

Technical report 06-005

Multi-agent model predictive control for transportation networks: Serial versus parallel schemes*

R.R. Negenborn, B. De Schutter, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:

R.R. Negenborn, B. De Schutter, and H. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial versus parallel schemes," *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'2006)*, Saint-Etienne, France, pp. 339–344, May 2006.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: <http://www.dcsc.tudelft.nl>

*This report can also be downloaded via http://pub.bdeschutter.org/abs/06_005.html

MULTI-AGENT MODEL PREDICTIVE CONTROL FOR TRANSPORTATION NETWORKS: SERIAL VERSUS PARALLEL SCHEMES

Rudy R. Negenborn ^{*,1} Bart De Schutter ^{*} Hans Hellendoorn ^{*}

** Delft Center for Systems and Control,
Delft University of Technology, Delft, The Netherlands
<http://www.dsc.tudelft.nl/>*

Abstract: We consider the control of large-scale transportation networks, like road traffic networks, power distribution networks, water distribution networks, etc. For control of these networks, we propose a multi-agent control scheme in which each agent employs Model Predictive Control. In order to obtain coordination and to improve decision making agents communicate with each other. We compare two Lagrangian-based communication and decision making schemes. One scheme is based on serial iterations between agents, while the other is based on parallel iterations. The schemes are explained theoretically and assessed experimentally by means of simulations on a particular type of transportation network, viz., a power distribution network. The serial scheme shows to have preferable properties compared to the parallel scheme in terms of solution speed and quality.
Copyright © 2006 IFAC

Keywords: Multi-agent control, model predictive control, transportation networks.

1. INTRODUCTION

Transportation networks, like road traffic networks, power distribution networks, water distribution networks, gas networks, etc. are usually large in size, consist of multiple subnetworks, have many actuators and sensors, and therefore show complex dynamics. These transportation networks can be considered at a generic level, at which commodity is brought into the network at sources, flows over links to sinks, and is influenced in its way of flowing by elements inside the network. The similarities between several types of transportation networks are the motivation for studying these networks in a generic way. Results obtained for generic transportation networks can then be specialized and fine-tuned for specific domains.

Typical control goals for transportation networks involve avoiding congestion of links, maximizing

throughput, minimizing costs of control actions, etc. In the daily operation of transportation networks, network operators have to adjust the actuators in the network to meet these control objectives. Control from a single point by a single, centralized, control agent is often not possible due to technical or commercial issues. Technical issues arise from, e.g., communication delays and too high computational requirements. Some commercial issues are, e.g., unavailability of information from one network operator to another, restricted control access, and costs of sensors. Also, robustness and reliability of the network cannot be guaranteed when the single control agent breaks down.

For these reasons, large-scale transportation networks have to be operated using a multi-agent, or distributed, control approach (Siljak, 1991). In such an approach the overall network is considered as consisting of multiple smaller subnetworks. Each of the subnetworks is controlled by an agent with only limited information

¹ Corresponding author, e-mail: r.r.negenborn@tudelft.nl

gathering and processing skills and moreover limited action capabilities.

To determine which actions to take, we propose the use of multi-agent Model Predictive Control (MPC). In a single-agent setting, MPC has shown successful application in the process industry over the last decades (Maciejowski, 2002; Morari and Lee, 1999), and is now gaining increasing attention in fields like power networks (Hines *et al.*, 2005), road traffic networks (Hegyi *et al.*, 2005), and railway networks (De Schutter *et al.*, 2002). Two main advantages of MPC are its explicit way of integrating constraints and its easy way of integrating forecasts. E.g., for transportation networks MPC provides a convenient way to include capacity limits on links, maximums on queue lengths, measurements from upstream sensors, and profiles of demands.

In our multi-agent setting we assume that the network has been divided into subnetworks, and that each subnetwork has been assigned an agent. Each of the agents uses MPC to determine its actions. Each agent performs the following steps:

- (1) Make a measurement of the current state of the subnetwork, and interpret information received from other agents.
- (2) Solve an optimization problem that finds over a certain horizon the actions that result in the best subnetwork behavior according to a specified objective. This typically involves communication.
- (3) Implement the found actions until the next step.
- (4) Move on to the next decision step.

