Multi-Agent Model Predictive Control

with Applications to Power Networks

R.R. Negenborn

Multi-Agent Model Predictive Control

with Applications to Power Networks

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College van Promoties,
in het openbaar te verdedigen op dinsdag 18 december 2007 om 10:00 uur
door Rudy Rafaël NEGENBORN,
doctorandus in de informatica,
geboren te Utrecht.

Dit proefschrift is goedgekeurd door de promotoren:
Prof.dr.ir. J. Hellendoorn
Prof.dr.ir. B. De Schutter

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. J. Hellendoorn	Technische Universiteit Delft, promotor
Prof.dr.ir. B. De Schutter	Technische Universiteit Delft, promotor
Prof.dr. G.J. Olsder	Technische Universiteit Delft
Prof.dr. J.-J.Ch. Meyer	Universiteit Utrecht
Prof.Dr. G. Andersson	ETH Zürich
Prof.Dr.-Ing. W. Marquardt	RWTH Aachen University
Ir. J.J.M. Langedijk	Siemens Nederland N.V.
Prof.dr. C. Witteveen	Technische Universiteit Delft (reservelid)

This thesis has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate studies. The research described in this thesis was supported by the project "Multi-agent control of large-scale hybrid systems" (DWV.6188) of the Dutch Technology Foundation STW and by an NWO Van Gogh grant (VGP79-99).

TRAIL Thesis Series T2007/14, The Netherlands TRAIL Research School

Published and distributed by: R.R. Negenborn
E-mail: rudy@negenborn.net
WWW: http://www.negenborn.net/mampc/

ISBN 978-90-5584-093-9

Keywords: multi-agent control, model predictive control, power networks, transportation networks.

Printed in The Netherlands

Contents

Chapter 2

Serial versus parallel schemes

In this chapter we consider multi-agent single-layer MPC, in which the network is divided into several non-overlapping subnetworks, and each subnetwork is controlled by one control agent, as shown in Figure 1.5. The agents have to locally choose those actions that give an overall optimal performance. In Section 2.1 we introduce the assumptions that we make on the network and control structure. In Section 2.2 we then formulate the MPC problem considering only one particular control agent, assuming that it knows how the surrounding network behaves. In Section 2.3 we relax this assumption and discuss how interconnections between control problems of different agents are formalized and how the multi-agent single-layer MPC approaches can differ in dealing with these interconnections. In Section 2.4 we focus on particular types of schemes, viz. *synchronous, multi-iteration, parallel*, and *serial* schemes. We propose a novel serial scheme based on Lagrange theory, and compare this scheme with a related parallel scheme. In Section 2.5 we propose the application of the approaches to the load-frequency control problem of power networks. A benchmark network is defined and through experimental simulation studies on this network we illustrate the performance of the parallel and the serial scheme.

Parts of this chapter have been published in [89, 107, 109] and presented in [112].

2.1 Network and control setup

2.1.1 Network dynamics

As discussed in Chapter 1, transportation networks are large-scale systems with complex dynamics. In order to analyze them, assumptions have to be made on the dynamics, i.e., on the way the networks behave. Therefore, assume a network that is divided into n subnetworks, where each subnetwork consists of a set of nodes and the interconnections between these nodes. Assume furthermore that the dynamics of subnetwork $i \in \{1,\ldots,n\}$ are given by a deterministic linear discrete-time time-invariant model (possibly obtained after symbolic or numerical linearization of a nonlinear model in combination with discretization):

$$\mathbf{x}_i(k+1) = \mathbf{A}_i\mathbf{x}_i(k) + \mathbf{B}_{1,i}\mathbf{u}_i(k) + \mathbf{B}_{2,i}\mathbf{d}_i(k) + \mathbf{B}_{3,i}\mathbf{v}_i(k) \qquad (2.1)$$

$$\mathbf{y}_i(k) = \mathbf{C}_i\mathbf{x}_i(k) + \mathbf{D}_{1,i}\mathbf{u}_i(k) + \mathbf{D}_{2,i}\mathbf{d}_i(k) + \mathbf{D}_{3,i}\mathbf{v}_i(k), \qquad (2.2)$$

Figure 2.1: From continuous time to discrete time.

where at time step k, for subnetwork i, $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathbf{x}_i}}$ are the local states, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathbf{u}_i}}$ are the local inputs, $\mathbf{d}_i(k) \in \mathbb{R}^{n_{\mathbf{d}_i}}$ are the local known exogenous inputs, $\mathbf{y}_i(k) \in \mathbb{R}^{n_{\mathbf{y}_i}}$ are the local outputs, $\mathbf{v}_i(k) \in \mathbb{R}^{n_{\mathbf{v}}}$ are the remaining variables influencing the local dynamical states and outputs, and $\mathbf{A}_i \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{x}_i}}$, $\mathbf{B}_{1,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{u}_i}}$, $\mathbf{B}_{2,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{d}_i}}$, $\mathbf{B}_{3,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{v}_i}}$, $\mathbf{C}_i \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{x}_i}}$, $\mathbf{D}_{1,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{u}_i}}$, $\mathbf{D}_{2,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{d}_i}}$, $\mathbf{D}_{3,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{v}_i}}$ determine how the different variables influence the local states and outputs of subnetwork i. The $\mathbf{v}_i(k)$ variables appear due to the fact that a subnetwork is connected to other subnetworks. Hence, the $\mathbf{v}_i(k)$ variables represent the influence of other subnetworks on subnetwork i. If the values of $\mathbf{v}_i(k)$ are fixed, then the dynamics of subnetwork i are decoupled from the other subnetworks.

Remark 2.1 For completeness inputs $\mathbf{u}_i(k)$ are also allowed to influence outputs $\mathbf{y}_i(k)$ at time k. A situation in which such direct feed-through terms typically appear is when algebraic relations are linearized, e.g., when linearizing equations describing instantaneous (power) flow distributions. □

Remark 2.2 In the subnetwork description that we consider here, all variables involved take on values in the real domain. This assumes that no discrete inputs, due to, e.g., switches, are present. In addition, in the subnetwork description that we consider here, the dynamics are assumed linear. Therefore, discrete behavior, e.g., due to saturation or discrete logic, cannot be included. In Chapter 3 we discuss issues related to including such discrete elements. □

Remark 2.3 In general the dynamics of the networks take place in continuous time. For computational reasons, however, it is convenient to assume that the continuous-time dynamics are adequately represented by discrete-time dynamics. Hence, instead of specifying and computing the dynamics of the network for each continuous-time instant $t \in [0, \infty)$, the dynamics are only specified and computed at discrete time or control cycle steps k, each representing T continuous-time time units, as shown in Figure 2.1. In Chapter 4 we discuss issues related to dealing with continuous-time dynamics in more detail. □

Remark 2.4 In general, the dynamics of the subnetworks are nonlinear. In Chapter 4 we discuss in more detail how to obtain linear models from nonlinear models by linearization. □

2.1.2 Control structure

We consider a multi-agent single-layer control structure as introduced in Section 1.3.2. Each of the subnetworks $i \in \{1, \ldots, n\}$ is controlled by a control agent i that:

- has a prediction model M_i of the dynamics of subnetwork i that matches the subnetwork dynamics given by (2.1)–(2.2);

- can measure the state $\mathbf{x}_i(k)$ of its subnetwork;
- can determine settings $\mathbf{u}_i(k)$ for the actuators of its subnetwork;
- can obtain exogenous inputs $\mathbf{d}_i(k+l)$ of its subnetwork over a certain horizon of length N, for $l = \{0,\ldots,N\}$;
- can communicate with neighboring agents, i.e., the agents controlling the subnetworks $j \in \mathcal{N}_i$, where $\mathcal{N}_i = \{j_{i,1},\ldots,j_{i,m_i}\}$ is the set of indexes of the m_i subnetworks connected to subnetwork i, also referred to as the *neighbors* of subnetwork or agent i.

Remark 2.5 The agents have no authority relations over one another, i.e., there is no agent that can force another agent to do something, and each agent has only information about its own subnetwork. In Chapter 4 we discuss how supervisory agents that can steer or direct other agents can be used. □

Remark 2.6 The multi-agent control structure studied here may be used not only for control of networks that span large geographical areas, but also for control of relatively small networks, when restrictions on acting and sensing make single-agent control impossible. □

2.2 MPC of a single subnetwork

Assume for now that the control agent of subnetwork i operates individually, that it therefore does not communicate with other agents, and that it knows how the surrounding network behaves. Below we will relax these assumptions.

The control agent employs MPC to determine which actions to take. In MPC, an agent determines its actions by computing optimal actions over a prediction horizon of N control cycles according to an objective function, subject to a model of the subnetwork, the behavior of the surrounding network, and additional constraints.

The MPC strategy of agent i at time k consists of measuring the initial local state[1] $\bar{\mathbf{x}}_i(k)$, determining local exogenous inputs over the horizon $\bar{\mathbf{d}}_i(k+l)$, for $l = \{0,\ldots,N-1\}$, and predicting influences of the rest of the network over the prediction horizon $\bar{\mathbf{v}}_i(k+l)$, for $l = \{0,\ldots,N-1\}$. Here, for notational convenience, the bar over variables indicates that the values of these variables are known. In addition, below the tilde over variables is used to denote variables over the prediction horizon, e.g., $\tilde{\mathbf{a}}_i(k) = [\,\mathbf{a}_i(k)^{\mathrm{T}},\ldots,\mathbf{a}_i(k+N-1)^{\mathrm{T}}\,]^{\mathrm{T}}$. Control agent i then solves the following optimization problem:

$$\min_{\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k)} J_{\mathrm{local},i}\left(\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k)\right) = \sum_{l=0}^{N-1} J_{\mathrm{stage},i}(\mathbf{x}_i(k+1+l),\mathbf{u}_i(k+l),\mathbf{y}_i(k+l)) \tag{2.3}$$

[1] The measured initial local state is in this case the exact initial local state, since no measurement noise is considered.

subject to

$$\mathbf{x}_i(k+1+l) = \mathbf{A}_i\mathbf{x}_i(k+l) + \mathbf{B}_{1,i}\mathbf{u}_i(k+l) + \mathbf{B}_{2,i}\bar{\mathbf{d}}_i(k+l) + \mathbf{B}_{3,i}\mathbf{v}_i(k+l) \tag{2.4}$$

$$\mathbf{y}_i(k+l) = \mathbf{C}_i\mathbf{x}_i(k+l) + \mathbf{D}_{1,i}\mathbf{u}_i(k+l) + \mathbf{D}_{2,i}\bar{\mathbf{d}}_i(k+l) + \mathbf{D}_{3,i}\mathbf{v}_i(k+l) \tag{2.5}$$

$$\mathbf{v}_i(k+l) = \bar{\mathbf{v}}_i(k+l) \tag{2.6}$$

$$\text{for } l = 0,\dots,N-1$$

$$\mathbf{x}_i(k) = \bar{\mathbf{x}}_i(k), \tag{2.7}$$

where $J_{\text{stage},i}$ is a twice differentiable function that gives the cost per prediction step given a certain local state, local input, and local output. A typical choice for the stage cost is:

$$J_{\text{stage},i}\left(\mathbf{x}_i(k+1), \mathbf{u}_i(k), \mathbf{y}_i(k)\right) = \begin{bmatrix} \mathbf{x}_i(k+1) \\ \mathbf{u}_i(k) \\ \mathbf{y}_i(k) \end{bmatrix}^{\mathrm{T}} \mathbf{Q}_i \begin{bmatrix} \mathbf{x}_i(k+1) \\ \mathbf{u}_i(k) \\ \mathbf{y}_i(k) \end{bmatrix} + \mathbf{f}_i^{\mathrm{T}} \begin{bmatrix} \mathbf{x}_i(k+1) \\ \mathbf{u}_i(k) \\ \mathbf{y}_i(k) \end{bmatrix}, \tag{2.8}$$

where $\mathbf{Q}_i$ and $\mathbf{f}_i$ are a positive definite weighting matrix and a vector, respectively. After control agent i has solved the optimization problem and found the N actions over the horizon, it implements the actions $\mathbf{u}_i(k)$ until the next control cycle, the control cycle k moves to $k+1$, and the control agent performs the MPC strategy at that control cycle by setting up and solving the MPC optimization problem for $k+1$.

We have assumed here through (2.6) that the agent does not use communication and that it can by itself locally predict the influence of the surrounding network over the prediction horizon, i.e., it knows $\mathbf{v}_i(k+l)$, for $l = 0,\dots,N-1$. However, control agent i cannot know this influence *a priori*, since actions taken by control agent i influence the dynamics of its own subnetwork and therefore also the dynamics of a neighboring subnetwork $j \in \mathcal{N}_i$, which therefore changes the decision making of neighboring agent j and, hence, changes the actions that control agent j chooses, which change the dynamics of subnetwork j, and thus changes $\mathbf{v}_i(k+l)$. Therefore, (2.6) cannot be added explicitly. To relax the assumption that this is possible, constraints between control problems and communication between control agents has to be used. Below we discuss this in more detail.

2.3 Interconnected control problems

The interconnections between control problems are modeled using so-called *interconnecting variables*. A particular variable of the control problem of agent i is an interconnecting variable with respect to the control problem of agent j if the variable of agent i corresponds to the same physical quantity as a variable in the control problem of agent j. E.g., a flow going from subnetwork i into subnetwork j is represented with an interconnecting variable in the control problems of both agents.

Given the interconnecting variables of two agents corresponding to the same quantity, it is convenient to define one of these variables as an interconnecting *input* variable and the other as an interconnecting *output* variable. On the one hand, interconnecting input variables $\mathbf{w}_{\text{in},ji}(k)$ of the control problem of agent i with respect to agent j at control cycle k can be seen as inputs caused by agent j on the control problem of agent i. On the other hand, interconnecting output variables $\mathbf{w}_{\text{out},ij}(k)$ of the control problem of agent j with

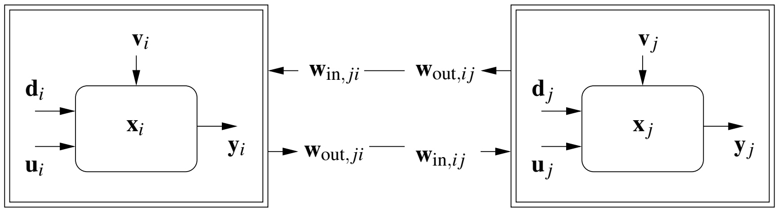


Figure 2.2: Illustration of the relation between the models and interconnecting variables of control agents i and j.

respect to the control problem of agent i can be seen as the influence that agent j has on the control problem of agent i. Figure 2.2 illustrates this. We consider interconnecting variables $\mathbf{w}_{\text{in},ji}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{in},ji}}}$ and $\mathbf{w}_{\text{out},ji}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{out},ji}}}$.

Define the interconnecting inputs and outputs for the control problem of agent i over a prediction horizon at control cycle k as:

$$\tilde{\mathbf{w}}_{\text{in},i}(k) = \tilde{\mathbf{v}}_i(k) \tag{2.9}$$

$$\tilde{\mathbf{w}}_{\text{out},i}(k) = \tilde{\mathbf{K}}_i \left[\tilde{\mathbf{x}}_i(k+1)^{\text{T}} \quad \tilde{\mathbf{u}}_i(k)^{\text{T}} \quad \tilde{\mathbf{y}}_i(k)^{\text{T}}\right]^{\text{T}}, \tag{2.10}$$

where $\tilde{\mathbf{K}}_i$ is an interconnecting output selection matrix that contains zeros everywhere, except for a single 1 per row corresponding to a local variable that relates to an interconnecting output variable.

The variables $\tilde{\mathbf{w}}_{\text{in},i}(k)$, $\tilde{\mathbf{w}}_{\text{out},i}(k)$ are partitioned such that:

$$\tilde{\mathbf{w}}_{\text{in},i}(k) = \left[\tilde{\mathbf{w}}_{\text{in},j_{i,1}i}(k)^{\text{T}}, \ldots, \tilde{\mathbf{w}}_{\text{in},j_{i,m_i}i}(k)^{\text{T}}\right]^{\text{T}} \tag{2.11}$$

$$\tilde{\mathbf{w}}_{\text{out},i}(k) = \left[\tilde{\mathbf{w}}_{\text{out},j_{i,1}i}(k)^{\text{T}}, \ldots, \tilde{\mathbf{w}}_{\text{out},j_{i,m_i}i}(k)^{\text{T}}\right]^{\text{T}}. \tag{2.12}$$

The interconnecting inputs to the control problem of agent i with respect to agent j must be equal to the interconnecting outputs from the control problem of agent j with respect to agent i, since the variables of both control problems model the same quantity. For agent i this thus gives rise to the following *interconnecting constraints*:

$$\tilde{\mathbf{w}}_{\text{in},ji}(k) = \tilde{\mathbf{w}}_{\text{out},ij}(k) \tag{2.13}$$

$$\tilde{\mathbf{w}}_{\text{out},ji}(k) = \tilde{\mathbf{w}}_{\text{in},ij}(k), \tag{2.14}$$

for all $j \in \mathcal{N}_i$.

