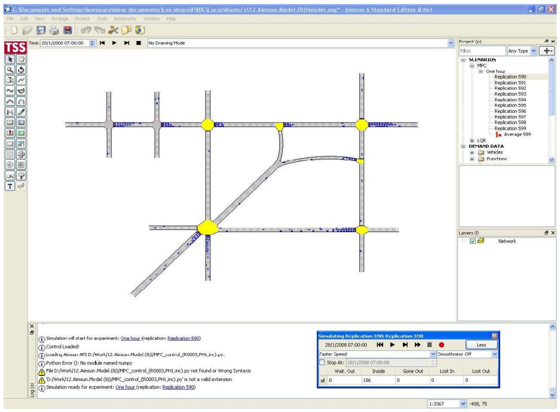
A simulation scenario based on the 8 junction network was implemented with the Aimsun© simulator. Fig. 5 illustrates the interface to Aimsun. Because the numerical results of cent$_{Br}$, dist$_{Br}$, and cent$_{QP}$ are essentially identical, it is plausible to assume that the end performance of these methods will be nearly the same. For this reason, only cent$_{Br}$ and the TUC-LQR strategy [16] were implemented in the simulator.

TABLE II
SIMULATION RESULTS

Statistics	Strategies	
	TUC-LQR	MPC
Delay Time Average (s/km)	144.59	142.92
Density Average (veh/km)	20.31	20.18
Flow Average (veh/h)	6210.67	6216.00
Speed Average (km/h)	22.24	22.25
Total Travel Time Average (h)	164.30	163.60

Details of the simulation set-up are as follows: the cycle time was 120 seconds; the sampling time for the control algorithms was identical to cycle time; the length of the horizon was one; the decrease rate for the barrier method was $\mu = 0.1$; the lost time was fixed in 6 seconds; and the weights on control action cost and state deviation from the nominal state were the same for the design of the LQR control law and the objective of the MPC problem.

Ten simulations of one hour long were performed with different initial conditions. The averages of traffic statistics over the results obtained from these simulations, with the split signals being given by TUC-LQR and MPC, appear in Table II. These results indicate that MPC is slightly better than the TUC-LQR strategy.



Simulation will start for experiment: One hour (replication: Replication 590)
Control Loaded!
Loading Aimsun API D:/Work/12.Aimsun.Model.(8j)/MPC_control_(R0003,PH1,inc).py.
Python Error (): No module named numpy
File D:/Work/12.Aimsun.Model.(8j)/MPC_control_(R0003,PH1,inc).py not found or Wrong Syntaxis
'D:/Work/12.Aimsun.Model.(8j)/MPC_control_(R0003,PH1,inc).py' is not a valid extension
Simulation ready for experiment: One hour (replication: Replication 590)

	Wait. Out	Inside	Gone Out	Lost In	Lost Out
all	0	186	0	0	0

Fig. 5. Aimsun simulation environment.

VI. CLOSING REMARKS

This paper presented a distributed optimization framework for implementing distributed model predictive control of linear dynamic networks. The work extends preceding research by handling constraints on state variables that couple agents through the constraint structure besides the objective. A distributed interior-point method that can reach optimal solutions was outlined. Preliminary results from the application of distributed MPC to the green-time control in a traffic network are also reported. Future research will be geared towards the full development of theoretical results and an extensive simulation analysis in large-scale traffic networks.

ACKNOWLEDGMENT

The authors would like to thank Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) for financial support from grant PRH-34/aciPG/ANP and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) from grant 473841/2007-0.

REFERENCES

[1] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer-Verlag, 2004.

[2] E. Camponogara, D. Jia, B. H. Krogh, and S. N. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[3] E. Camponogara and S. N. Talukdar, "Designing communication networks to decompose network control problems," *INFORMS Journal on Computing*, vol. 17, no. 2, pp. 207–223, 2005.

[4] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, pp. 549–558, 2006.

[5] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, pp. 2105–2115, 2006.

[6] S. Li, Y. Zhang, and Q. Zhu, "Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem," *Information Sciences*, vol. 170, no. 2-4, pp. 329–349, 2005.

[7] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 6680–6685.

[8] R. R. Negenborn, B. D. Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: serial versus parallel schemes," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, 2007.

[9] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1192–1206, 2008.

[10] E. Camponogara and S. N. Talukdar, "Distributed model predictive control: synchronous and asynchronous computation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, vol. 37, no. 5, pp. 732–745, 2007.

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

[12] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] M. Papageorgiou, "Overview of road traffic control strategies," in *Information and Communication Technologies: From Theory to Applications*, 2004, pp. LIX–LLX.

[16] C. Diakaki, "Integrated control of traffic flow in corridor networks," Ph.D. dissertation, Department of Production Engineering and Management, Technical University of Crete, 1999.

[17] L. B. de Oliveira and E. Camponogara, "Predictive control for urban traffic networks: initial evaluation," in *Proceedings of the 3rd IFAC Symposium on System, Structure and Control*, Brazil, 2007.

[18] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Control and optimization methods for traffic signal control in large-scale congested urban road networks," in *Proceedings of the American Control Conference*, New York, USA, 2007, pp. 3132–3138.