The challenge in implementing such a scheme comes from step 2, since the other steps are trivial once step 2 is solved. The actions that an agent takes influence both the evolution of its own subnetwork, and the evolution of the subnetworks connected to its subnetwork. Since the agents in a multi-agent setting usually have no global overview and can only access a relatively small number of sensors and actuators, predicting the evolution of a subnetwork over a horizon involves even more uncertainty than when a single agent is employed. Communication can reduce this uncertainty, since it allows agents to inform one another about their plans (Camponogara *et al.*, 2002; Dunbar and Murray, 2002). The agents can then take into account these plans and anticipate any undesirable situation. Moreover, through communication agents can obtain agreement on taking actions that give both locally and overall good performance.

We discuss two schemes for obtaining agreement between agents: the first involving serial iterations between agents, the second involving parallel iterations. To make a well-founded choice for either a serial or a parallel scheme, it is important to compare these schemes. To the authors' best knowledge, such a comparison has not been made before in the context of obtaining agreement between agents. Although the par-

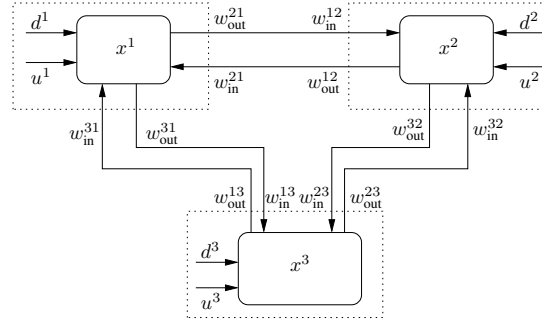


Fig. 1. Each subnetwork model has a set of variables. Internetwork variables form interconnecting constraints between variables of two subnetwork.

allel scheme is more frequently used in literature, in our experiments the serial approach shows preferable properties in terms of solution speed and quality. The rest of this paper is organized as follows. In Section 2 we formalize the setup of the transportation network and the control problem. In Section 3 we give the underlying principle of the schemes we consider, and present the implementation of the two schemes. In Section 4 we experimentally assess the performance of both schemes on a particular transportation network.

2. CONTROL SETTING

2.1 Generic Transportation Network Model

We consider a transportation network partitioned into n subnetworks, each controlled by a control agent that has a model of its subnetwork. Subnetworks are physically connected to each other. The interconnections are modeled using so-called *internetwork variables*. A variable of the model of subnetwork i is an internetwork variable with respect to the model of subnetwork j if it corresponds to a variable in the model of that subnetwork j . We distinguish two types of internetwork variables: internetwork *input* variables and internetwork *output* variables. On one side, the model of subnetwork i contains an internetwork input variable $w_{in,k}^{ji}$ that represents the input caused by subnetwork j on the state of subnetwork i at time step k . On the other hand, the model of subnetwork j contains an internetwork output variable $w_{out,k}^{ij}$ that represents the influence that subnetwork j has on subnetwork i . Since the internetwork input to subnetwork i from j must be equal to the internetwork output from subnetwork j to i , the interconnecting constraints for subnetwork i are:

$$w_{in,k}^{ji} = w_{out,k}^{ij}, \quad w_{out,k}^{ji} = w_{in,k}^{ij},$$

for all $j \in N_i$, where $N_i = \{j_1, \dots, j_{m_i}\}$ is the set of indexes of the m_i subnetworks connected to subnetwork i , as illustrated in Fig. 1.