An interconnecting constraint depends on variables of two different control agents. Therefore, a particular control agent will always miss information that it requires to include the interconnecting constraint explicitly in its MPC control problem formulation. Hence, the agent has to use communication with another agent to exchange information that it uses to determine which values it should give to the interconnecting inputs and outputs. Below, we survey how schemes for multi-agent single-layer MPC differ in the type of information exchanged and the moments at which information exchange takes place.

2.3.1 Types of information exchange

The challenge is to find a suitable way for the control agents to deal with the interconnecting variables $\tilde{\mathbf{w}}_{\text{in},ji}(k)$ and $\tilde{\mathbf{w}}_{\text{out},ji}(k)$. In order to make a prediction of the evolution of the subnetwork, values of the interconnecting variables have to be known or assumed over the prediction horizon. There are several approaches to dealing with the interconnecting variables, each yielding different types of information that is exchanged:

1. Ignore the influence of the interconnecting variables. This approach is used in a completely decentralized setting. A control agent ignores the presence of other subnetworks completely. This type of control scheme can be used when interconnecting variables have negligible effect on the subnetwork dynamics. An advantage of this approach is the absence of communication overhead. However, if the influence of the interconnecting variables turns out not to be negligible, control performance will degenerate.

2. Use constant values for the values of the interconnecting variables over the full prediction horizon based on a local measurement made or obtained from a neighboring agent. This approach may be useful when the interconnecting variables change slowly. This approach may also be used to monitor the interconnecting variables online and to switch to a different way of dealing with the interconnecting variables when the variables start changing significantly. An advantage of this approach is relatively fast control, since the control agents only exchange information at the beginning of each control cycle once and after that solve their control problems decentralized. A disadvantage of this approach is that if the values of the interconnecting variables exchanged at the beginning of a control cycle are not valid over the complete prediction horizon, the performance of the control will decrease.

3. Use predictions of the values of the interconnecting variables over the full prediction horizon as obtained from a neighboring agent [28, 48, 75]. An advantage of this approach is that there is only communication at the beginning of a control cycle, after which the control agents solve their control problems decentralized. However, the neighboring agent providing the predictions has to make sure that the predictions are correct. In practice, if the subnetwork of the neighboring agent relies on other neighboring subnetworks this will be difficult to ensure. Iterations as discussed below in Section 2.3.2 are then necessary.

4. Use upper and lower bounds on the values of the interconnecting variables, as obtained from a neighboring agent. This assumes that neighboring agents do not communicate exact trajectories, but instead bounds on the values of the interconnecting variables. By enforcing these bounds, an agent can compute worst-case optimal inputs. The agent providing the bounds also has to make sure that its actual trajectory stays within the bounds it has communicated. So-called compatibility constraints can do this for certain linear-time invariant systems [37]. Hence, an advantage of this approach is that control agents do not have to make accurate predictions of the values of interconnecting variables. However, the resulting control may be conservative, since the control agents determine worst-case optimal inputs. In addition, if a control agent requires accurate values for the interconnecting variables in order to make accurate

predictions of the evolution of its subnetwork, only using upper and lower bounds may give bad predictions, and consequently, bad performance.

5. Use a model that predicts the values of the interconnecting variables based on dynamics of neighboring subsystems [37]. When this is used a control agent knows the dynamics or part of the dynamics that generate the values of the interconnecting variables [37]. This is, e.g., the case when the local agent has a copy of the subnetwork models used by its neighbors. These models will depend on variables of the neighboring subnetworks, like inputs, and perhaps interconnecting variables of neighbors of neighbors. An advantage of this approach is that more about the interconnecting variables is known. A disadvantage of this approach can be increased computational time required to determine the predictions.

6. Use a model about the evolution of the interconnecting variables that has been learned given available information from neighboring agents. This approach can be employed if the agent does not have a model of the subnetwork that generates the interconnecting variables. Instead it may employ learning techniques and build up experience to learn a model. An advantage of this approach is that the control agent may exploit the model learned from experience to improve its performance. However, learning such a model in the first place is challenging.

7. Use knowledge about the objective function of neighboring agents together with models of the dynamics of the neighboring system [79]. The control agent can use this information to compute which actions the neighbors will take [79]. It can determine the actions that will be applied to that subsystem and consequently determine the evolution of the values of the interconnecting variables. Knowledge about the objectives of neighboring subnetworks can be used to make local decisions that are not counteracting the objectives of other control agents. Hence, an advantage of this approach is that a control agent can anticipate what other control agents are going to do and therefore possibly increase the efficiency of the decision making. A disadvantage of this approach is that one controller effectively is solving the control problems of multiple subnetworks. Hence, the computational requirements will increase significantly, even more than when approach 5 is used. In an approach that somehow communicates the computed actions to the neighboring subnetworks this could become an advantage however.

2.3.2 Timing of information exchange

Schemes for multi-agent MPC do not only differ in the type of information exchanged, but also in the moment at which information exchange takes place, as shown in Figure 2.3. The schemes are distinguished by the following characteristics:

1. *Synchronous* or *asynchronous*, i.e., do agents have to wait for one another when it comes to sending and receiving information and determining which actions to take, or can they send and receive information and determine which action to take at any time.

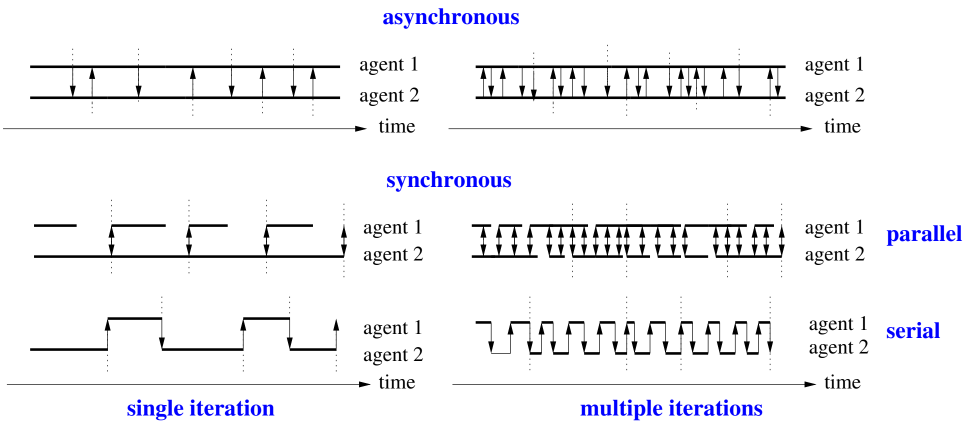


Figure 2.3: Different communication schemes between two agents. Arrows indicate information exchange. Dotted lines indicate actions being implemented. Horizontal lines indicate optimization problems being solved.

2. *Single* or *multiple* iterations, i.e., do agents decide on their actions after sending and receiving information once, or do agents decide on their actions after a number of information exchanges.

3. *Parallel* or *serial*, i.e., are multiple agents performing computations at the same time, or is there only one agent at a time performing its computations.

Asynchronous schemes have as advantage over synchronous schemes that agents do not have to wait for other agents to solve their problems and decide on which actions to take. However, agents will have to include newly received information from neighboring agents at any time while solving their own optimization problems. No multi-agent MPC methods can do this at present.

Single-iteration schemes have as advantage over multiple-iteration schemes that the amount of communication between agents is less, since information is exchanged only once after an agent has solved its problem, and that time required to make a decision is less, since only one iteration is done. Multiple-iteration schemes have as advantage over single-iteration schemes that it is more likely that interconnecting constraints are satisfied at the end of the iterations. In addition, over the iterations agents obtain implicit information about the objectives of their neighbors. Multiple-iteration schemes therefore have a larger potential to achieve overall optimal performance than single-iteration schemes.

Serial schemes have as advantage over parallel schemes that agents use the most up-to-date information from their neighbors. In parallel schemes, the information that is received is usually outdated. However, in serial schemes only one agent is performing computations at a time and therefore decision making is potentially slower than when a parallel scheme is used.

In the literature, several aspects of synchronous single-iteration parallel schemes have been considered, e.g., in [37, 75, 79]. For certain linear time-invariant systems stability can be proved when a so-called contracting stability constraint is placed on the first state of each subsystem [75]. Stability results for settings where the evolution of interconnecting variables does not depend on neighbors of neighbors are given in [37, 79]. Synchronous multiple-iterations serial and parallel schemes have been considered in [28, 48, 74]. Conditions for convergence of iterations to local solutions and global solutions are given in [28]. A Lagrange-based scheme for the parallel case is employed in [48].

In the following we relax the assumption made in Section 2.2 that the control agent operates individually and knows what the influence of the neighboring agents is going to be. We extend the scheme of Section 2.2 to take into account the neighbors through an iterative procedure. The procedure uses as information predictions over the full horizon as obtained from neighboring agents, and employs multiple iterations in a synchronous fashion, aiming for satisfaction of the interconnecting constraints.

2.4 Lagrange-based multi-agent single-layer MPC

For feasible overall solutions, the interconnecting constraints as defined in (2.13)–(2.14) have to be satisfied at the moment that control agents decide on which action to take. As discussed above, when one agent solves its optimization problem it has to assume trajectories for the interconnecting variables of its neighboring subnetworks over the horizon. If the neighboring control agents do not respect the assumed trajectories that they communicated, it is unlikely that such a trajectory will appear in the true system evolution. The neighboring control agents will only have an incentive to respect their communicated trajectories if these trajectories yield optimal inputs for their own subsystems.

Even if the agents make an agreement in advance to respect the trajectories communicated, in practice they may not be able to implement this agreement. The reason for this is that at the time of trajectory generation the agents did not know what the values of the interconnecting variables of the other agents will be. Therefore, they may require infeasible inputs to local subsystems to respect the communicated trajectories. To deal with this, a scheme can be used in which the agents perform a number of iterations to come to an agreement on interconnecting variable trajectories that are acceptable to all agents, instead of holding on to the first trajectories communicated.

In each iteration each agent optimizes both over its actions and over the predictions of trajectories of neighboring subnetworks. In this way, each agent is sure that the predicted trajectories it assumes are optimal for its own subsystem. After each of the agents has in this way determined its own optimal actions and predicted interconnecting variables trajectory, it communicates the predicted interconnecting variable trajectories to the neighboring agents. This basically means that each agent tells its neighboring agents how it would like to see the interconnecting variables of those agents evolve over the horizon.

Ideally, the interconnecting variable trajectories that those neighboring agents receive will exactly correspond to their predictions of their interconnecting variable trajectories if they would implement their optimal input sequences. However, it is more likely that the received trajectories will not correspond to their predicted trajectories, as discussed before. To encourage the agents to come to an agreement on the predicted interconnecting variable trajectories a penalty term is added to the objective function of each agent. By updating the penalty terms over a series of iterations using the information received from neighboring agents, convergence may be obtained under appropriate assumptions, as we will discuss below.

To derive a scheme that implements these ideas we consider the following steps:

1. Formulate the combined overall control problem, i.e., aggregate the subproblems including the interconnecting constraints;
2. Construct an augmented Lagrange formulation by replacing each interconnecting constraint with an additional linear cost term, based on Lagrange multipliers, and a quadratic penalty term [19, 23];
3. Decompose this formulation again into subproblems for each agent.

We now focus on these steps in more detail.

2.4.1 Combined overall control problem

We define the combined overall control problem as the problem formed by the aggregation of the local control problems without assuming that the influence from the rest of the network formulated through (2.6) is known, but including the definition of the interconnecting inputs and outputs (2.9)–(2.10) and the interconnecting constraints (2.13)–(2.14). After defining:

$$\begin{aligned}\tilde{\mathbf{X}}(k+1) &= [\tilde{\mathbf{x}}_1(k+1)^{\mathrm{T}}, \dots, \tilde{\mathbf{x}}_n(k+1)^{\mathrm{T}}]^{\mathrm{T}} \\ \tilde{\mathbf{U}}(k) &= [\tilde{\mathbf{u}}_1(k)^{\mathrm{T}}, \dots, \tilde{\mathbf{u}}_n(k)^{\mathrm{T}}]^{\mathrm{T}} \\ \tilde{\mathbf{Y}}(k) &= [\tilde{\mathbf{y}}_1(k)^{\mathrm{T}}, \dots, \tilde{\mathbf{y}}_n(k)^{\mathrm{T}}]^{\mathrm{T}},\end{aligned}$$

the control problem at control cycle k is defined as:

$$\min_{\tilde{\mathbf{X}}(k+1),\tilde{\mathbf{U}}(k),\tilde{\mathbf{Y}}(k)} \sum_{i=1}^{n} J_{\mathrm{local},i}(\tilde{\mathbf{x}}_i(k+1), \tilde{\mathbf{u}}_i(k), \tilde{\mathbf{y}}_i(k)) \tag{2.15}$$

subject to, for $i = 1, \dots, n$,

$$\tilde{\mathbf{w}}_{\mathrm{in},j_{i,1}i}(k) = \tilde{\mathbf{w}}_{\mathrm{out},ij_{i,1}}(k) \tag{2.16}$$

$$\vdots$$

$$\tilde{\mathbf{w}}_{\mathrm{in},j_{i,m_i}i}(k) = \tilde{\mathbf{w}}_{\mathrm{out},ij_{i,m_i}}(k) \tag{2.17}$$

and the dynamics (2.4)–(2.5) of subnetwork i over the horizon, and the initial constraint (2.7) of subnetwork i. Note that it is sufficient to include in the combined overall control problem formulation only the interconnecting input constraints (2.9) for each agent i, since the interconnecting output constraints (2.10) of agent i will also appear as interconnecting input constraints of its neighboring agents.

2.4.2 Augmented Lagrange formulation

The overall control problem (2.15) is not separable into subproblems using only local variables $\tilde{\mathbf{x}}_i(k+1)$, $\tilde{\mathbf{u}}_i(k)$, $\tilde{\mathbf{y}}_i(k)$ of one agent i alone due to the interconnecting constraints (2.16)–(2.17). In order to deal with the interconnecting constraints, an augmented Lagrange formulation of this problem can be formulated [19, 23]. An augmented Lagrange formulation combines a penalty formulation with a Lagrange formulation and therefore can provide improved convergence [18]. Using such a formulation, the interconnecting constraints are removed from the constraint set and added to the objective function in the form of additional linear cost terms, based on Lagrange multipliers, and additional quadratic terms. The augmented Lagrange function is defined as:

$$\begin{aligned} &L(\tilde{\mathbf{X}}(k+1), \tilde{\mathbf{U}}(k), \tilde{\mathbf{Y}}(k), \tilde{\mathbf{W}}_{\text{in}}(k), \tilde{\mathbf{W}}_{\text{out}}(k), \tilde{\mathbf{\Lambda}}_{\text{in}}(k)) \\ &= \sum_{i=1}^{n} \Bigg(J_{\text{local},i}\left(\tilde{\mathbf{x}}_i(k+1), \tilde{\mathbf{u}}_i(k), \tilde{\mathbf{y}}_i(k),\right) \\ &\qquad + \sum_{j \in \mathcal{N}_i} \left(\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k) \left(\tilde{\mathbf{w}}_{\text{in},ji}(k) - \tilde{\mathbf{w}}_{\text{out},ij}(k) \right) + \frac{\gamma_{\text{c}}}{2} \left\| \tilde{\mathbf{w}}_{\text{in},ji}(k) - \tilde{\mathbf{w}}_{\text{out},ij}(k) \right\|_2^2 \right) \Bigg), \end{aligned} \tag{2.18}$$

where

$$\begin{aligned} \tilde{\mathbf{W}}_{\text{in}}(k) &= [\tilde{\mathbf{w}}_{\text{in},j_{1,1}1}(k)^{\mathrm{T}}, \ldots, \tilde{\mathbf{w}}_{\text{in},j_{n,m_n}n}(k)^{\mathrm{T}}]^{\mathrm{T}} \\ \tilde{\mathbf{W}}_{\text{out}}(k) &= [\tilde{\mathbf{w}}_{\text{out},j_{1,1}1}(k)^{\mathrm{T}}, \ldots, \tilde{\mathbf{w}}_{\text{out},j_{n,m_n}n}(k)^{\mathrm{T}}]^{\mathrm{T}} \\ \tilde{\mathbf{\Lambda}}_{\text{in}}(k) &= [\tilde{\boldsymbol{\lambda}}_{\text{in},j_{1,1}1}(k)^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{\lambda}}_{\text{in},j_{n,m_n}n}(k)^{\mathrm{T}}]^{\mathrm{T}}, \end{aligned}$$

and where γ_{c} is a positive constant, and $\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)$ are the Lagrange multipliers associated with the interconnecting constraints $\tilde{\mathbf{w}}_{\text{in},ji}(k) = \tilde{\mathbf{w}}_{\text{out},ji}(k)$.

