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 4

Multi-layer control using MPC

In the previous chapters we have discussed common issues arising due to the nature of large-scale transportation networks. In those chapters we have focused on particular issues in single-layer control, i.e., control in which control agents have equal authority relationships with respect to one another and control dynamics that take place at similar time scales. In this chapter we consider particular issues involved in multi-layer control, i.e., control in which control agents of higher control layers have authority over control agents in lower control layers, and control agents in higher control layers typically control dynamics at lower time scales. In Section 4.1 we introduce multi-layer MPC control for transportation networks, and in particular discuss how the prediction models that the control agents use can to be constructed. In Section 4.2 we discuss prediction models constructed for a higher-layer control agent using object-oriented modeling, which is suited for making prediction models of large-scale systems, and prediction models derived from such object-oriented models by linearization. We formulate an MPC problem based on such an object-oriented prediction model in Section 4.3. As we will see, the MPC problem based on the object-oriented prediction model leads to a nonconvex MPC problem, with an objective function that is expensive to evaluate. We consider two approaches for addressing this issue: i) the nonlinear MPC optimization problem is solved directly, using pattern search as solver; ii) a linearized approximation of the nonlinear optimization problem is solved, using an efficient linear programming solver.

In this chapter we consider as application emergency voltage control. In Section 4.4 we develop an object-oriented model of a 9-bus dynamic power network and experimentally assess the performance of the proposed approaches on an emergency voltage control problem.

Parts of this chapter have been published in [110] and presented in [113].

4.1 Multi-layer control of transportation networks

As we have discussed in Chapter 1, there are several characteristics of transportation networks that make their control challenging. In Chapters 2 and 3, we have discussed how to deal with the large geographical region and the hybrid dynamics that transportation networks typically have. In this chapter, we discuss how to deal with the wide range of time scales over which the dynamics of transportation networks typically evolve. Multi-layer control can be used for this.

4.1.1 Multi-layer control

If dynamics evolve over a wide range of time scales, then control of such dynamics can be done using multiple control agents that each consider a particular range of time scales. The control agents can be grouped into layers depending on the time scales they control.

Figure 4.1 illustrates multi-layer control of a network, in which the control structure consists of a higher, medium, and lower control layer. At lower layers, control agents that control faster dynamics are located. The faster dynamics will typically require faster control, hence at the lower control layers the time available to determine control actions is relatively small. However, to adequately describe the fast dynamics, more detailed dynamics have to be considered. Therefore at lower control layers, typically, more localized models of the dynamics will be used. Control agents that control slower dynamics are located at higher control layers. There more time is available to determine actions. However, the slower dynamics considered at the higher layers will typically involve larger regions of the network. Therefore, at higher control layers less detailed models are used. The result is a multi-layer or hierarchical control structure in which control takes place at different control layers based on space and/or time division [17]. The higher-layer control agents determine both actions to be implemented directly in the physical network, and set points to be provided to the control agents in a lower control layer. Hence, control agents in higher control layers can be seen as supervisory control agents.

In principle each control layer can consist of multiple control agents, each controlling their own group of control agents in a lower control layer. Communication among the control agents in each layer may or may not be present.

4.1.2 Multi-layer control in power networks

As an example of the multi-layer control of transportation networks, we consider power networks. Power networks in general are controlled using multi-layer control in which control of the physical network is the result of the joint effort of several control layers at local, regional, national, and sometimes international level [45, 60]. The physical power network consists of multiple interconnected subsystems, like generators, loads, transmission lines, etc. This physical network is controlled by several control layers in order to control the network in a desired way. The lowest control layer consists of control agents that locally control the actuators in the physical network. The higher control layers consists of control agents that determine actions and set-points for lower control layers. The set-points can be used to obtain coordination between the control agents of the lower control layers. The higher control layers typically consist of, e.g., regional or national human network operators. These human operators decide on the actions to take based on offline studies, experience, heuristics, knowledge bases, and actual system conditions obtained via telemetry or obtained from state estimators and soft sensors. The set-points should be determined in such a way that objectives defined for the higher control layer are achieved [100, 131]. The higher control layer hereby typically takes into account nonlinear behavior of the system, behavior that may be neglected by lower control layers.

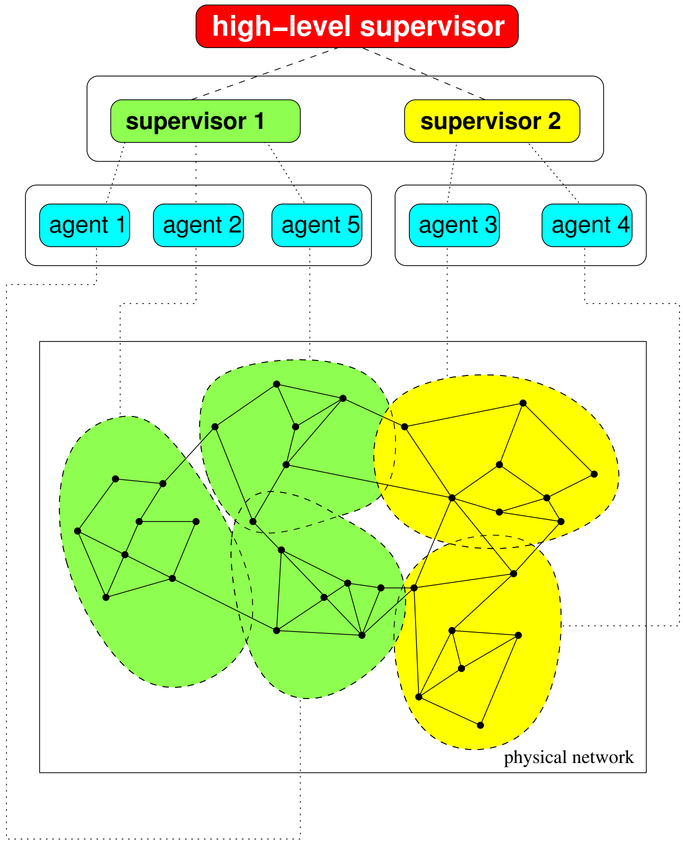

Figure 4.1: Illustration of multi-layer control. A higher layer provides set-points to a lower layer (dashed lines). The lower layer controls the actuators in the physical system (dotted lines).