We model the dynamics of subnetwork i by a discrete-time difference equation:

$$x_{k+1}^i = f^i(x_k^i, u_k^i, d_k^i, w_{in,k}^i) \quad (1)$$

$$w_{\text{out},k}^{ji} = C^{ji} [(x_k^i)^T (u_k^i)^T (d_k^i)^T]^T, \quad (2)$$

for all $j \in N_i$, where at time step k , for subnetwork i , x_k^i are the dynamic states, u_k^i are the inputs, d_k^i are the local disturbances, $w_{\text{in},k}^i = [(w_{\text{in},k}^{j_1 i})^T \dots (w_{\text{in},k}^{j_{m_i} i})^T]^T$ are the internetwork input variables, and $w_{\text{out},k}^{ji}$ are the internetwork output variables. The function f^i represents the dynamic state transitions, and the matrix C^{ji} is of appropriate dimensions and contains 0 entries on each row, except for a single 1 corresponding to a variable that is an internetwork output of subnetwork i with respect to subnetwork j . Those elements of $[(x_k^i)^T (u_k^i)^T (d_k^i)^T]^T$ that are not also internetwork output variable are called *local variables* z_k^i of agent i . The remaining variables, i.e., $w_{\text{in},k}^{ji}$ and $w_{\text{out},k}^{ji}$ together, are called *internetwork variables* w_k^{ji} of agent i .

2.2 Distributed Control Problem

All agents have to determine inputs u_k^i to their actuators that give the best local and overall performance. In MPC, the agents do this by determining over a horizon of N steps optimal inputs according to an objective function $J_z^i(\tilde{z}_k^i)$, where the symbol $\tilde{\cdot}$ over a variable indicates variables over the horizon, e.g., $\tilde{z}_k^i = [(z_k^i)^T \dots (z_{k+N-1}^i)^T]^T$. We formulate the *overall* control problem as:

$$\min_{\tilde{z}_k^1, \tilde{w}_k^1, \dots, \tilde{z}_k^n, \tilde{w}_k^n} \sum_{i=1}^n \left(J_z^i(\tilde{z}_k^i) \right) \quad (3)$$

subject to:

$$\tilde{w}_{\text{in},k}^{ji} = \tilde{w}_{\text{out},k}^{ij} \quad \text{for } i = 1, \dots, n, \text{ and } j \in N_i \quad (4)$$

$$\tilde{w}_{\text{in},k}^{ij} = \tilde{w}_{\text{out},k}^{ji} \quad \text{for } i = 1, \dots, n, \text{ and } j \in N_i, \quad (5)$$

and the dynamics of the subnetworks (1)–(2) over the horizon, where $\tilde{w}_k^i = [(\tilde{w}_k^{j_1 i})^T \dots (\tilde{w}_k^{j_{m_i} i})^T]^T$.

The overall control problem (3) is not separable into subproblems using only local variables \tilde{z}_k^i and internetwork variables \tilde{w}_k^i of one agent i alone due to the interconnecting constraints (4)–(5). Using an augmented Lagrangian formulation of the problem (Bertsekas and Tsitsiklis, 1997), we remove the equality constraints by adding for each constraint an additional cost term to the objective function of each agent based on Lagrangian multipliers (Boyd and Vandenberghe, 2004), and a quadratic term involving the difference between internetwork input and internetwork output variables between two subnetworks. Through iterations of solving an optimization problem and doing a Lagrangian multiplier update, a solution to the original problem may be found. Denoting by subscript $s|k$ a variable at iteration s of decision step k , the following steps are made:

(1) Find $\tilde{z}_{s+1|k}^1, \tilde{w}_{s+1|k}^1, \dots, \tilde{z}_{s+1|k}^n, \tilde{w}_{s+1|k}^n$ from:

$$\min_{\tilde{z}_k^1, \tilde{w}_k^1, \dots, \tilde{z}_k^n, \tilde{w}_k^n} \sum_{i=1}^n \left[J_z^i(\tilde{z}_k^i) + \sum_{j \in N_i}$$

$$\left((\tilde{\lambda}_s^{ji})^T (\tilde{w}_{\text{in},k}^{ji} - \tilde{w}_{\text{out},k}^{ij}) + \frac{c}{2} \left\| \tilde{w}_{\text{in},k}^{ji} - \tilde{w}_{\text{out},k}^{ij} \right\|^2 + (\tilde{\lambda}_s^{ij})^T (\tilde{w}_{\text{in},k}^{ij} - \tilde{w}_{\text{out},k}^{ji}) + \frac{c}{2} \left\| \tilde{w}_{\text{in},k}^{ij} - \tilde{w}_{\text{out},k}^{ji} \right\|^2 \right), \quad (6)$$

subject to the dynamics of the subnetworks (1)–(2) over the horizon.