By duality theory [19, 23], the resulting optimization problem follows as maximization over the Lagrange multipliers while minimizing over the other variables, i.e.:

$$\max_{\tilde{\mathbf{\Lambda}}_{\text{in}}(k)} \left\{ \min_{\substack{\tilde{\mathbf{X}}(k+1), \tilde{\mathbf{U}}(k), \tilde{\mathbf{Y}}(k), \\ \tilde{\mathbf{W}}_{\text{in}}(k), \tilde{\mathbf{W}}_{\text{out}}(k)}} L\left(\tilde{\mathbf{X}}(k+1), \tilde{\mathbf{U}}(k), \tilde{\mathbf{Y}}(k), \tilde{\mathbf{W}}_{\text{in}}(k), \tilde{\mathbf{W}}_{\text{out}}(k), \tilde{\mathbf{\Lambda}}_{\text{in}}(k)\right) \right\}, \tag{2.19}$$

subject to the dynamics (2.4)–(2.5) of subnetwork i over the horizon, and the initial constraint (2.7) of subnetwork i, for $i = 1, \ldots, n$.

Under convexity assumptions on the objective functions and affinity of the subnetwork model constraints it can be proved that a minimum of the original problem (2.15) can be found iteratively by repeatedly solving the minimization part of (2.19) for fixed Lagrange multipliers, followed by updating the Lagrange multipliers using the solution of the minimization, until the Lagrange multipliers do not change anymore from one iteration to the next [19]. These convexity assumptions are satisfied for the linear model (2.1)–(2.2) that we assume, in combination with a linear local objective function, or in combination with a quadratic local objective function as defined in (2.8). In Section 2.5 we show an example of such a model with a quadratic local objective function.

2.4.3 Distributing the solution approach

The iterations to compute the solution of the combined overall control problem based on the augmented Lagrange formulation (2.18) include quadratic terms and can therefore not directly be distributed over the agents. To deal with this, the non-separable problem (2.18) can be approximated by solving n separated problems, each of which is based on local dynamics, local objectives $J_{\text{local},i}$, and an additional cost term $J_{\text{inter},i}$. The problem for the control agent controlling subnetwork i is defined as follows:

$$\min_{\substack{\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k),\\ \tilde{\mathbf{w}}_{\text{in},j_{i,1}i}(k),\ldots,\tilde{\mathbf{w}}_{\text{in},j_{i,m_i}i}(k),\\ \tilde{\mathbf{w}}_{\text{out},j_{i,1}i}(k),\ldots,\tilde{\mathbf{w}}_{\text{out},j_{i,m_i}i}(k)}} J_{\text{local},i}\left(\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k)\right) + \sum_{j\in\mathcal{N}_i} J_{\text{inter},i}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k),\tilde{\mathbf{w}}_{\text{out},ji}(k),\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)},\tilde{\boldsymbol{\lambda}}_{\text{out},ij}(k)^{(s)}\right), \tag{2.20}$$

subject to the dynamics (2.4)–(2.5) of subnetwork i over the horizon, and the initial constraint (2.7) of subnetwork i. As we will see below, the structure of the additional cost term $J_{\text{inter},i}$ differs depending on the type of communication scheme used. At iteration s, the variables $\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)}$ are the Lagrange multipliers computed by agent i for its interconnecting constraints $\tilde{\mathbf{w}}_{\text{in},ji}(k)=\tilde{\mathbf{w}}_{\text{out},ij}(k)$, and the variables $\tilde{\boldsymbol{\lambda}}_{\text{out},ij}(k)^{(s)}$ are the Lagrange multipliers for its interconnecting constraints $\tilde{\mathbf{w}}_{\text{out},ji}(k)=\tilde{\mathbf{w}}_{\text{in},ij}(k)$. The $\tilde{\boldsymbol{\lambda}}_{\text{out},ij}(k)^{(s)}$ variables are received by agent i through communication with agent j, which computed these variables for its interconnecting constraints with respect to agent i. The general multi-agent MPC scheme that results from this comprises at control cycle k the following steps:

1. For $i=1,\ldots,n$, agent i makes a measurement of the current state of the subnetwork $\bar{\mathbf{x}}_i(k)=\mathbf{x}(k)$ and estimates the expected exogenous inputs $\bar{\mathbf{d}}_i(k+l)$, for $l=0,\ldots,N-1$.
2. The agents cooperatively solve their control problems in the following iterative way:
   (a) Set the iteration counter s to 1 and initialize the Lagrange multipliers $\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)}$, $\tilde{\mathbf{w}}_{\text{out},ij}(k)^{(s)}$ arbitrarily.
   (b) Either serially or in parallel, for $i=1,\ldots,n$, agent i determines $\tilde{\mathbf{x}}_i(k+1)^{(s)}$, $\tilde{\mathbf{u}}_i(k)^{(s)}$, $\tilde{\mathbf{w}}_{\text{in},ji}(k)^{(s)}$, $\tilde{\mathbf{w}}_{\text{out},ij}(k)^{(s)}$, for $j\in\mathcal{N}_i$, by solving:

$$\min_{\substack{\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k),\\ \tilde{\mathbf{w}}_{\text{in},j_{i,1}i}(k),\ldots,\tilde{\mathbf{w}}_{\text{in},j_{i,m_i}i}(k),\\ \tilde{\mathbf{w}}_{\text{out},j_{i,1}i}(k),\ldots,\tilde{\mathbf{w}}_{\text{out},j_{i,m_i}i}(k)}} J_{\text{local},i}\left(\tilde{\mathbf{x}}_i(k+1),\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{y}}_i(k)\right) + \sum_{j\in\mathcal{N}_i} J_{\text{inter},i}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k),\tilde{\mathbf{w}}_{\text{out},ji}(k),\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)},\tilde{\boldsymbol{\lambda}}_{\text{out},ij}(k)^{(s)}\right), \tag{2.21}$$

   subject to the local dynamics (2.4)–(2.5) of subnetwork i over the horizon and the initial constraint (2.7) of subnetwork i.
   (c) Update the Lagrange multipliers,

$$\tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s+1)} = \tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)} + \gamma_{\text{c}}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k)^{(s)} - \tilde{\mathbf{w}}_{\text{out},ij}(k)^{(s)}\right). \tag{2.22}$$

   (d) Move on to the next iteration $s+1$ and repeat steps 2.(b)–2.(c). The iterations stop when the following stopping condition is satisfied:

$$\left\| \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{\text{in},j_{1,1}1}(k)^{(s+1)} - \tilde{\boldsymbol{\lambda}}_{\text{in},j_{1,1}1}(k)^{(s)} \\ \vdots \\ \tilde{\boldsymbol{\lambda}}_{\text{in},j_{n,m_n}n}(k)^{(s+1)} - \tilde{\boldsymbol{\lambda}}_{\text{in},j_{n,m_n}n}(k)^{(s)} \end{bmatrix} \right\|_\infty \leq \gamma_{\epsilon,\text{term}}, \tag{2.23}$$

where $\gamma_{\epsilon,\mathrm{term}}$ is a small positive scalar and $\|\cdot\|_\infty$ denotes the infinity norm. Note that satisfaction of this stopping condition can be determined in a distributed way, because each individual component of the infinity norm depends only on variables of one particular agent [111].

3. The agents implement the actions until the beginning of the next control cycle.

4. The next control cycle is started.

Remark 2.7 The Lagrange multipliers can be initialized arbitrarily; however, initializing them with values close to the optimal Lagrange multipliers will increase the convergence of the decision making process. Therefore, also initializing the Lagrange multipliers with values obtained from the previous decision-making step is beneficial, since typically these Lagrange multipliers will be good initial guesses for the new solution. We refer to this as a *warm start*. □

The schemes proposed in the literature implement step 2.(b) in a parallel fashion, e.g., [28, 41, 48]. In the following we first discuss a scheme that implements step 2.(b) in a parallel fashion and then we propose a novel scheme that implements it in a serial fashion. We then assess the performance of both schemes experimentally.

2.4.4 Serial versus parallel schemes

Parallel implementation

The parallel implementation is the result of using the *auxiliary problem principle* [14, 81, 127] of approximating the non-separable quadratic term in the augmented Lagrange formulation of the combined overall control problem. The parallel scheme involves a number of parallel iterations in which all agents perform their local computing step at the same time.

Given for the agents $j \in \mathcal{N}_i$, the previous information $\tilde{\mathbf{w}}_{\mathrm{in,prev},ij}(k) = \tilde{\mathbf{w}}_{\mathrm{in},ij}(k)^{(s-1)}$ and $\tilde{\mathbf{w}}_{\mathrm{out,prev},ji}(k) = \tilde{\mathbf{w}}_{\mathrm{out},ji}(k)^{(s-1)}$ *of the last iteration* $s-1$, agent i solves problem (2.21) using the following additional objective function term for the interconnecting constraints:

$$
\begin{aligned}
&J_{\mathrm{inter},i}\left(\tilde{\mathbf{w}}_{\mathrm{in},ji}(k), \tilde{\mathbf{w}}_{\mathrm{out},ji}(k), \tilde{\boldsymbol{\lambda}}_{\mathrm{in},ji}(k)^{(s)}, \tilde{\boldsymbol{\lambda}}_{\mathrm{out},ij}(k)^{(s)}\right) \\
&= \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{\mathrm{in},ji}(k)^{(s)} \\ -\tilde{\boldsymbol{\lambda}}_{\mathrm{out},ij}(k)^{(s)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\mathbf{w}}_{\mathrm{in},ji}(k) \\ \tilde{\mathbf{w}}_{\mathrm{out},ji}(k) \end{bmatrix} + \frac{\gamma_{\mathrm{c}}}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\mathrm{in,prev},ij}(k) - \tilde{\mathbf{w}}_{\mathrm{out},ji}(k) \\ \tilde{\mathbf{w}}_{\mathrm{out,prev},ij}(k) - \tilde{\mathbf{w}}_{\mathrm{in},ji}(k) \end{bmatrix} \right\|_2^2 \\
&\qquad + \frac{\gamma_{\mathrm{b}} - \gamma_{\mathrm{c}}}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\mathrm{in},ji}(k) - \tilde{\mathbf{w}}_{\mathrm{in,prev},ji}(k) \\ \tilde{\mathbf{w}}_{\mathrm{out},ji}(k) - \tilde{\mathbf{w}}_{\mathrm{out,prev},ji}(k) \end{bmatrix} \right\|_2^2 .
\end{aligned}
$$

This scheme uses only information computed during the last iteration $s-1$. The parallel implementation of step 2.(b) of the general multi-agent MPC scheme therefore consists of the following steps at decision step k, iteration s:

2 (b) For all agents $i \in \{1, \ldots, n\}$, *at the same time*, agent i solves the problem (2.21) to determine $\tilde{\mathbf{x}}_i(k+1)^{(s)}, \tilde{\mathbf{u}}_i(k)^{(s)}$, $\tilde{\mathbf{w}}_{\mathrm{in},ji}(k)^{(s)}$, $\tilde{\mathbf{w}}_{\mathrm{out},ji}(k)^{(s)}$, and sends to agent $j \in \mathcal{N}_i$ the computed values $\tilde{\mathbf{w}}_{\mathrm{out},ji}(k)^{(s)}$.

The positive scalar γ_c penalizes the deviation from the interconnecting variable iterates that were computed during the last iteration. This causes that when γ_c is chosen larger, the interconnecting variables that agent i computes at the current iteration will stay close to the interconnecting variables that neighboring agent $j \in \mathcal{N}_i$ computed earlier. With increasing γ_c, it becomes more expensive for an agent to deviate from the interconnecting-variable values computed by the other agents. This results in a faster convergence of the interconnecting variables to values that satisfy the interconnecting constraints. However, it may still take some iterations to obtain optimal values for the local variables. A higher γ_c results in a higher number of iterations before reaching optimality, although the interconnecting constraints will be satisfied quickly. A lower γ_c results in a lower number of iterations before reaching optimality and interconnecting constraints that are satisfied. However, when γ_c is chosen too small, a larger number of iterations will again be necessary, since it will take a longer time for the interconnecting constraints to be satisfied.

As additional parameter this scheme uses a positive scalar γ_b. If $\gamma_b > \gamma_c$, then the term penalizes the deviation between the interconnecting variables of the current iteration and the interconnecting variables of the last iteration of agent i; it thus gives the agent less incentive to change its interconnecting variables from one iteration to the next. When $\gamma_b \geq 2\gamma_c$, and moreover the overall combined problem is convex, it can be proved that the iterations converge toward the overall minimum for sufficiently small $\gamma_{\epsilon,\text{term}}$ [20, 81].

Serial implementation

The novel serial implementation that we propose is the result of using *block coordinate descent* [20, 127] for dealing with the non-separable quadratic term in the augmented Lagrange formulation of the combined overall control problem (2.18). This approach minimizes the quadratic term directly, in a serial way. Contrarily to the parallel implementation, in the serial implementation one agent after another minimizes its local and interconnecting variables while the other variables stay fixed.

Given the information $\tilde{\mathbf{w}}_{\text{in,prev},ij}(k) = \tilde{\mathbf{w}}_{\text{in},ij}(k)^{(s)}, \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) = \tilde{\mathbf{w}}_{\text{out},ij}(k)^{(s)}$ computed at the current iteration s for each agent $j \in \mathcal{N}_i$ that has solved its problem *before* agent i in the *current* iteration s, and given the previous information $\tilde{\mathbf{w}}_{\text{prev},ij}(k) = \tilde{\mathbf{w}}_{ij}^{(s-1)}(k)$ of the *last* iteration $s-1$ for the other agents, agent i solves problem (2.20) using the following additional objective function:

$$
\begin{aligned}
J_{\text{inter},i}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k), \tilde{\mathbf{w}}_{\text{out},ji}(k), \tilde{\boldsymbol{\lambda}}_{\text{in},ji}(k)^{(s)}, \tilde{\boldsymbol{\lambda}}_{\text{out},ij}(k)^{(s)}\right) &= \begin{bmatrix} \tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k) \\ -\tilde{\boldsymbol{\lambda}}^{(s)}_{\text{out},ij}(k) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out},ji}(k) \end{bmatrix} \\
&\quad + \frac{\gamma_c}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{out},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{in},ji}(k) \end{bmatrix} \right\|_2^2 .
\end{aligned}
$$

Thus, contrarily to the parallel implementation, the serial implementation uses both information from the current iteration and from the last iteration. The serial implementation implements step 2.(b) of the general scheme as follows at decision step k, iteration s:

(ii) 2 For $i = 1, \ldots, n$, *one agent after another*, agent i determines $\tilde{\mathbf{x}}_i(k+1)^{(s)}$, $\tilde{\mathbf{u}}_i(k)^{(s)}$, $\tilde{\mathbf{w}}_{\text{in},ji}(k)^{(s)}$, $\tilde{\mathbf{w}}_{\text{out},ji}(k)^{(s)}$ by solving (2.21), and sends to each agent $j \in \mathcal{N}_i$ the computed values $\tilde{\mathbf{w}}_{\text{out},ji}(k)^{(s)}$.

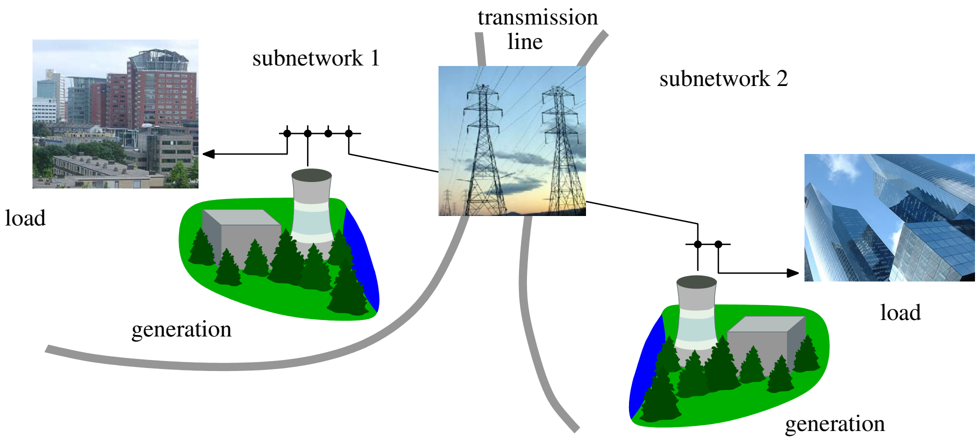


Figure 2.4: Example of two subnetworks, each with loads and power generation facilities. Power flows over the transmission line between the subnetworks. The load-frequency control problem involves adjusting the generation in each subnetwork such that the frequency deviation is maintained close to zero under load disturbances.

The role of the scalar γ_c is similar as for the parallel implementation, except that for the serial implementation γ_c penalizes the deviation from the interconnecting variable iterates that were computed by the agents before agent i in the current iteration and by the other agents during the last iteration. Note that when for the parallel scheme $\gamma_b = \gamma_c$ the additional objective functions are the same for the parallel and the serial scheme, except for the previous information used: the parallel implementation uses only information from the last iteration, the serial also from the current.

In the next section we experimentally assess the performance of the parallel and the serial scheme and discuss which of the two schemes yields a better performance.

2.5 Application: Load-frequency control

In this section we propose the use of the techniques for multi-agent single-layer MPC discussed above for a particular problem in power networks. The problem that we consider is *load-frequency control* [82]. The frequency is one of the main variables characterizing the power network. The purpose of load-frequency control is to keep power generation close to power consumption under consumption disturbances, such that the frequency is maintained close to a nominal frequency of typically 50 or 60 Hz [82]. At an international level power networks become more interconnected and in addition power flows become more unpredictable, e.g., due to large-scale unpredictable power generation using wind turbines. In order to assure correct load-frequency control in the future, current control strategies will be replaced by more advanced strategies that automatically and online determine how the actuators in the network have to be set. Since at an international level countries are not willing to give away access to actuators and sensors in their own subnetworks, they will have to

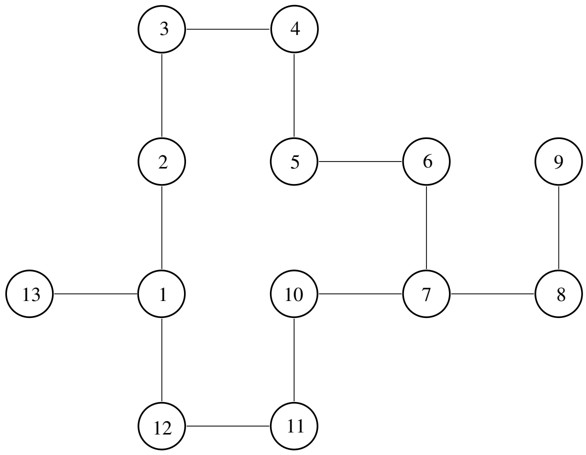


Figure 2.5: The overall network, consisting of 13 subnetworks.

install controllers that cooperatively control the overall network.

A large number of control strategies has been developed for load-frequency control [70]. In the 70s, load-frequency control started being developed with control strategies based on centralized, non-MPC control (see [42, 47, 125]). From the 80s on also, distributed, non-MPC schemes appeared [3, 78, 119, 151, 152]. Recently, also MPC-based schemes have been proposed. A centralized MPC scheme for load-frequency control was proposed in [126]. A decentralized MPC scheme for load-frequency control was proposed in [8]. The latter approach is a decentralized approach that does not take the interconnections between subnetworks explicitly into account. In [28] a distributed MPC scheme is proposed for load-frequency control assuming that only once per control step information between agents can be exchanged. Also in [144] a distributed MPC scheme is applied to a load-frequency control example. The scheme uses distributed state estimation to provide nominal stability and performance properties. We consider distributed MPC using the parallel and serial scheme of Section 2.4.4, which explicitly take into account the interconnections between subnetworks, and use multiple iterations of information exchange before deciding on which actions to take.

2.5.1 Benchmark system

Our benchmark network consists of subnetworks with consumption and generation capabilities, as illustrated in Figure 2.4 for two subnetworks. We consider a network divided into 13 subnetworks as shown in Figure 2.5. Each subnetwork is controlled by one control agent. This control agent has to keep the frequency deviation within its subnetwork close to zero under minimal generation changes. Each control agent can only make measurements and set actuators in its own subnetwork.

We consider rather simplified dynamics for the subnetwork models, that do however include the basic elements of power injection, power consumption, and power flow over

constant	value
$\eta_{\mathrm{K},i}$	120
$\eta_{\mathrm{S},ij}$	0.5
$\eta_{\mathrm{S},ji}$	0.5
$\eta_{\mathrm{T},i}$ (s)	20

Table 2.1: Values of the parameters of the subnetworks, for $i \in \{1,\ldots,n\}$ and $j \in \mathcal{N}_i$.

power lines, and that do show the basic characteristics of the load-frequency control problem. Let the continuous-time linearized dynamics of subnetwork i be described by the following second-order dynamics, as taken from [28]:

$$\begin{aligned}
\frac{\mathrm{d}x_{\Delta\delta,i}}{\mathrm{d}t}(t) &= 2\pi x_{\Delta f,i}(t) \\
\frac{\mathrm{d}x_{\Delta f,i}}{\mathrm{d}t}(t) &= -\frac{1}{\eta_{\mathrm{T},i}}x_{\Delta f,i}(t) + \frac{\eta_{\mathrm{K},i}}{\eta_{\mathrm{T},i}}u_{\Delta P_{\mathrm{gen}},i}(t) - \frac{\eta_{\mathrm{K},i}}{\eta_{\mathrm{T},i}}d_{\Delta P_{\mathrm{dist}},i}(t) \\
&\quad + \frac{\eta_{\mathrm{K},i}}{\eta_{\mathrm{T},i}}\left(\sum_{j\in\mathcal{N}_i}\frac{\eta_{\mathrm{S},ij}}{2\pi}\left(x_{\Delta\delta,j}(t) - x_{\Delta\delta,i}(t)\right)\right) \\
\mathbf{y}_i(t) &= \begin{bmatrix} x_{\Delta\delta,i}(t) \\ x_{\Delta f,i}(t)\end{bmatrix},
\end{aligned}$$

where at time t, for subnetwork i, $x_{\Delta\delta,i}(t)$ is the incremental phase angle deviation in rad, $x_{\Delta f,i}(t)$ is the incremental frequency deviation in Hz, $u_{\Delta P_{\mathrm{gen}},i}(t)$ is the incremental change in power generation in per unit (p.u.), $d_{\Delta P_{\mathrm{dist}},i}(t)$ is a disturbance in the load in p.u., $\mathbf{y}_i(t)$ are the measurements of the states, and $\eta_{\mathrm{K},i}$ is the subnetwork gain, $\eta_{\mathrm{T},i}$ is the subnetwork time constant in s, $\eta_{\mathrm{S},ij}$ is a synchronizing coefficient of the line between subnetwork i and j. The values for these constants are given in Table 2.1. Since we assume that the outputs $\mathbf{y}_i(t)$ measure the state variables noise-free, we will without loss of generality leave out the outputs $\mathbf{y}_i(t)$ and only focus on the states $\mathbf{x}_i(t)$ in the following.

Remark 2.8 For subnetwork i the derivative $\frac{\mathrm{d}x_{\Delta f,i}}{\mathrm{d}t}(t)$ depends on $x_{\Delta\delta,j}(t)$, for $j \in \mathcal{N}_i$, which are variables of the subnetworks $j \in \mathcal{N}_i$. The variables $x_{\Delta\delta,j}(t)$ will therefore cause an interconnecting constraint between the control problems of agents i and j. □

Defining the local control input $u_i(k) = u_{\Delta P_{\mathrm{gen}},i}(k)$, the local exogenous input $d_i(k) = d_{\Delta P_{\mathrm{dist}},i}(k)$, the local states $\mathbf{x}_i(k) = [x_{\Delta\delta,i}(k),\ x_{\Delta f,i}(k)]^{\mathrm{T}}$, the remaining variables $\mathbf{v}_i(k) = [x_{\Delta\delta,j_{i,1}}(k),\ldots,x_{\Delta\delta,j_{i,m_i}}(k)]^{\mathrm{T}}$, and discretizing the continuous-time model using an Euler approximation (with a step size of $T_{\mathrm{p}} = 0.25$ s), the dynamics of subnetwork i can be written as:

$$\mathbf{x}_i(k+1) = \mathbf{A}_i\mathbf{x}_i(k) + \mathbf{B}_{1,i}u_i(k) + \mathbf{B}_{2,i}d_i(k) + \mathbf{B}_{3,i}\mathbf{v}_i(k), \tag{2.24}$$

where

$$\mathbf{A}_i = \begin{bmatrix} 1 & T_{\mathrm{p}}2\pi \\ \sum_{j\in\mathcal{N}_i}\left(T_{\mathrm{p}}\frac{-\eta_{\mathrm{K},i}\eta_{\mathrm{S},ij}}{2\pi\eta_{\mathrm{T},i}}\right) & 1 - T_{\mathrm{p}}\frac{1}{\eta_{\mathrm{T},i}}\end{bmatrix} \qquad \mathbf{B}_{1,i} = \begin{bmatrix} 0 \\ T_{\mathrm{p}}\frac{\eta_{\mathrm{K},i}}{\eta_{\mathrm{T},i}}\end{bmatrix}$$

$$\mathbf{B}_{2,i} = \begin{bmatrix} 0 \\ -T_\mathrm{p}\frac{\eta_{\mathrm{K},i}}{\eta_{\mathrm{T},i}} \end{bmatrix} \qquad \mathbf{B}_{3,i} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ T_\mathrm{p}\frac{\eta_{\mathrm{K},i}\eta_{\mathrm{S},ij_{i,1}}}{2\pi\eta_{\mathrm{T},i}} & T_\mathrm{p}\frac{\eta_{\mathrm{K},i}\eta_{\mathrm{S},ij_{i,2}}}{2\pi\eta_{\mathrm{T},i}} & \dots & T_\mathrm{p}\frac{\eta_{\mathrm{K},i}\eta_{\mathrm{S},ij_{i,m_i}}}{2\pi\eta_{\mathrm{T},i}} \end{bmatrix}.$$

2.5.2 Control setup

The agents use the multi-agent single-layer MPC approach as discussed in Section 2.4.3. In order to implement this scheme, the prediction model, the interconnecting variables, the control objectives, and possibly additional constraints have to be specified:

- Prediction model. Agent i uses as prediction model M_i (2.24) over the time span from $k+1$ until $k+N$.
- Interconnecting variables. The interconnecting inputs for agent i are defined as in (2.9), and the interconnecting outputs for agent i are defined as in (2.10), with:

$$\tilde{\mathbf{K}}_i = \begin{bmatrix} 1 & 0 & & & & 0 & & \\ \vdots & \vdots & & & & \vdots & & \\ 1 & 0 & & & & 0 & & \\ & & \ddots & & & & \ddots & \\ & & & 1 & 0 & & & 0 \\ & & & \vdots & \vdots & & & \vdots \\ & & & 1 & 0 & & & 0 \end{bmatrix},$$

  such that the interconnecting inputs are $x_{\Delta\delta,j}(k+1+l)$, and the interconnecting outputs are $x_{\Delta\delta,i}(k+1+l)$, for $j \in \mathcal{N}_i$ and $l = 0,\dots,N-1$.
- Local control objectives. Since agent i has to minimize the frequency deviation and the control input changes in its subnetwork, it uses the following quadratic local objective function:

$$J_{\mathrm{local},i}\left(\tilde{\mathbf{x}}_i(k+1), \tilde{\mathbf{u}}_i(k)\right) = \sum_{l=0}^{N-1} \begin{bmatrix} \mathbf{x}_i(k+1+l) \\ u_i(k+l) \end{bmatrix}^\mathrm{T} \begin{bmatrix} \mathbf{Q}_{i,\mathrm{x}} & 0 \\ 0 & Q_{i,\mathrm{u}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i(k+1+l) \\ u_i(k+l) \end{bmatrix}$$

  where

$$\mathbf{Q}_{i,\mathrm{x}} = \begin{bmatrix} 0 & 0 \\ 0 & 100 \end{bmatrix}, \quad Q_{i,\mathrm{u}} = 10.$$

  A quadratic function has the advantage that larger deviations are penalized more, and moreover that the objective function is convex.
- Additional constraints. Upper and lower bounds are imposed on the changes in power generation and on the changes in angle and frequency:

$$u_{\mathrm{min},i} \leq \mathbf{u}_i(k+l) \leq u_{\mathrm{max},i}$$
$$\mathbf{x}_{i,\mathrm{min}} \leq \mathbf{x}_i(k+1+l) \leq \mathbf{x}_{i,\mathrm{max}},$$

  for $l = 0,\dots,N-1$, and $u_{\mathrm{min},i} = -0.3$, $u_{\mathrm{max},i} = 0.3$, $\mathbf{x}_{i,\mathrm{min}} = [-10,-10]^\mathrm{T}$, $\mathbf{x}_{i,\mathrm{max}} = [10,10]^\mathrm{T}$.

The defined subnetwork models, interconnecting variables, local control objectives, and additional constraints lead to an overall combined control problem (2.15) that is convex.

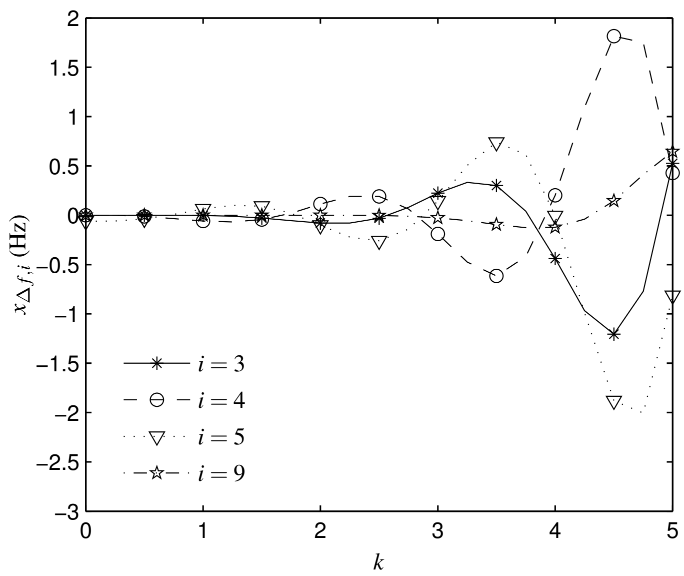


Figure 2.6: Uncontrolled simulation of frequency deviation after a small disturbance in subnetwork 5.

2.5.3 Simulations

We simulate the network in Matlab v7.3 [98]. The network is simulated in discrete time steps of 0.25 s, for $k_\mathrm{f} = 20$ steps. Every 0.25 s the control agents measure the state of their subnetwork after which they either employ the serial or the parallel scheme to determine which action to take next. As reference for the performance a hypothetical single agent that uses the overall combined control problem (2.15) is employed. Each of the schemes uses a warm start when possible, i.e., when the solution from a previous control cycle is available. Iterations are stopped when the stopping condition (2.23) is satisfied, or when a maximum number of 5000 iterations has been performed.

The MPC problems solved by the individual control agents at each iteration are quadratic programming problems with linear constraints. These problems are efficiently solved by the ILOG CPLEX v10 Barrier QP solver [71], which we use through the Tomlab v5.7 [66] interface in Matlab v7.3 [98].

To assess the performance of the schemes discussed above, we first illustrate the uncontrolled behavior of the network after a disturbance for a particular scenario, then we consider the performance of the schemes over the full simulation span for a particular setting of the parameters, and then we focus on how the parameters γ_c and $\gamma_{\epsilon,\mathrm{term}}$ influence the performance of the schemes at a single control cycle.