(2) Compute new Lagrangian multipliers,

$$\tilde{\lambda}_{s+1}^{ji} = \tilde{\lambda}_s^{ji} + c(\tilde{w}_{\text{in},s+1|k}^{ji} - \tilde{w}_{\text{out},s+1|k}^{ij}) \quad (7)$$

$$\tilde{\lambda}_{s+1}^{ij} = \tilde{\lambda}_s^{ij} + c(\tilde{w}_{\text{in},s+1|k}^{ij} - \tilde{w}_{\text{out},s+1|k}^{ji}). \quad (8)$$

(3) Move on to the next iteration $s+1$ and repeat the cycle. Stop when the Lagrangian multipliers do not change anymore.

The constant c is a positive scalar (see Section 3 for more details). The Lagrangian multipliers $\tilde{\lambda}_s^{ji}$ and $\tilde{\lambda}_s^{ij}$ can be initialized arbitrarily. Under convexity assumptions on the objective functions and constraints it can be proved that when the Lagrangian multipliers do not change anymore from one iteration to the next, a local minimum of the original problem has been found (Bertsekas and Tsitsiklis, 1997).

Although the new formulation does not have the interconnecting constraints explicitly, due to the quadratic terms in (6) the overall problem is still not separable. The schemes we discuss next transform the non-separable problem into a sequence of separable problems of the form:

$$\min_{\tilde{z}_k^i, \tilde{w}_k^i} J_z^i(\tilde{z}_k^i) + \sum_{j \in N_i} J_w^i(\tilde{w}_k^{ji}), \quad (9)$$

subject to the dynamics of the subnetwork (1)–(2) over the horizon, where the additional cost term $J_w^i(\tilde{w}_k^{ji})$ deals with the internetwork variables.

3. SERIAL VERSUS PARALLEL SCHEME

The two schemes that we introduce have the following intuitive motivation. Since the evolution of a subnetwork is influenced by neighboring subnetworks, the actions that an agent computes are best with respect to the predicted internetwork input variables $\tilde{w}_{\text{in},k}^{ji}$ for all $j \in N_i$. To reduce the uncertainty in the predictions of the internetwork input variables agent i has to come to an agreement with the agents $j \in N_i$ on the values of their internetwork output variables $\tilde{w}_{\text{out},k}^{ij}$. The agents obtain agreement through iterations that inform the agents about how the other agents prefer their internetwork inputs to be. To obtain agreement, an agent i does not only compute optimal local variables for its own subnetwork, but also optimal internetwork input variables $\tilde{w}_{\text{in},k}^{ji}$. Moreover, the other agents $j \in N_i$ compute both their optimal local variables and optimal internetwork output variables $\tilde{w}_{\text{out},k}^{ij}$. Through exchange of these optimal internetwork variables, the values of the internetwork output and input variables have to converge to each other, and a set of actions

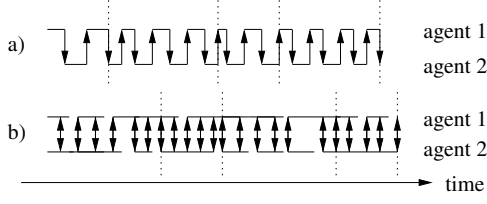


Fig. 2. Considered communication schemes between two agents. Arrows indicate information exchange. Dotted lines are actions being implemented. Horizontal lines are optimization problems being solved. a) Serial; b) Parallel.

that is (perhaps locally) optimal for all agents has to be found.

We consider two schemes that implement the above ideas, see Fig. 2, but that differ in the times at which the agents solve their problems: they either do this after serial iterations, or after parallel iterations.

3.1 Serial Implementation

The serial implementation is the result of using a *Block Coordinate Descent* (Bertsekas and Tsitsiklis, 1997; Royo, 2001) for dealing with the non-separable quadratic term of (6). The approach minimizes the quadratic term directly, in a serial way. One agent after another minimizes its local and internetwork variables while the other variables are fixed.