Scenario without control

It is easy to verify by inspection of the eigenvalues of the overall network that the network is unstable when no control is employed. To illustrate this instability, we first consider the following uncontrolled scenario. The subnetworks start in steady state, i.e., $\mathbf{x}_i(0) = [0,0]^{\mathrm{T}}$, for $i = \{1,\ldots,n\}$). A disturbance $d_{\triangle P_{\mathrm{dist}},i}(k) = 1.10^{-2}$ is imposed at $k = -1$ in subnetwork 5.

Figure 2.6 shows the evolution of the frequency deviation right after the disturbance has appeared, i.e., starting from $k = 0$, in a number of representative subnetworks when no control actions are taken. Clearly, without control agents acting on the generation, the dynamics of the network directly after the disturbance become unstable, and the magnitudes of the oscillations of the frequency deviations increase quickly after the fault.

Performance of control over the full simulation span

We now consider the performance that the parallel and serial schemes discussed in this chapter can achieve for particular values of the control parameters. We compare the performance of the serial and parallel scheme with each other and with a hypothetical centralized control agent that solves the overall combined MPC problem.

We consider 50 scenarios in which a randomly chosen disturbance $d_{\triangle P_{\mathrm{dist}},i}$ from the domain $[-1.10^{-2}, 1.10^{-2}]$ appears in a randomly chosen subnetwork $i \in \{1,\ldots,13\}$. In each scenario, we let time step $k = 0$ correspond to the time step right after the disturbance has appeared. Hence, we consider the performance of the control agents with respect to dealing with the consequences of the disturbance.

To compare the performance of the schemes over the full simulation period, costs are computed over the full simulation as:

$$J_{\mathrm{sim}} = \sum_{i=1}^{n} \sum_{l=0}^{k_{\mathrm{f}}-1} J_{\mathrm{stage},i}\left(\bar{\mathbf{x}}_i(1+l), \bar{\mathbf{u}}_i(l), \bar{\mathbf{y}}_i(l)\right),$$

where the bar indicates that the value of the variable is the actual value as appearing in the evolution of the network, and not the predicted value as predicted by a control agent during its optimization. E.g., $\bar{\mathbf{x}}_i(k)$ refers to the actual state of subnetwork i at time k, and not to the state predicted by a control agent. No penalty term is included for violation of the upper or lower bounds on the variables.

As parameters we here consider as specific setting for the length of the prediction horizon $N = 5$, and for the values of the parameters of the schemes $\gamma_{\mathrm{c}} = 1$, $\gamma_{\epsilon,\mathrm{term}} = 1e^{-4}$, and $\gamma_{\mathrm{b}} = 2\gamma_{\mathrm{c}}$, which for overall convex problems guarantees convergence toward an overall optimal solution. Below we will further discuss the influence of different values of the parameters on the performance of the control.

Table 2.2 shows over all scenarios the average results of the schemes, consisting of the average performance $J_{\mathrm{sim,avg}}$, the average number of iterations required $N_{\mathrm{iter,avg}}$, and the total computation time in seconds[2]. We observe that the average performance $J_{\mathrm{sim,avg}}$ that is obtained over a full simulation by the serial and the parallel scheme are very close to each other. In addition the performance of these multi-agent schemes is very close to

[2] For computing the total computation time required for the parallel and the serial scheme, only the time spent on solving the optimization problems is summed, since the time involved in setting up the optimization problems is negligible. The simulations are implemented in a central simulation environment. Hence, the parallel scheme is in fact executed in a serial fashion. Therefore, the computation time of a single iteration is taken as the maximum computation time required for solving either of the local optimization problems. Since the simulations are executed in a central simulation environment also no communication delays are accounted for.

scheme	$J_{\text{sim,avg}}$	$N_{\text{iter,avg}}$	$T_{\text{comp,avg}}$
centralized	0.2746	-	0.3
serial	0.2746	129	16.6
parallel	0.2746	335	5.9

Table 2.2: Results of the schemes over all experiments. The table shows over all 50 scenarios the average performance $J_{sim,avg}$, the average number of iterations $N_{iter,avg}$, and the average total computation time $T_{comp,avg}$ (s). The results have been obtained for parameter settings $N = 5$, $k_f = 20$, $\gamma_c = 1$, $\gamma_{\epsilon,term} = 1.10^{-4}$, and starting from 50 different initial states, each of which are a state appearing right after a random disturbance between -0.01 and 0.01 p.u. in one of the subnetworks has occurred.

the performance of the centralized scheme. Hence, the controls agents have obtained the performance of the centralized control agent in a distributed way.

We also observe from Table 2.2 that the serial scheme on average requires fewer iterations $N_{\text{iter,avg}}$ per simulation than the parallel scheme. This can be explained by the fact that the serial scheme uses information from both the previous and the current iteration, whereas the parallel scheme only uses information from a previous iteration.

In Table 2.2 we also observe that the total computation time in seconds per simulation on average $T_{\text{comp,avg}}$ is larger for the serial scheme than for the parallel scheme. This is explained by the fact that in the serial scheme only one agent at a time performs a computation step within an iteration, whereas in the parallel scheme multiple control agents perform computations at the same time. Compared to the centralized scheme, the parallel and serial scheme have a larger total computation time than the centralized scheme.

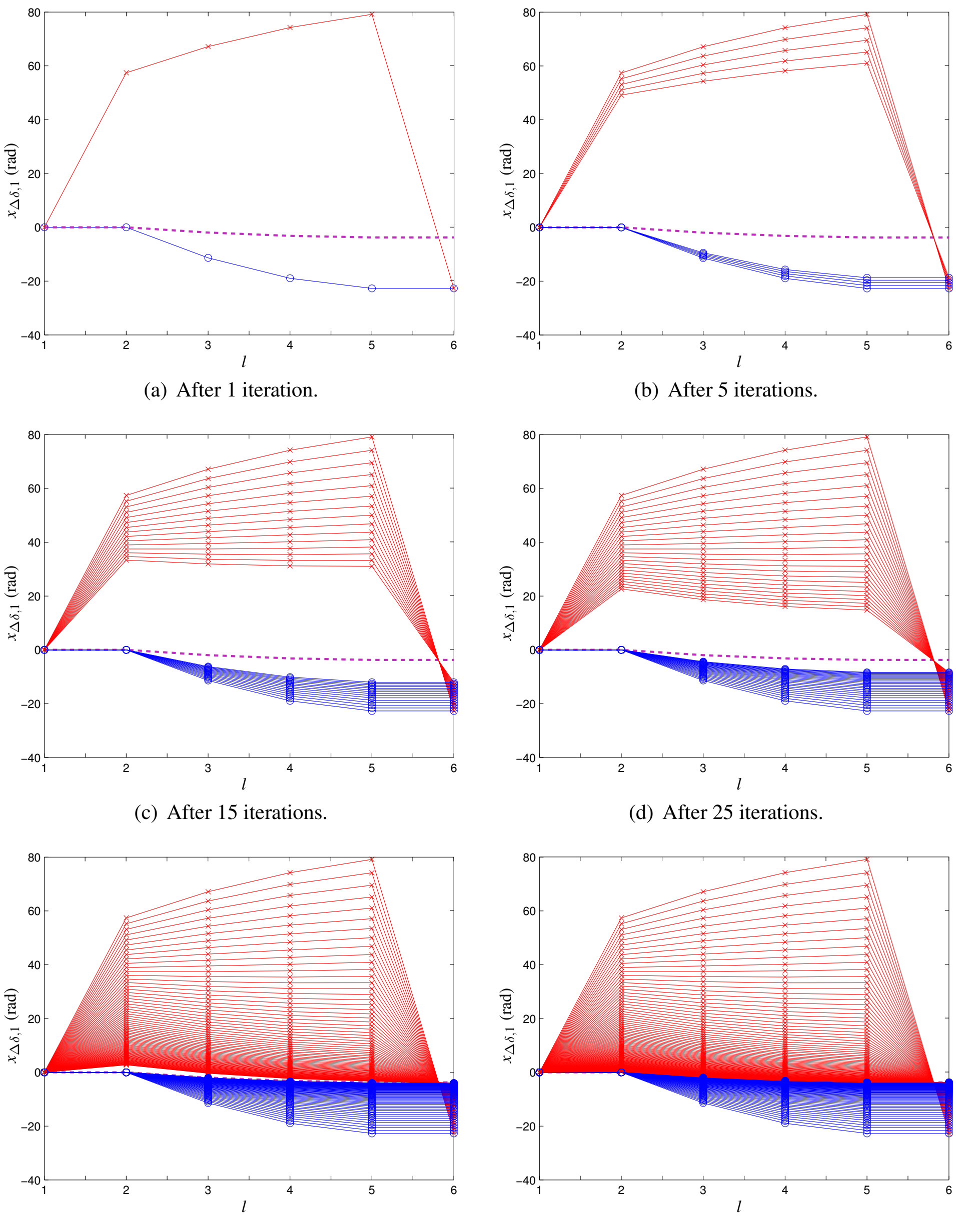
Below we will discuss these results further, after illustrating the influence of different parameter values on the performance of the parallel and serial scheme.

Iterations at a single control cycle

To illustrate the operation of the serial scheme at a particular control cycle, consider Figure 2.7. The figure illustrates the typical behavior of values of interconnecting variables going toward each other over the iterations at a particular control cycle for a network consisting of two subnetworks. In this network, the values of $\mathbf{x}_i$ are unconstrained.

The figure illustrates for a particular interconnecting input variable of agent 1 over the prediction horizon and the corresponding interconnecting output variable of agent 2 over the prediction horizon, the values that both agents would like their interconnecting variable to take on. After each local computation step, these values are communicated to the other agent, which uses these to update its interconnecting objective function. As the iterations progress the values of the interconnecting input and the corresponding interconnecting output converge to each other, indicating that the values go toward satisfying interconnecting constraints. In addition, since in our case the combined overall problem is convex, the values converge to the solution that would have been obtained with a centralized control agent that would have access to all actuators and sensors in the network.

Depending on the value of the parameter $\gamma_{\epsilon,\text{term}}$ the iterations will terminate sooner or



(a) After 1 iteration.
(b) After 5 iterations.
(c) After 15 iterations.
(d) After 25 iterations.
(e) After 80 iterations.
(f) At convergence.

Figure 2.7: Convergence of the values for interconnecting input variables of agent 1 (solid line with circle) and the corresponding interconnecting output variables of agent 2 (solid line with cross), each corresponding to the variables $x_{\Delta\delta,1}(k+l)$ over a prediction horizon of 6 steps, hence, for $l = 1,2,\ldots,6$. Over the iterations the values converge to the overall optimal solution (dashed line).

later, and depending on the value of the parameter γ_c the values of the interconnecting variables will converge sooner or later to values for which the interconnecting constraints are satisfied. Below we go into this in more detail.

Parameter sensitivity at a single control cycle

To gain more insight into the role of the parameters γ_c and $\gamma_{\epsilon,\text{term}}$ and into the iterations that the serial and the parallel scheme perform, we illustrate the performance of the schemes for a particular representative control problem at a particular control cycle under varying parameter values. The control problem that we consider is the MPC control problem that the agents have to solve right after a disturbance $d_{\Delta P_{\text{dist}},i}$ of magnitude 1.10^{-2} has occurred in subnetwork 5.

To evaluate the solution over the prediction horizon determined by the different schemes at a single control cycle, the inputs coming from the different schemes are implemented to determine the resulting state trajectory, after which the cycle performance J_{cycle} is determined as:

$$J_{\text{cycle}} = \sum_{i=1}^{n} \sum_{l=0}^{N-1} J_{\text{stage},i} \left(\bar{\mathbf{x}}_i(1+l), \mathbf{u}_i(l), \bar{\mathbf{y}}_i(l) \right).$$

No penalty term is included for violations of the bound constraints.

Varying the penalty coefficient We first vary the parameter γ_c, while keeping $\gamma_{\epsilon,\text{term}}$ fixed at 1.10^{-6}. For varying values of the parameter γ_c we determine the cycle performance J_{cycle} at each intermediate iteration. Hence, after each iteration, the actions that the control agents would then choose are used to evaluate the cycle performance J_{cycle} over the prediction horizon.

Figures 2.8 and 2.9 illustrate how the cycle performance J_{cycle} of the control agents using the serial and the parallel scheme changes over the iterations, under various values for γ_c. We clearly observe that as the number of iterations increases, the performance of the solution that the control agents have determined increases as well.

We observe in Figure 2.8 that, indeed, on the one hand for very small values of the penalty term γ_c, the convergence is slow, whereas on the other hand, for larger values of the penalty term γ_c, the convergence is faster. However, we observe in Figure 2.9 that, indeed, when the penalty term γ_c is chosen too large, the convergence slows down again.

For a given value of γ_c, the serial scheme requires fewer iterations and converges faster than the parallel scheme. This behavior is best observed for larger values of γ_c in Figure 2.9. The difference in the number of iterations required is due to the fact that the serial scheme uses information earlier than the parallel scheme. For smaller values of γ_c, as those shown in Figure 2.8, the influence of the additional objective function $J_{\text{inter},i}$ of both the parallel and the serial scheme vanishes, making that the difference between the two schemes vanishes as well.

Varying the stopping tolerance Given a value for γ_c we determine the cycle performance J_{cycle} that the control agents obtain at termination using various values for the stopping tolerance $\gamma_{\epsilon,\text{term}}$. We vary $\gamma_{\epsilon,\text{term}}$ in the set $\{1.10^{-8}, 1.10^{-7}, \ldots, 1, 10\}$.

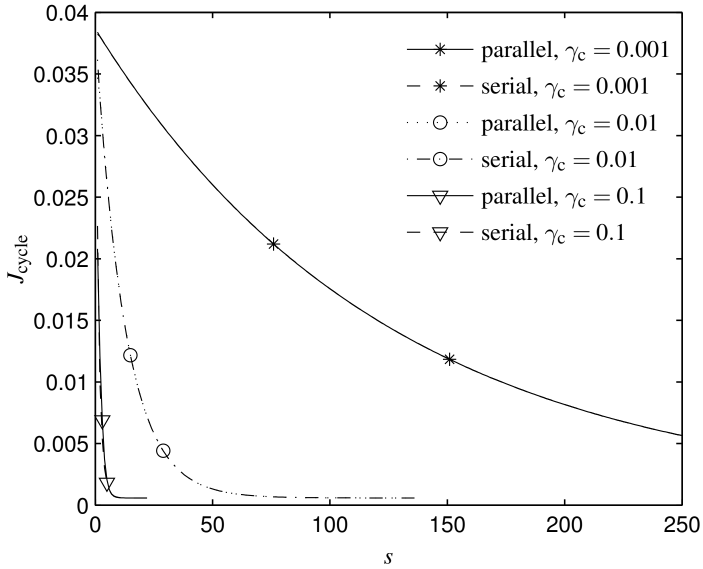


Figure 2.8: The performance of solutions after each iteration for smaller values of γ_c.

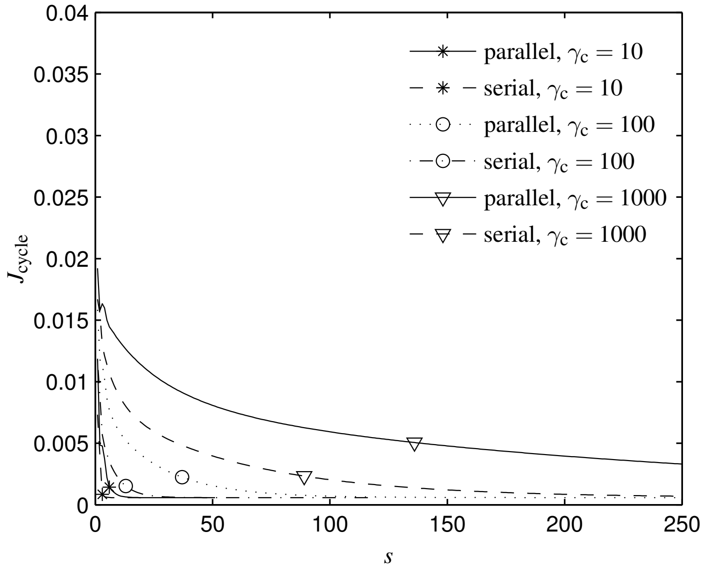


Figure 2.9: The performance of solutions after each iteration, for larger values of γ_c.