Given the information $\tilde{w}_{\text{prev},k}^{ij} = \tilde{w}_{s+1|k}^{ij}$ computed at the current iteration s for each agent $j \in N_i$ that has solved its problem *before agent i in the current iteration s* , and given the previous information $\tilde{w}_{\text{prev},k}^{ij} = \tilde{w}_{s|k}^{ij}$ of the last iteration $s - 1$ for the other agents, agent i solves problem (9) using the following additional objective function:

$$J_w^i(\tilde{w}_k^{ji}) = \left[(\tilde{\lambda}_s^{ji})^T \quad (-\tilde{\lambda}_s^{ij})^T \right] \begin{bmatrix} \tilde{w}_{\text{in},k}^{ji} \\ \tilde{w}_{\text{out},k}^{ji} \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{w}_{\text{in},\text{prev},k}^{ji} \\ \tilde{w}_{\text{out},\text{prev},k}^{ji} \end{bmatrix} - \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}_{\text{in},k}^{ji} \\ \tilde{w}_{\text{out},k}^{ji} \end{bmatrix} \right\|^2.$$

The serial implementation consists of the following steps. At decision step k , iteration s :

- (1) For $i = 1, \dots, n$, *one agent after another*:
 - (a) Agent i determines $\tilde{z}_{s+1|k}^i, \tilde{w}_{s+1|k}^{ji}$ by solving (9).
 - (b) Agent i sends to agent $j \in N_i$ the computed values $\tilde{w}_{s+1|k}^{ji}$.
- (2) After all agents have solved their problems at one iteration, they update their Lagrangian multipliers using (7)–(8).
- (3) Each agent moves to the next iteration $s + 1$, and the cycle starts over, unless the Lagrangian multipliers do not change anymore.

The positive scalar c penalizes the deviation from the internetwork variable iterates that were computed by

the other agents before agent i or during the last decision step. This makes that the internetwork variables that agent i computes at the current iteration will stay close to the internetwork variables of agent j computed earlier if c is chosen larger. With c increasing, it becomes more expensive for an agent to deviate from the internetwork-variable values computed by the other agents. This results in a faster convergence of the internetwork variables to values that satisfy the interconnecting constraints. However, it may still take some iterations to obtain optimal values for the local variables. In particular, a higher c results in a higher number of iterations before reaching optimality.

3.2 Parallel Implementation

The parallel implementation is the result of the *Auxiliary Problem Principle* (Batut and Renaud, 1992; Kim and Baldick, 1997; Royo, 2001) of approximating the non-separable quadratic term in the augmented Lagrangian (6) with a separable term. In the parallel scheme, the agents do not wait for each other, but instead perform a series of parallel iterations in which all agents are computing at the same time.

Given the previous information $\tilde{w}_{\text{prev},k}^{ij} = \tilde{w}_{s|k}^{ij}$, and $\tilde{w}_{\text{prev},k}^{ji} = \tilde{w}_{s|k}^{ji}$ of the agents $j \in N_i$ of the last iteration $s - 1$, agent i solves problem (9) using the following additional objective function term for the interconnecting constraints:

$$J_w^i(\tilde{w}_k^{ji}) = \left[(\tilde{\lambda}_s^{ji})^T \quad (-\tilde{\lambda}_s^{ij})^T \right] \begin{bmatrix} \tilde{w}_{\text{in},k}^{ji} \\ \tilde{w}_{\text{out},k}^{ji} \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{w}_{\text{in},\text{prev},k}^{ji} \\ \tilde{w}_{\text{out},\text{prev},k}^{ji} \end{bmatrix} - \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}_{\text{in},k}^{ji} \\ \tilde{w}_{\text{out},k}^{ji} \end{bmatrix} \right\|^2 + \frac{b-c}{2} \left\| \begin{bmatrix} \tilde{w}_{\text{in},k}^{ji} \\ \tilde{w}_{\text{out},k}^{ji} \end{bmatrix} - \begin{bmatrix} \tilde{w}_{\text{in},\text{prev},k}^{ji} \\ \tilde{w}_{\text{out},\text{prev},k}^{ji} \end{bmatrix} \right\|^2.$$

Thus, this scheme uses information computed during the last iteration $s - 1$ only, contrarily to the serial scheme.

The parallel implementation then consists of the following steps. At decision step k , iteration s :

- (1) For all agents $i \in \{1, \dots, n\}$, *at the same*:
 - (a) Agent i solves the problem (9) to determine $\tilde{z}_{s+1|k}^i$ and $\tilde{w}_{s+1|k}^{ji}$ for all $j \in N_i$.
 - (b) Agent i sends to agent j the computed values $\tilde{w}_{s+1|k}^{ji}$ for all $j \in N_i$.
- (2) After all agents have finished solving their problem, they update their Lagrangian multipliers using (7)–(8) and the received information.
- (3) Each agent moves to the next iteration $s + 1$, and the cycle starts over, unless the Lagrangian multipliers do not change anymore.

The role of the scalar c is the same as before. As additional parameter this scheme uses a positive scalar b . If $b = c$ we obtain the same additional cost function

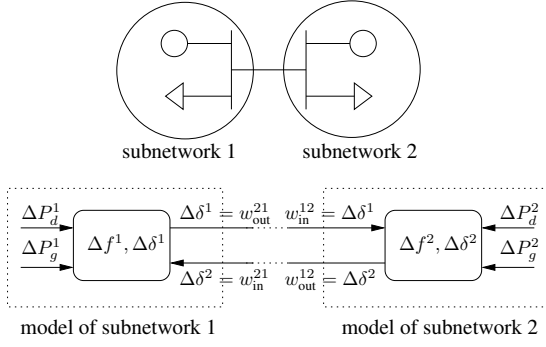


Fig. 3. Illustration of the physical network and the variables of the subnetwork models.

as in the serial scheme, but now based on information from the previous iteration alone. If $b > c$, then the term penalizes the deviation between the internetwork variables of the current iteration and the internetwork variables of the last iteration of agent i ; it thus gives the agent less incentive to change its internetwork variables from one iteration to the next. When $b \geq 2c$, and moreover the overall problem is convex, it can be proved that the iterations converge to the overall minimum (Bertsekas and Tsitsiklis, 1997; Kim and Baldick, 1997). In the following, we set $b = 2c$.

4. EXPERIMENTS

We experiment with the two schemes on a so-called *load-frequency* control problem from the domain of power networks, which involves keeping power generation equal to power consumption under consumption disturbances. We consider a network divided into two subnetworks, each controlled by a control agent, see Fig. 3. Each agent has to keep the so-called frequency deviation within its subnetwork close to zero under minimal control input accessing only local variables.

Each subnetwork $i \in \{1, 2\}$ has as control input the power generation, $u_k^i = [\Delta P_{g,k}^i]$, and as local disturbance the power consumption, $d_k^i = [\Delta P_{d,k}^i]$. The state of subnetwork i is $x_k^i = [\Delta \delta_k^i, \Delta f_k^i]^T$, where $\Delta \delta_k^i$ is the angle deviation and Δf_k^i is the frequency deviation. To keep Δf_k^i close to zero, generation should equal consumption and export over the interconnecting line. The power exported over the line from subnetwork i to j is proportional to $\Delta \delta^i - \Delta \delta^j$. Therefore, for subnetwork i , we have internetwork input variable $w_{in,k}^{ji} = [\Delta \delta_k^j]$, and internetwork output variable $w_{out,k}^{ji} = [\Delta \delta_k^i]$. The model of subnetwork i is a linearized discrete-time state-space model. This model is a Euler approximation (with a step-size of $\tau = 1$ s) of the continuous-time model described in (Camponogara *et al.*, 2002), and can be written as:

$$\begin{aligned} x_{k+1}^i &= A^i x_k^i + B_1^i u_k^i + B_2^i d_k^i + B_3^i w_{in,k}^{ji} \\ w_{out,k}^{ji} &= C^{ji} [(x_k^i)^T (u_k^i)^T (d_k^i)^T]^T, \end{aligned}$$