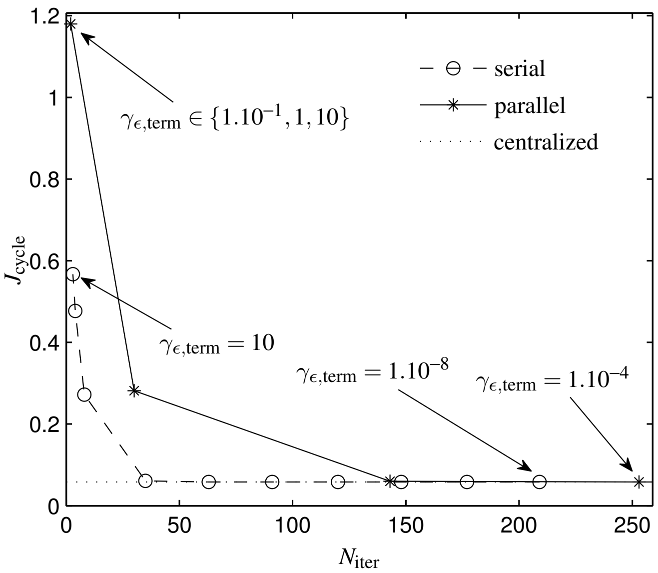


Figure 2.10: The cycle performance versus the number of iterations for $N = 5$, $\gamma_c = 100$, and varying $\gamma_{\epsilon,term}$. Each circle and star represent a scenario with a different value for $\gamma_{\epsilon,term}$. The value of $\gamma_{\epsilon,term}$ decreases when going from left to right.

Figure 2.10 shows the results for varying $\gamma_{\epsilon,\text{term}}$, while keeping γ_c fixed at 100. We observe that with a decreasing value of the stopping tolerance $\gamma_{\epsilon,\text{term}}$, more iterations are required before the stopping condition is satisfied. We also observe that if an appropriate value for $\gamma_{\epsilon,\text{term}}$ is chosen, convergence toward the centralized solution is obtained within a reasonable bound.

It is noted that there is minimal performance that is achieved when $\gamma_{\epsilon,\text{term}}$ becomes larger than a certain value. In Figure 2.10 this is observed for the parallel scheme, which for values of $\gamma_{\epsilon,\text{term}}$ larger than 0.1 achieves the same performance.

When comparing the serial scheme with the parallel scheme, we observe that the serial scheme outperforms the parallel scheme in convergence speed and performance. Furthermore, Figure 2.10 illustrates that over the iterations the performance of both schemes converges toward the performance of the centralized overall scheme.

Discussion

The experiments reported in this section represent a relatively small portion of all experiments that could have been done, involving multiple combinations of network topologies, scheme parameters, prediction horizons, etc. Nonetheless, the results obtained here give an indication of the potential of the approaches discussed in this chapter.

It is noted that both schemes discussed only communicate information common to the control problems of several agents; all other data is only used locally. Agents have only a prediction model of their own subnetwork. This gives flexibility and security, since other agents do not have to know the exact parameters of a particular subnetwork, and in fact the subnetwork may be changed, without having to inform other agents.

The time required to complete the iterations at one control cycle in these experiments is typically larger than a real-time online implementation would allow. However, as Figures 2.8 and 2.9 illustrate, after already a few iterations a relatively good solution can have been obtained, and thus if necessary the iterations could be stopped earlier.

In our experiments we have seen that the serial scheme can outperform the parallel scheme in terms of convergence speed in terms of iterations and the performance obtained. However, we also observed that the serial scheme requires more computation time in seconds in order to perform its computations, when compared to the parallel scheme.

If the time required for one serial iteration is reduced, the serial scheme may also outperform the parallel scheme in total computation time required. Our idea to achieve this is to parallelize the serial scheme, either only within an iteration, or also over iterations. Parallelization can be done when the topology of subnetworks can be seen as a tree. This tree structure of the network makes that control problems of control agents can be solved (partially) in parallel, thus reducing total computation time. Groups of agents operating in parallel may be constructed. Within each group, the serial scheme may be employed [111].

It should be noted that the overall network that the control agents control in this section is highly unstable. As we have seen, a small disturbance in the overall network gives large oscillations if not controlled properly. For this reason, it is important for the control agents to obtain very accurate values of the interconnecting variables over the prediction horizon. For applications in which the local subnetwork dynamics and objectives do not depend as much on the values of the interconnecting variables decision making speed can be increased by lowering the value of the stopping tolerance $\gamma_{\epsilon,\text{term}}$.

The dynamics used in this section for representing the power networks dynamics are highly simplified, and the values representing the deviations therefore can also not directly be related to physical values. The linear dynamics assumed are typically valid only over small prediction horizons. However, for our purpose of showing the performance of the control schemes, this is not an issue. More advanced linear models may be used in combination with the schemes considered above to more adequately represent the actual network physics.

2.6 Summary

In this chapter we have considered multi-agent single-layer MPC for the control of transportation networks. We have started with formalizing the dynamics of the subnetworks and the control structure. Then, we have formulated the MPC problem for an individual control agent, assuming that it knows how its surrounding network behaves. Subsequently, we have relaxed this assumption and introduced interconnections between control problems. We have surveyed how these interconnections can be dealt with by discussing the various ways of information exchange and moments at which information exchange takes place. Then, we have focused on a particular type of schemes and have proposed a novel serial scheme, which we have compared with a related parallel scheme. Although under convexity assumptions on the overall combined control problem the schemes converge to overall optimal solutions, it remains to be investigated what the rate of convergence is, how the rate of convergence can be improved, and how this scheme can be extended to other classes of models.

We have proposed the application of the schemes for a load-frequency control problem. Through experimental studies on a network consisting of 13 subnetworks, we have compared the serial scheme with the parallel and a centralized overall scheme. For the serial and the parallel schemes, the performance of the solution obtained converged toward the performance of the solution obtained by the overall control problem, provided that the overall control problem is convex.

The results of the experiments illustrate that the proposed serial scheme generally has preferable properties in terms of the solution quality and the number of iterations required. However, the parallel scheme requires less time. Through parallelization the total computation time required per iteration by the serial scheme may be made more efficient, ultimately resulting in a scheme that requires also fewer total computation time than the parallel scheme.

In Chapter 3 we extend the serial method to situations in which the problem of controlling the transportation network cannot be formulated as a convex problem. In particular we extend the method to deal with networks modeled as hybrid systems in which both continuous and discrete dynamics appear, a situation typically appearing when, e.g., continuous flows together with discrete actions are present.

In Chapter 4 we discuss how a supervisory control layer can control the control agents of a lower control layer, that are organized as, e.g., the structure considered in this chapter. The supervisory control layer takes into account the dynamics of both the lower control layer and the underlying physical network.

In Chapter 5 we consider how an even higher supervisory control layer can control the control agents in a lower control layer. The control agents in the higher control layer do not take into accounts the dynamics of the lower layer, but only consider steady-state characteristics. A scheme related to the schemes addressed in this chapter is used to obtain coordination among the control agents controlling subnetworks that are overlapping and may have nonlinear steady-state characteristics.

Bibliography

[1] L. Acar. Some examples for the decentralized receding horizon control. In *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 1356–1359, Tucson, Arizona, 1992.

[2] M. Aicardi, G. Casalino, R. Minciardi, and R. Zoppoli. On the existence of stationary optimal receding-horizon strategies for dynamic teams with common past information structures. *IEEE Transactions on Automatic Control*, 37:1767–1771, November 1992.

[3] M. Aldeen and J. F. Marsh. Observability, controllability and decentralized control of interconnected power systems. *International Journal on Computers and Electrical Engineering*, 16(4):207–220, 1990.

[4] P.J. Antsaklis and A. Nerode, editors. Special issue on hybrid systems. *IEEE Transactions on Automatic Control*, 43(4), April 1998.

[5] Power Systems Test Case Archive. Parameters of the IEEE 57-bus grid. http://www.ee.washington.edu/research/pstca/.

[6] K. J. Åström and B. Wittenmark. *Computer-Controlled Systems*. Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[7] K. J. Åström, H. Elmqvist, and S. E. Mattsson. Evolution of continuous-time modeling and simulation. In *Proceedings of the 12th European Simulation Multiconference*, pages 9–18, Manchester, UK, June 1998.

[8] N. Atic, D. Rerkpreedapong, A. Hasanovic, and A. Feliachi. NERC compliant decentralized load frequency control design using model predictive control. In *Proceedings on the IEEE Power Engineering Society General Meeting*, Toronto, Canada, July 2003.

[9] C. Audet and J. E. Dennis Jr. Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization*, 11(3):573–594, 2000.

[10] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2007.

[11] T. Başar and G. J. Olsder. *Dynamic Non-Cooperative Game Theory*. Academic Press, London, UK, 1998.

[12] M. Baglietto, T. Parisini, and R. Zoppoli. Neural approximators and team theory for dynamic routing: A receding-horizon approach. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 3283–3288, Phoenix, Arizona, 1999.

[13] P. Barton and C. Pantelides. Modeling of combined discrete/continuous processes. *AIChE Journal*, 40(6):966–979, 1994.

[14] J. Batut and A. Renaud. Daily generation scheduling optimization with transmission constraints: a new class of algorithms. *IEEE Transactions on Power Systems*, 7(3): 982–989, August 1992.

[15] A. G. Beccuti and M. Morari. A distributed solution approach to centralized emergency voltage control. In *Proceedings of the 2006 IEEE American Control Conference*, pages 3445–3450, Minneapolis, Minnesota, June 2006.

[16] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[17] J. Bernussou and A. Titli. *Interconnected Dynamical Systems: Stability, Decomposition and Decentralisation*. North-Holland Publishing Company, Amsterdam, The Netherlands, 1982.

[18] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Beltmore, Massachusetts, 2003.

[19] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, London, UK, 1982.

[20] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, New Hampshire, 1997.

[21] P. R. Bhave and R. Gupta. *Analysis of Water Distribution Networks*. Alpha Science International, Oxford, UK, 2006.

[22] L. G. Bleris, P. D. Vouzis, J. G. Garcia, M. G. Arnold, and M. V. Kothare. Pathways for optimization-based drug delivery. *Control Engineering Practice*, 15(10):1280–1291, October 2007.

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

[24] S. D. Braithwait. Real-time pricing and demand response can work within limits. *Natural Gas and Electricity*, 21(11):1–9, 2005.

[25] M. W. Braun, D. E. Rivera, M. E. Flores, W. M. Carlyle, and K. G. Kempf. A model predictive control framework for robust management of multi-product, multi-echelon demand networks. *Annual Reviews in Control*, 27:229–245, 2003.

[26] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, Philadelphia, Pennsylvania, 1996.

[27] E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Germany, 1995.

[28] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 1:44–52, February 2002.

[29] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[30] C. G. Cassandras, S. Lafortune, and G. J. Olsder. Introduction to the modelling, control and optimization of discrete event systems. In A. Isidori, editor, *Trends in Control: A European Perspective*, pages 217–291. Springer-Verlag, Berlin, Germany, 1995.

[31] A. J. Conejo, F. J. Nogales, and F. J. Prieto. A decomposition procedure based on approximate newton directions. *Mathematical Programming, Series A*, 93(3):495–515, December 2002.

[32] A. R. Conn, K. Scheinberg, and P. L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79(1–3):397–414, 1997.

[33] C. F. Daganzo. *Fundamentals of Transportation and Traffic Operations*. Pergamon Press, New York, New York, 1997.

[34] R. David. Modeling of dynamic systems by Petri nets. In *Proceedings of the 1st European Control Conference*, pages 136–147, Grenoble, France, July 1991.

[35] B. De Schutter and T. J. J. van den Boom. Model predictive control for max-min-plus-scaling systems. In *Proceedings of the 2001 American Control Conference*, pages 319–324, Arlington, Virginia, June 2001.

[36] B. De Schutter, T. van den Boom, and A. Hegyi. A model predictive control approach for recovery from delays in railway systems. *Transportation Research Record*, (1793):15–20, 2002.

[37] W. B. Dunbar and R. M. Murray. Model predictive control of coordinated multi-vehicle formations. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4631–4636, Las Vegas, Nevada, December 2002.

[38] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, April 2006.

[39] Dynasim. Dymola – User's Manual. Technical report, Dynasim AB, Lund, Sweden, 2004.

[40] A. Edris, R. Adapa, M. H. Baker, L. Bohmann, K. Clark, K. Habashi, L. Gyugyi, J. Lemay, A. S. Mehraban, A. K. Meyers, J. Reeve, F. Sener, D. R. Torgerson, and R. R. Wood. Proposed terms and definitions for flexible AC transmission system (FACTS). *IEEE Transactions on Power Delivery*, 12(4):1848–1853, October 1997.

[41] H. El Fawal, D. Georges, and G. Bornard. Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian. In *Proceedings of the 1998 International Conference on Systems, Man, and Cybernetics*, pages 3874–3879, San Diego, California, 1998.

[42] O. I. Elgerd and C. Fosha. Optimum megawatt frequency control of multi-area electric energy systems. *IEEE Transactions on Power Apparatus and Systems*, PAS-89 (4):556–563, February 1970.

[43] Elkraft Systems. Power failure in Eastern Denmark and Southern Sweden on 23 September 2003 – preliminary report on the course of events. Technical report, Elkraft Systems, Holte, Denmark, 2003.

[44] H. Elmqvist, F. E. Cellier, and M. Otter. Object-oriented modeling of hybrid systems. In *Proceedings of the European Simulation Symposium*, pages xxxi–xli, Delft, The Netherlands, October 1998.

[45] B. Fardanesh. Future trends in power system control. *IEEE Computer Applications in Power*, 15(3):24–31, July 2002.

[46] R. G. Farmer and P. M. Anderson. *Series Compensation of Power Systems*. PBLSH, Encinitas, California, 1996.

[47] C. E. Fosha and O. I. Elgerd. The megawatt frequency control problem: A new approach via optimal control theory. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(4):563–577, April 1970.

[48] G. Georges. Decentralized adaptive control for a water distribution system. In *Proceedings of the 3rd IEEE Conference on Control Applications*, pages 1411–1416, Glasgow, UK, 1999.

[49] T. Geyer, M. Larsson, and M. Morari. Hybrid emergency voltage control in power systems. In *Proceedings of the European Control Conference 2003*, Cambridge, UK, September 2003.

[50] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimisation*, 12(4):979–1006, 2002.

[51] G. Glanzmann and G. Andersson. FACTS control for large power systems incorporating security aspects. In *Proceedings of X SEPOPE*, Florianopolis, Brazil, May 2006.

[52] G. Glanzmann and G. Andersson. Using FACTS devices to resolve congestions in transmission grids. In *Proceedings of the CIGRE/IEEE PES International Symposium*, San Antonio, Texas, October 2005.

[53] M. Gomez, J. Rodellar, F. Vea, J. Mantecon, and J. Cardona. Decentralized predictive control of multireach canals. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3885–3890, San Diego, California, 1998.

[54] A. H. González, D. Odloak, and J. L. Marchetti. Predictive control applied to heat-exchanger networks. *Chemical Engineering and Processing*, 45(8):661–671, August 2006.

[55] E. González-Romera, M. Á. Jaramillo-Morán, and D. Carmona-Fernández. Forecasting of the electric energy demand trend and monthly fluctuation with neural networks. *Computers and Industrial Engineering*, 52:336–343, April 2007.

[56] G. C. Goodwin, M. M. Seron, R. H. Middleton, M. Zhang, B. F. Hennessy, P. M. Stone, and M. Menabde. Receding horizon control applied to optimal mine planning. *Automatica*, 42(8):1337–1342, August 2006.

[57] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[58] A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):102–112, March 2005.

[59] D. J. Hill. Nonlinear dynamic load models with recovery for voltage stability studies. *IEEE Transactions on Power Systems*, 8(1):166–176, February 1993.

[60] D. J. Hill, Y. Guo, M. Larsson, and Y. Wang. Global control of complex power systems. In G. Chen, D. J. Hill, and X. Yu, editors, *Bifurcation Control: Theory and Applications*, Lecture Notes in Control and Information Sciences, pages 155–187. Springer, Berlin, Germany, 2003.

[61] P. Hines, L. Huaiwei, D. Jia, and S. Talukdar. 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*, pages 273–278, Tucson, Arizona, March 2005.

[62] N. G. Hingorani and L. Gyugyi. *Understanding FACTS concepts and technology of flexible AC transmission systems*. IEEE Press, New York, New York, 2000.

[63] I. A. Hiskens and K. Mitsumoto. Dynamical systems benchmark library. URL: http://psdyn.ece.wisc.edu/IEEE_benchmarks/, 2005.

[64] Y. C. Ho, P. B. Luh, and G. J. Olsder. A control-theoretic view on incentives. In *Proceedings of the 19th IEEE Conference on Decision and Control*, pages 1160–1170, Albuquergue, New Mexico, December 1980.

[65] K. Holmström, A. O. Göran, and M. M. Edvall. User's guide for Tomlab /SNOPT, December 2006.

[66] K. Holmström, A. O. Göran, and M. M. Edvall. User's guide for Tomlab /CPLEX, June 2007.

[67] M. Houwing, A. N. Ajah, P. M. Herder, and I. Bouwmans. Addressing uncertainties in the design and operation of residential distributed energy resources: Case study

of a micro-CHP system. In *Proceedings of the 10th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction*, Ischia Island, Italy, June 2007.

[68] M. Houwing, R. R. Negenborn, P. Heijnen, B. De Schutter, and J. Hellendoorn. Least-cost model predictive control of residential energy resources when applying μCHP. In *Proceedings of Power Tech 2007*, Lausanne, Switzerland, July 2007. Paper 291.

[69] G. Hug-Glanzmann, R. R. Negenborn, G. Andersson, B. De Schutter, and J. Hellendoorn. Multi-area control of overlapping areas in power systems for FACTS control. In *Proceedings of Power Tech 2007*, Lausanne, Switzerland, July 2007. Paper 277.

[70] Ibraheem, P. Kumar, and D. P. Kothari. Recent philosophies of automatic generation control strategies in power systems. *IEEE Transactions on Power Systems*, 20(1): 346–357, February 2005.

[71] ILOG. CPLEX. URL: http://www.ilog.com/products/cplex/, 2007.

[72] R. Irizarry-Rivera and W. D. Seider. Model-predictive control of the Czochralski crystallization process. Part I. Conduction-dominated melt. *Journal of Crystal Growth*, 178(4):593–611, 1997.

[73] N. Jenkins, R. Allan, P. Crossley, D. Kirschen, and G. Strbac. *Embedded Generation*. TJ International, Padstow, UK, 2000.

[74] D. Jia and B. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the 2002 American Control Conference*, pages 4507–4512, Anchorage, Alaska, May 2002.

[75] D. Jia and B. H. Krogh. Distributed model predictive control. In *Proceedings of the 2001 American Control Conference*, pages 2767–2772, Arlington, Virginia, June 2001.

[76] D. Karlsson and D. J. Hill. Modelling and identification of nonlinear dynamic loads in power systems. *IEEE Transactions on Power Systems*, 9(1):157–163, February 1994.

[77] M. R. Katebi and M. A. Johnson. Predictive control design for large-scale systems. *Automatica*, 33(3):421–425, 1997.

[78] H. Kawabata and M. Kido. A decentralized scheme of load frequency control power system. *Electrical Engineering Japan*, 102(4):100–106, July 1982.

[79] T. Keviczky, F. Borrelli, and G. J. Balas. A study on decentralized receding horizon control for decoupled systems. In *Proceedings of the 2004 American Control Conference*, volume 6, pages 4921–4926, Boston, Massachusetts, June 2004.

[80] B. H. Kim and R. Baldick. A comparison of distributed optimal power flow algorithms. *IEEE Transactions on Power Systems*, 15(2):599–604, May 2000.

[81] B. H. Kim and R. Baldick. Coarse-grained distributed optimal power flow. *IEEE Transactions on Power Systems*, 12(2):932–939, May 1997.

[82] P. Kundur. *Power System Stability and Control*. McGraw-Hill, New York, New York, 1994.

[83] S. Leirens, J. Buisson, P. Bastard, and J.-L. Coullon. A hybrid approach for voltage stability of power systems. In *Proceedings of the 15th Power Systems Computation Conference*, Liège, Belgium, August 2005. Paper 291.

[84] R. M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[85] R. M. Lewis and V. Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002.

[86] R. M. Lewis and V. Torczon. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099, 1999.

[87] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(1–2):191–207, December 2000.

[88] G. Lodewijks. *Dynamics of Belt Systems*. PhD thesis, Delft University of Technology, The Netherlands, 1996.

[89] Z. Lukszo, M. P. C. Weijnen, R. R. Negenborn, B. De Schutter, and M. Ilić. Challenges for process system engineering in infrastructure operation and control. In W. Marquardt and C. Pantelides, editors, *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering* (Garmisch-Partenkirchen, Germany, July 2006), volume 21 of *Computer-Aided Chemical Engineering*, pages 95–100. Elsevier, Amsterdam, The Netherlands, 2006.

[90] Z. Lukszo, M. P. C. Weijnen, R. R. Negenborn, and B. De Schutter. Tackling challenges in infrastructure operation and control using multi-level and multi-agent control. Technical Report 07-027, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, 2007. Submitted to a journal.

[91] E. Lysen, S. Van Egmond, and S. Hagedoorn. Opslag van elektriciteit: Status en toekomstperspectief voor Nederland. Technical Report NEO 0268-05-05-01-002, Utrecht Centrum voor Energieonderzoek – SenterNovem, Utrecht, The Netherlands, 2006. In Dutch.

[92] J. Machowski, J. Bialek, and J. R. Bumby. *Power System Dynamics and Stability*. John Wiley & Sons, New York, New York, 1997.

[93] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice-Hall, Harlow, UK, 2002.

[94] Y. Majanne. Model predictive pressure control of steam networks. *Control Engineering Practice*, 13(12):1499–1505, December 2005.

[95] A. Manzoni, A. S. de Silva, and I. C. Decker. Power systems, dynamics simulation using object-oriented programming. *IEEE Transactions on Power Systems*, 14(1): 249–255, 1999.

[96] R. Martí. Multi-Start Methods. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 12, pages 355–368. Springer, New York, New York, 2006.

[97] MathWorks. Genetic Algorithm and Direct Search Toolbox 2 – User's Guide, 2007.

[98] Mathworks. Matlab. URL: http://www.mathworks.com/, 2007.

[99] S. E. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with Modelica. *Control Engineering Practice*, 6(4):501–510, April 1998.

[100] M. D. Mesarovic, D. Macko, and Y. Takahara. *Theory of Hierarchical Multilevel Systems*. Academic Press, New York, New York, 1970.

[101] P. J. Modi, W. M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous Distributed Constraint Optimization with quality guarantees. *Artificial Intelligence*, 161 (1-2):149–180, January 2005.

[102] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23(4):667–682, 1999.

[103] J. J. Moré and S. J. Wright. *Optimization Software Guide*. SIAM, Philadelphia, Pennsylvania, 1993.

[104] A. S. Morse, C. C. Pantelides, S. Sastry, and J. M. Schumacher, editors. Special issue on hybrid systems. *Automatica*, 35(3), March 1999.

[105] I. R. Navarro, M. Larsson, and G. Olsson. Object-oriented modeling and simulation of power systems using modelica. In *Proceedings of the IEEE Power Engineering Society Winter Meeting*, pages 790–795, Singapore, January 2000.

[106] R. R. Negenborn, B. De Schutter, M.A. Wiering, and H. Hellendoorn. Learning-based model predictive control for Markov decision processes. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005. Paper 2106 / We-M16-TO/2.

[107] R. R. Negenborn, B. De Schutter, and H. Hellendoorn. Multi-agent model predictive control of transportation networks. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, pages 296–301, Fort Lauderdale, Florida, April 2006.

[108] R. R. Negenborn, B. De Schutter, and H. Hellendoorn. Multi-agent model predictive control for transportation networks with continuous and discrete elements. In *Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*, pages 609–614, Delft, The Netherlands, August 2006.

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

[110] R. R. Negenborn, A. G. Beccuti, T. Demiray, S. Leirens, G. Damm, B. De Schutter, and M. Morari. Supervisory hybrid model predictive control for voltage stability of power networks. In *Proceedings of the American Control Conference 2007*, pages 5444–5449, New York, New York, July 2007.

[111] R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Efficient implementation of serial multi-agent model predictive control by parallelization. In *Proceedings of the 2007 IEEE International Conference on Networking, Sensing, and Control*, pages 175–180, London, UK, July 2007.

[112] R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. Technical Report 07-024, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, 2007. To appear in Engineering Applications of Artificial Intelligence.

[113] R. R. Negenborn, S. Leirens, B. De Schutter, and J. Hellendoorn. Supervisory nonlinear MPC for emergency voltage control using pattern search. Technical Report 07-025, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, 2007. Submitted to a journal.

[114] P. G. Neumann. Widespread network failures. *Communications of the ACM*, 50(2): 112, 2007.

[115] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999. ISBN 0-387-98793-2.

[116] F. J. Nogales, F. J. Prieto, and A. J. Conejo. Multi-area AC optimal power flow: A new decomposition approach. In *Proceedings of the 13th Power Systems Control Conference (PSCC)*, pages 1201–1206, Trondheim, Germany, 1999.

[117] S. Ochs, S. Engell, and A. Draeger. Decentralized vs. model predictive control of an industrial glass tube manufacturing process. In *Proceedings of the 1998 IEEE Conference on Control Applications*, pages 16–20, Trieste, Italy, 1998.

[118] M. Otter, H. Elmqvist, and S. E. Mattson. Hybrid modeling in Modelica based on the synchronous data flow principle. In *Proceedings of the International Symposium on Computer Aided Control System Design*, pages 151–157, Kohala Coast-Island of Hawai'i, Hawai'i, August 1999.

[119] Y. M. Park and K. Y. Lee. Optimal decentralized load frequency control. *Electrical Power Systems Research*, 7(4):279–288, September 1984.

[120] M. Pehnt, M. Cames, C. Fischer, B. Praetorius, L. Schneider, K. Schumacher, and J. Vob. *Micro Cogeneration: Towards Decentralized Energy Systems*. Springer, Berlin, Germany, 2006.

[121] L. R. Petzold. A description of DASSL - A differential/algebraic system solver. In *Proceedings of the 10th World Congress on System Simulation and Scientific Computation*, pages 430–432, Montreal, Canada, August 1983.

[122] P. C. Piela, T. G. Epperly, K. M. Westerberg, and A. W. Westerberg. ASCEND: an object-oriented computer environment for modeling and analysis: The modeling language. *Computers and Chemical Engineering*, 15(1):53–72, January 1991.

[123] S. Piñón, E. F. Camacho, B. Kuchen, and M. Peña. Constrained predictive control of a greenhouse. *Computers and Electronics in Agriculture*, 49(3):317–329, December 2005.

[124] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes*. Cambridge University Press, Cambridge, UK, 2007.

[125] G. Quazza. Noninteracting controls of interconnected electric power systems. *IEEE Transactions on Power Apparatus and Systems*, PAS-85(7):727–741, July 1966.

[126] D. Rerkpreedapong, N. Atic, and A. Feliachi. Economy oriented model predictive load frequency control. In *Proceedings of the 2003 Large Engineering Systems Conference on Power Engineering*, pages 12–16, Montreal, Canada, May 2003.

[127] C. B. Royo. *Generalized Unit Commitment by the Radar Multiplier Method*. PhD thesis, Technical University of Catalonia, Barcelona, Spain, May 2001.

[128] P. W. Sauer and M. A. Pai. *Power System Dynamics and Stability*. Prentice-Hall, London, UK, 1998.

[129] S. Sawadogo, R. M. Faye, P. O. Malaterre, and F. Mora-Camino. Decentralized predictive controller for delivery canals. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3380–3884, San Diego, California, 1998.

[130] N. I. Shaikh and V. Prabhu. Model predictive controller for cryogenic tunnel freezers. *Journal of Food Engineering*, 80(2):711–718, May 2007.

[131] M. G. Singh and A. Titli. *Systems Decomposition, Optimisation and Control*. Pergamon Press, Oxford, UK, 1978.

[132] J.-E. Skog, K. Koreman, and B. Pääjärvi. The Norned HVDC cable link – A power transmission highway between Norway and The Netherlands. In *Proceedings of Energex 2006*, Stavanger, Norway, June 2006.

[133] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, April 1981.

[134] K. Staňková, M. C. J. Bliemer, and G. J. Olsder. Inverse Stackelberg games and their application to dynamic bilevel optimal toll design problem. In *Proceedings of the 12th International symposium on dynamic games and applications*, Sophia Antipolis, France, July 2006.

[135] K. P. Sycara. Multiagent systems. *AI Magazine*, 2(19):79–92, 1998.

[136] The Modelica Association. Modelica - A Unified Object-Oriented Language for Physical Systems Modeling – Language specification. URL: http://www.modelica.org/documents/ModelicaSpec22.pdf, 2005.

[137] M. M. Thomas, J. L. Kardos, and B. Joseph. Shrinking horizon model predictive control applied to autoclave curing of composite laminate materials. In *Proceedings of the American Control Conference*, pages 505–509, Baltimore, Maryland, June 1994.

[138] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[139] UCTE. Final report of the investigation committee on the 28 September 2003 blackout in Italy. Technical report, Union for the Coordination of Transmission of Electricity (UCTE), Brussels, Belgium, 2003.

[140] UCTE. Final report system disturbance on 4 November 2006. Technical report, Union for the Coordination of Transmission of Electricity (UCTE), Brussels, Belgium, 2006.

[141] U.S.-Canada Power System Outage Task Force. Final report on the August 14, 2003 blackout in the United States and Canada: causes and recommendations. Technical report, April 2004.

[142] T. Van Cutsem and C. Vournas. *Voltage Stability of Electric Power Systems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[143] A. J. van der Schaft and J. M. Schumacher. *An Introduction to Hybrid Dynamical Systems*, volume 251 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, London, 2000.

[144] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed output feedback MPC for power system control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, California, December 2006.

[145] D. D. Šiljak. *Decentralized Control of Complex Systems*. Academic Press, Boston, Massachusetts, 1991.

[146] W. Wang, D. E. Rivera, and K. G. Kempf. Model predictive control strategies for supply chain management in semiconductor manufacturing. *International Journal of Production Economics*, 107(1):56–77, May 2007.

[147] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, Massachusetts, 2000.

[148] Wikipedia. List of famous wide-scale power outages. URL: http://en.wikipedia.org/wiki/List_of_power_outages, 2007.

[149] H. P. Williams. *Model Building in Mathematical Programming*. Wiley, New York, New York, 1993.

[150] M. H. Wright. Direct search methods: Once scorned, now respectable. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1995*, pages 191–208. Addison Wesley, Harlow, UK, 1996.

[151] T. C. Yang, H. Cimen, and Q. M. Zhu. Decentralised load-frequency controller design based on structured singular values. *IEE Proceedings Generation, Transmission and Distribution*, 145(1):7–14, January 1998.

[152] T. C. Yang, Z. T. Ding, and H. Yu. Decentralised power system load frequency control beyond the limit of diagonal dominance. *International Journal on Electrical Power Energy Systems*, 24(3):173–184, March 2002.

Samenvatting

Multi-Agent Modelgebaseerd Voorspellend Regelen met Toepassingen in Elektriciteitsnetwerken

Transportnetwerken, zoals elektriciteitsnetwerken, verkeersnetwerken, spoornetwerken, waternetwerken, etc., vormen de hoekstenen van onze moderne samenleving. Een soepele, efficiënte, betrouwbare en veilige werking van deze netwerken is van enorm belang voor de economische groei, het milieu en de leefbaarheid, niet alleen wanneer deze netwerken op de grenzen van hun kunnen moeten opereren, maar ook onder normale omstandigheden. Aangezien transportnetwerken dichter en dichter bij hun capaciteitslimieten moeten werken, en aangezien de dynamica van dergelijke netwerken alsmaar complexer wordt, wordt het steeds moeilijker voor de huidige regelstrategieën om adequate prestaties te leveren onder alle omstandigheden. De regeling van transportnetwerken moet daarom naar een hoger niveau gebracht worden door gebruik te maken van nieuwe geavanceerde regelstrategieën.

Elektriciteitsnetwerken vormen een specifieke klasse van transportnetwerken waarvoor nieuwe regelstrategieën in het bijzonder nodig zijn. De structuur van elektriciteitsnetwerken is aan het veranderen op verschillende niveaus. Op Europees niveau worden de elektriciteitsnetwerken van individuele landen meer en meer geïntegreerd door de aanleg van transportlijnen tussen landen. Op nationaal niveau stroomt elektriciteit niet langer alleen van het transmissienetwerk via het distributienetwerk in de richting van bedrijven en steden, maar ook in de omgekeerde richting. Daarnaast wordt op lokaal niveau regelbare belasting geïnstalleerd en kan energie lokaal gegenereerd en opgeslagen worden. Om minimumeisen en -serviceniveaus te kunnen blijven garanderen, moeten *state-of-the-art* regeltechnieken ontwikkeld en geïmplementeerd worden.