together with interconnecting constraints (4)–(5), where

c	serial scheme			parallel scheme		
	# iterat.	Δ costs	time sec.	# iterat.	Δ costs	time sec.
0.1	21	0.0004	<1	26	0.0004	<1
10	150	0.0005	1	467	0.0005	4
50	597	0.0005	5	1793	0.0008	15
100	1079	0.0005	9	3130	0.0013	26
200	1934	0.0006	16	5358	0.0030	45
300	2707	0.0007	23	7250	0.0053	61

Table 1. For serial and parallel schemes, for varying c , the number of iterations required, additional costs of found solution, and total time in seconds required.

$$\begin{aligned} A^i &= \begin{bmatrix} 1 & \tau 2\pi \\ \tau \frac{-K_{P^i} K_{S_{ij}}}{2\pi T_{P^i}} & 1 - \tau \frac{1}{T_{P^i}} \end{bmatrix} & B_1^i &= \begin{bmatrix} 0 \\ \tau \frac{K_{P^i}}{T_{P^i}} \end{bmatrix} \\ B_2^i &= \begin{bmatrix} 0 \\ -\tau \frac{K_{P^i}}{T_{P^i}} \end{bmatrix} & B_3^i &= \begin{bmatrix} 0 \\ \tau \frac{K_{P^i} K_{S_{ij}}}{2\pi T_{P^i}} \end{bmatrix}, \end{aligned}$$

and $C^{ji} = [1 \ 0 \ 0 \ 0]$, $K_{P^1} = 1125$, $K_{P^2} = 120$, $K_{S_{12}} = K_{S_{21}} = 0.5$, $T_{P^1} = 25$, $T_{P^2} = 20$.

Agent i uses the following local objective function to minimize frequency deviation and control input:

$$J_z^i(z_k^i) = \sum_{p=0}^{N-1} q_{\Delta f}^i (\Delta f_{k+p+1}^i)^2 + q_{\Delta P_g}^i (\Delta P_{g,k+p}^i)^2,$$

where $q_{\Delta f}^i = 100$ and $q_{\Delta P_g}^i = 10$ are weight factors.

In the simulations the two subnetworks are in steady-state, until a consumption disturbance of $\Delta P_{d,k}^2 = 0.5$ occurs in subnetwork 2. At that time the dynamics of the two subnetworks become highly dependent on one another, and the agents cannot make adequate predictions on the evolution of their own subnetworks without communication. We consider one agreement making step between the agents, right after the disturbance has taken place. The agents have to obtain agreement on the values of their internetwork variables, i.e., $\Delta \delta^1$ and $\Delta \delta^2$ over the horizon. We compare the solution quality of the serial and parallel scheme with a centralized scheme, and moreover examine the influence of the c parameter.

The third and sixth columns of Table 1 show for the serial and parallel scheme the difference between the optimal costs of the distributed scheme and the centralized scheme. Since the optimization problem turns out to be convex, the introduced methods should converge to the optimal solution. The table shows that this is indeed the case, although the solutions that both schemes find have increasingly higher costs than the centralized optimal solution for increasing c . The serial scheme shows much slower deviation from the optimal cost than the parallel scheme.

Fig. 4 shows the convergence of the difference between two internetwork variables in one of the interconnecting constraints under varying c . We see that when the value of c increases, the difference converges

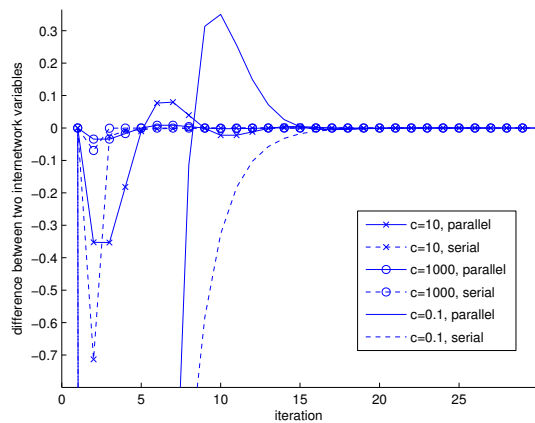


Fig. 4. Comparison of convergence of serial and parallel implementation for different values of c .

to zero faster for both the serial and the parallel approach. This is because with higher c the internetwork variables are pushed more to attain equal values. This means that the interconnecting constraints are satisfied after fewer iterations with higher c .