In dit proefschrift stellen wij verschillende regelstrategieën voor die erop gericht zijn om de opkomende problemen in transportnetwerken in het algemeen en elektriciteitsnetwerken in het bijzonder het hoofd te bieden. Om het grootschalige en gedistribueerde karakter van de regelproblemen te beheersen gebruiken wij *multi-agent* aanpakken, waarin verschillende regelagenten elk hun eigen deel van het netwerk regelen en samenwerken om de best mogelijke netwerkbrede prestaties te behalen. Om alle beschikbare informatie mee te kunnen nemen en om vroegtijdig te kunnen anticiperen op ongewenst gedrag maken wij gebruik van modelgebaseerd voorspellend regelen (MVR). In de regelstrategieën die wij in dit proefschrift voorstellen, combineren wij multi-agent aanpakken met MVR. Hieronder volgt een overzicht van de regelstrategieën die wij voorstellen en de regelproblemen uit de specifieke klasse van elektriciteitsnetwerken, waarop wij de voorgestelde regelstrategieën toepassen.

Multi-agent modelgebaseerd voorspellend regelen

In een multi-agent regeling is de regeling van een systeem gedistribueerd over verschillende regelagenten. De regelagenten kunnen gegroepeerd worden aan de hand van de autoriteitsrelaties die tussen de regelagenten gelden. Een dergelijke groepering resulteert in een gelaagde regelstructuur waarin regelagenten in hogere lagen meer autoriteit hebben over regelagenten in lagere lagen en waarin regelagenten in dezelfde laag dezelfde autoriteitsrelaties met betrekking tot elkaar hebben. Gebaseerd op de ideeën van MVR bepalen in multi-agent MVR de regelagenten welke actie zij nemen aan de hand van voorspellingen. Deze voorspellingen maken zij met behulp van voorspellingsmodellen van die delen van het algehele systeem die zij regelen. Daar waar de regelagenten in hogere lagen typisch minder gedetailleerde modelen en langzamere tijdschalen beschouwen, beschouwen regelagenten op lagere regellagen typisch meer gedetailleerde modelen en snellere tijdschalen. In dit proefschrift worden de volgende regelstrategieën voorgesteld en bediscussieerd:

- Voor de coördinatie van regelagenten in een regellaag wordt een nieuw serieel schema voor multi-agent MVR voorgesteld en vergeleken met een bestaand parallel schema. In de voorgestelde aanpak wordt aangenomen dat de dynamica van de deelnetwerken alleen uit continue dynamica bestaat en dat de dynamica van het algehele netwerk gemodelleerd kan worden met verbonden lineaire tijdsinvariante modellen, waarin alle variabelen continue waarden aannemen.

- In de praktijk komt het regelmatig voor dat deelnetwerken hybride dynamica vertonen, veroorzaakt door zowel continue als discrete dynamica. We bediscussiëren hoe discrete dynamica gevat kan worden in modellen bestaande uit lineaire vergelijkingen en ongelijkheden en hoe regelagenten dergelijke modellen kunnen gebruiken bij het bepalen van hun acties. Daarnaast stellen wij een uitbreiding voor van de coördinatieschema's voor continue systemen naar systemen met continue en discrete variabelen.

- Voor een individuele regelagent die richtpunten bepaalt voor regelagenten in een lagere regellaag wordt het opzetten van object-georiënteerde voorspellingsmodellen bediscussieerd. Een dergelijk object-georiënteerd voorspellingsmodel wordt dan gebruikt om een MVR-regelprobleem te formuleren. Wij stellen voor om de optimalisatietechniek *pattern search* te gebruiken om het resulterende MVR-regelprobleem op te lossen. Daarnaast stellen wij omwille van de efficiëntie een MVR-regelstrategie voor die gebaseerd is op een gelineariseerde benadering van het object-georiënteerde voorspellingsmodel.

- Regelmatig worden deelnetwerken gedefinieerd op basis van reeds bestaande netwerkregio's. Dergelijke deelnetwerken overlappen meestal niet. Als deelnetwerken echter gebaseerd worden op bijvoorbeeld invloedsgebieden van actuatoren, dan kunnen de deelnetwerken overlappend zijn. Wij stellen een regelstrategie voor voor het regelen van overlappende deelnetwerken door regelagenten in een hogere regellaag.

Multi-agent regelproblemen in elektriciteitsnetwerken

Elektriciteitsnetwerken vormen een specifieke klasse van transportnetwerken waarvoor de ontwikkeling van geavanceerde regeltechnieken noodzakelijk is om adequate prestaties te behalen. De regelstrategieën die in dit proefschrift worden voorgesteld worden daarom aan de hand van toepassing op specifieke regelproblemen uit elektriciteitsnetwerken geëvalueerd. In het bijzonder worden de volgende regelproblemen besproken:

- We beschouwen een gedistribueerd *load-frequency* probleem, wat het probleem is van het dicht bij nul houden van frequentie-afwijkingen na verstoringen. Regelagenten regelen elk hun eigen deel van het netwerk en moeten samenwerken om de best mogelijke netwerkbrede prestaties te behalen. Om deze samenwerking te bewekstellingen gebruiken de regelagenten de seriële of de parallele MVR-strategieën. We beschouwen zowel samenwerking gebaseerd op voorspellingsmodellen die alleen continue variabelen bevatten, als met gebruikmaking van voorspellingsmodellen die zowel continue als ook discrete variabelen bevatten. Met behulp van simulaties illustreren we de prestaties die de schema's kunnen behalen.

- In de nabije toekomst zullen huishoudens de mogelijkheid hebben om hun eigen energie lokaal te produceren, lokaal op te slaan, te verkopen aan een energie-aanbieder en mogelijk uit te wisselen met naburige huishoudens. We stellen een MVR-strategie voor die gebruikt kan worden door een regelagent die het energiegebruik in een huishouden regelt. Deze regelagent neemt in zijn regeling verwachte energieprijzen, voorspelde energieconsumptiepatronen en de dynamica van het huishouden mee. We illustreren de prestaties die de regelagent kan behalen voor een gegeven scenario van energieprijzen en consumptiepatronen.

- Spanningsinstabiliteiten vormen een belangrijke bron van elektriciteitsuitval. Om te voorkomen dat spanningsinstabiliteiten ontstaan is lokaal bij generatielokaties een laag van regelagenten geïnstalleerd. Een dergelijke lokale regeling werkt onder normale omstandigheden goed, maar levert ten tijde van grote verstoringen geen adequate prestaties. In dergelijke situaties moeten de acties van de lokale regelagenten gecoördineerd worden. Wij stellen een MVR-regelagent voor die tot taak heeft deze coördinatie te realiseren. De voorgestelde MVR-strategie maakt gebruik van ofwel een object-georiënteerd model van het elektriciteitsnetwerk ofwel van een benadering van dit model verkregen na linearisatie. We illustreren de prestaties die behaald kunnen worden met behulp van simulaties op een dynamisch 9-bus elektriciteitsnetwerk.

- Regeling gebaseerd op *optimal power flow* (OPF) kan gebruikt worden om in transmissienetwerken de *steady-state* spanningsprofielen te verbeteren, het overschrijden van capaciteitslimieten te voorkomen, en vermogensverliezen te minimaliseren. Een type apparaat waarvoor met behulp van OPF-regeling actuatorinstellingen bepaald kunnen worden zijn *flexible alternating current transmission systems* (FACTS). Wij beschouwen een situatie waarin verschillende FACTS-apparaten aanwezig zijn en elk FACTS-apparaat geregeld wordt door een regelagent. Elke regelagent beschouwt als zijn deelnetwerk dat deel van het netwerk dat zijn FACTS-apparaat kan beïnvloeden. Aangezien de deelnetwerken gebaseerd zijn op beïnvloedingsregio's kunnen verschillende deelnetwerken overlappend zijn. Wij stellen een coördinatie- en communicatieschema voor dat kan omgaan met een dergelijke overlap. Via simulatiestudies op een aangepast elektriciteitsnetwerk met 57 bussen illustreren we de prestaties.

Rudy R. Negenborn

Summary

Multi-Agent Model Predictive Control with Applications to Power Networks

Transportation networks, such as power distribution and transmission networks, road traffic networks, water distribution networks, railway networks, etc., are the corner stones of modern society. A smooth, efficient, reliable, and safe operation of these systems is of huge importance for the economic growth, the environment, and the quality of life, not only when the systems are pressed to the limits of their performance, but also under regular operating conditions. As transportation networks have to operate closer and closer to their capacity limits and as the dynamics of these networks become more and more complex, currently used control strategies can no longer provide adequate performance in all situations. Hence, control of transportation networks has to be advanced to a higher level using novel control techniques.

A class of transportation networks for which such new control techniques are in particular required are power networks. The structure of power networks is changing at several levels. At a European level the electricity networks of the individual countries are becoming more integrated as high-capacity power lines are constructed to enhance system security. At a national level power does not any longer only flow from the transmission network in the direction of the distribution network and onwards to the industrial sites and cities, but also in the other direction. Furthermore, at the local level controllable loads are installed, energy can be generated locally with small-scale generators, and energy can be stored locally using batteries. To still guarantee basic requirements and service levels and to meet the demands and requirements of the users while facing the changing structure of power networks, state-of-the-art control techniques have to be developed and implemented.

In this PhD thesis we propose several new control techniques designed for handling the emerging problems in transportation networks in general and power networks in particular. To manage the typically large size and distributed nature of the control problems encountered, we employ multi-agent approaches, in which several control agents each control their own part of the network and cooperate to achieve the best possible overall performance. To be able to incorporate all available information and to be able to anticipate undesired behavior at an early stage, we use model predictive control (MPC).

Next we give a summary of the control techniques proposed in this PhD thesis and the control problems from a particular class of transportation networks, viz. the class of power networks, to which we apply the proposed control techniques in order to assess their

performance.

Multi-agent model predictive control

In multi-agent control, control is distributed over several control agents. The control agents can be grouped according to the authority relationships that they have among each other. The result is a layered control structure in which control agents at higher layers have authority over control agents in lower layers, and control agents within a control layer have equal authority relationships. In multi-agent MPC, control agents take actions based on predictions that they make using a prediction model of the part of the overall system they control. At higher layers typically less detailed models and slower time scales are considered, whereas at lower layers more detailed models and faster time scales are considered.

In this PhD thesis the following control strategies for control agents at various locations in a control structure are proposed and discussed:

- For coordination of control agents within a control layer a novel serial scheme for multi-agent MPC is proposed and compared with an existing parallel scheme. In the approach it is assumed that the dynamics of the subnetworks that the control agents control are purely continuous and can be modeled with interconnected linear discrete-time time-invariant models in which all variables take on continuous values.

- In practice, the dynamics of the subnetworks may show hybrid dynamics, caused by both continuous and discrete dynamics. We discuss how discrete dynamics can be captured by systems of linear equalities and inequalities and how control agents can use this in their decision making. In addition, we propose an extension of the coordination schemes for purely continuous systems that deals with interconnected linear time-invariant subnetworks with integer inputs.

- For an individual control agent that determines set-points for control agents in a lower control layer, creating object-oriented prediction models is discussed. Such an object-oriented prediction model is then used to formulate an MPC control problem. We propose to use the optimization technique pattern search to solve the resulting MPC control problem. In addition, for efficiency reasons, we propose an MPC control strategy based on a linearization of the object-oriented prediction model.

- Commonly, subnetworks are defined based on already existing network regions. Such subnetworks typically do not overlap. However, when subnetworks are based on, e.g., regions of influence of actuators, then the subnetworks may be overlapping. For multiple control agents in a higher control layer, at which it can be assumed that the behavior of the underlying control layers is static, we propose an MPC strategy for control of overlapping subnetworks.

Multi-agent control problems in power networks

Power networks are a particular class of transportation networks and are subject to a changing structure. This changing structure requires the development of advanced control techniques in order to maintain adequate control performance. The control strategies proposed

in this PhD thesis are applied to and assessed on specific power domain control problems. In particular, we discuss the following power network problems and control approaches:

- We consider a distributed load-frequency control problem, which is the problem of maintaining frequency deviations after load disturbances close to zero. Control agents each control their own part of the network and have to cooperate in order to achieve the best possible overall network performance. The control agents achieve this by obtaining agreement on how much power should flow among the subnetworks. The serial and parallel MPC strategies are employed for this, both when the prediction models involve only continuous variables, and when the prediction models involve both continuous and discrete variables. In simulations we illustrate the performance that the schemes can obtain.

- In the near future households will be able to produce their own energy, store it locally, sell it to an energy supplier, and perhaps exchange it with neighboring households. We propose an MPC strategy to be used by a control agent controlling the energy usage in a household. This control agent takes into account expected energy prices, predicted energy consumption patterns, and the dynamics of the household, including dynamics of local energy generation and storage devices. For a given scenario of energy prices and consumption patterns, the performance that the control agent can achieve are illustrated.

- Voltage instability is a major source of power outages. To prevent voltage instability from emerging, a lower layer of control agents is installed in power networks at generation sites. These agents locally adjust generation to maintain voltage magnitudes. Such local control works well under normal operating conditions. However, under large disturbances such local control does not provide adequate performance. In such situations, the actions of the local control agents have to be coordinated. We propose an MPC control agent that has the task to coordinate the local control agents. The MPC strategy that the agent uses is based on either an object-oriented model of the power network or on a linearized approximation of this model. The object-oriented model includes a model of the physical network and the local control agents. We illustrate the performance of the MPC control agent using the object-oriented model or the linearized approximation via simulations on a dynamic 9-bus power network.

- Optimal power flow control is commonly used to improve steady-state power network security by improving the voltage profile, preventing lines from overloading, and minimizing active power losses. Using optimal power flow control, device settings for flexible alternating current transmission systems (FACTS) can be determined. We consider the situation in which there are several FACTS devices, each controlled by a different control agent. The subnetwork that each control agent considers consists of a region of influence of its FACTS device. Since the subnetworks are based on regions of influence, the subnetworks of several agents may be overlapping. We propose a coordination and communication scheme that takes this overlap into account. In simulation experiments on an adjusted 57-bus IEEE power network the performance of the scheme is illustrated.

Rudy R. Negenborn

Curriculum vitae

Rudy R. Negenborn was born on June 13, 1980 in Utrecht, The Netherlands. He finished his pre-university education (*VWO*) in 1998 at the Utrechts Stedelijk Gymnasium, Utrecht, The Netherlands. After this, Rudy Negenborn started his studies in Computer Science at the Utrecht University, Utrecht, The Netherlands. He received the title of *doctorandus* (comparable with Master of Science) in Computer Science, with a specialization in Intelligent Systems, *cum laude* from this university in 2003. For his graduation project, he performed research on Kalman filtering and robot localization. The research involved in this project was carried out during a one-year visit to the Copenhagen University, Denmark, and was supervised by Prof.Dr.Phil. P. Johansen and Dr. M. Wiering.

Since 2004, Rudy Negenborn has been working on his PhD project at the Delft Center for Systems and Control of Delft University of Technology, The Netherlands. The research of his PhD project has been on multi-agent model predictive control with applications to power networks, and has been supervised by Prof.dr.ir. B. De Schutter and Prof.dr.ir. J. Hellendoorn. During his PhD project, Rudy Negenborn obtained the DISC certificate for fulfilling the course program requirements of the Dutch Institute for Systems and Control. Furthermore, he cooperated with and spent time at various research groups, including the Hybrid System Control Group of Supélec, Rennes, France, and the Power Systems Laboratory and Automatic Control Laboratory of ETH Zürich, Zürich, Switzerland.

Rudy Negenborn's more fundamental research interests include multi-agent systems, hybrid systems, distributed control, and model predictive control. His more applied research interests include applications to transportation networks in general, and power networks in particular.

Since 2004, Rudy Negenborn has been a member of the DISC and of The Netherlands Research School for Transport, Infrastructure, and Logistics (TRAIL). Moreover, from 2004 until 2007, Rudy Negenborn fulfilled the positions of public relations representative and treasurer in the board of Promood, the representative body of the PhD candidates at Delft University of Technology.