However, simply having satisfied the interconnecting constraints does not imply that the values computed for the local variables are optimal under these internetwork variables. Even when internetwork variables have been found that satisfy the interconnecting constraints, some iterations will be necessary to find the optimal local variables. The second and fifth columns in Table 1 show the number of iterations required to come to optimality. We observe that the serial approach requires fewer iterations than the parallel approach under any c . Also in terms of seconds, the serial approach outperforms the parallel approach, as we see in the fourth and seventh columns in Table 1. These preferable characteristics of the serial approach are due to the fact that the serial approach includes information from neighboring agents directly when it comes available, instead of from the next iteration.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have considered multi-agent Model Predictive Control for the control of large-scale transportation networks, like road traffic networks, power distribution networks, etc. In particular, we have considered a serial and a parallel scheme for agents to obtain agreement on the evolution of the internetwork variables. Based on a generic transportation-network model, we have discussed these schemes theoretically and experimentally.

Although the parallel scheme is more frequently used throughout the literature, we have not found advantages that make this scheme in particular better than the serial scheme. In fact, our numerical experiments show that the serial scheme has more preferable features in terms of solution speed and solution quality.

Future research consists of extending the methods to situations in which the problem of controlling the transportation network cannot be formulated as a convex problem. In particular we will extend the methods to deal with networks modeled as hybrid systems in which both continuous and discrete dynamics appear.

ACKNOWLEDGMENTS

This research was supported by the project “HYbrid CONTROL: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)”, contract number FP6-IST-511368, the project “Multi-agent control of large-scale hybrid systems” (DWV.6188) of the Dutch Technology Foundation STW, and the BSIK project “Next Generation Infrastructures (NGI)”.

REFERENCES

- Batut, J. and A. Renaud (1992). Daily generation scheduling optimization with transmission constraints: a new class of algorithms. *IEEE Transactions on Power Systems* **7**(3), 982–989.
- Bertsekas, D. P. and J. N. Tsitsiklis (1997). *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Nashua, New Hampshire.
- Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press, Cambridge, United Kingdom.
- Camponogara, E., D. Jia, B. H. Krogh and S. Talukdar (2002). Distributed model predictive control. *IEEE Control Systems Magazine* **1**, 44–52.
- De Schutter, B., T. van den Boom and A. Hegyi (2002). A model predictive control approach for recovery from delays in railway systems. *Transportation Research Record* (1793), 15–20.
- Dunbar, W. B. and R. M. Murray (2002). Model predictive control of coordinated multi-vehicle formations. In: *Proceedings of the 41st IEEE Conference on Decision and Control*. Las Vegas, Nevada, pp. 4631–4636.
- Hegyi, A., B. De Schutter and J. Hellendoorn (2005). Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems* **6**(1), 102–112.
- Hines, P., L. Huaiwei, D. Jia and S. Talukdar (2005). Autonomous agents and cooperation for the control of cascading failures in electric grids. In: *Proceedings of the 2005 IEEE International Conference on Networking, Sensing and Control*. Tucson, Arizona, pp. 273–278.
- Kim, B. H. and R. Baldick (1997). Coarse-grained distributed optimal power flow. *IEEE Transactions on Power Systems* **12**(2), 932–939.
- Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, England.
- Morari, M. and J. H. Lee (1999). Model predictive control: past, present and future. *Computers and Chemical Engineering* **23**, 667–682.
- Royo, C. B. (2001). Generalized Unit Commitment by the Radar Multiplier Method. PhD thesis. Technical University of Catalonia, Barcelona, Spain.
- Siljak, D. D. (1991). *Decentralized Control of Complex Systems*. Academic Press, Boston.