It is in general not possible to rapidly change the set-points used by a lower control layer in an online and coordinated manner to achieve improved performance [45]. As it becomes more complex for human operators to adequately predict the consequences of faults and disturbances in the network (e.g., for power networks, due to deregulation of the energy market, the increase in power demands, and the emergence of embedded generation [73]), the need for intelligent automatic online control systems increases. These automatic control systems can be used to determine which set-points should be provided, at a first stage outside the control loop in the form of a decision support system, and at a later stage inside the control loop in the form of closed-loop control.

4.1.3 MPC in multi-layer control

Although in general there can be many control layers, and each control layer can consist of multiple control agents, in this chapter we restrict our focus to two control layers, a medium and a lower control layer. The medium control layer consists of a single control agent, and the lower control layer consists of multiple decentralized control agents. In Chapter 5 we consider the control by a higher control layer that consists of multiple control agents that can communicate with one another.

To be able to obtain its control objectives, a control agent in a particular layer has to monitor the current state of the part of the lower control layer of its interest and the underlying physical network. Based on this, the control agent has to foresee when the behavior of the system is going into an undesirable direction such that it can provide adequate set-points to that part of the lower control layer that it considers. We propose a medium-layer control agent that at each control cycle uses MPC to determine which set-points to provide to the lower control layer.

In order for the medium-layer MPC control agent to meet its control objectives, it has to be able to predict how set-point changes influence the dynamics of the network. The performance of the control agent relies for a large part on the accuracy of the prediction model that it uses. The prediction model has to describe well how the actions of the control agent affect the behavior of the network and the lower-layer control agents. Ideally, the control agent should have a model of the complete dynamics of the network, including the behavior of the other control agents. However, such an ideal model can be very complex or impossible to construct, thus making the optimization procedure in the control agent slow or impossible. Instead, the control agent has to use an approximation of the model. If this approximation fits in a suitable form, relatively efficient optimization techniques can be used to determine the actions to take (e.g., linear or mixed-integer linear programming).

Suppose that the dynamics of the transportation network can be represented by a system of ordinary differential equations (ODEs) as:

$$\begin{bmatrix} \frac{\mathrm{d}\mathbf{x}_{\text{very slow}}}{\mathrm{d}t}(t) \\ \frac{\mathrm{d}\mathbf{x}_{\text{slow}}}{\mathrm{d}t}(t) \\ \frac{\mathrm{d}\mathbf{x}_{\text{fast}}}{\mathrm{d}t}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{very slow}}(\mathbf{x}_{\text{very slow}}(t), \mathbf{x}_{\text{slow}}(t), \mathbf{x}_{\text{fast}}(t)) \\ \mathbf{f}_{\text{slow}}(\mathbf{x}_{\text{very slow}}(t), \mathbf{x}_{\text{slow}}(t), \mathbf{x}_{\text{fast}}(t)) \\ \mathbf{f}_{\text{fast}}(\mathbf{x}_{\text{very slow}}(t), \mathbf{x}_{\text{slow}}(t), \mathbf{x}_{\text{fast}}(t)) \end{bmatrix},$$

where the dynamics have been grouped into "very slow", "slow", and "fast dynamics". Suppose that the medium-layer control agent has as objective to control the slow dynamics only. The question is whether and how this control agent has to take into account the very slow and the fast dynamics. Although the control agent is not directly interested in the very slow and fast dynamics, these dynamics can influence the slow dynamics in which the control agent is interested. Simply ignoring the very slow and the fast dynamics may lead to unacceptable loss of model accuracy. Instead of ignoring the very slow and the fast dynamics completely, the control agent can approximate the very slow dynamics with constants, and the fast dynamics with instantaneous dynamics. The model that the control agent then considers can be described as:

$$\begin{bmatrix} \frac{\mathrm{d}\mathbf{x}_{\text{very slow}}}{\mathrm{d}t}(t) \\ \frac{\mathrm{d}\mathbf{x}_{\text{slow}}}{\mathrm{d}t}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}_{\text{slow}}(\mathbf{x}_{\text{very slow}}(t), \mathbf{x}_{\text{slow}}(t), \mathbf{x}_{\text{fast}}(t)) \\ \mathbf{f}_{\text{fast}}(\mathbf{x}_{\text{very slow}}(t), \mathbf{x}_{\text{slow}}(t), \mathbf{x}_{\text{fast}}(t)) \end{bmatrix}, \tag{4.1}$$

which constitutes a system of differential-algebraic equations (DAEs). Note that with respect to the fast dynamics a model such as discussed in Chapter 5 emerges. Note also that the very slow dynamics can include changes in set-points used by the medium-layer control agent. The medium-layer control agent can receive updates from higher-layer control agents with respect to the very slow dynamics that it assumes constant, including the set-points, and it can in addition determine the set points for the lower-layer control agents, e.g., in such a way that the objectives related to its time scale dynamics are achieved.

Constructing the prediction model

Due to the complexity of transportation networks, constructing appropriate prediction models that can be used in control of these networks is a difficult task. Constructing a model implementing (4.1) involves formalizing many components, differential equations, algebraic equations, mixed continuous and discrete elements, and dynamics at different time scales. Over the last decade modeling languages and simulation environments have been introduced that allow general-purpose physical modeling based on acausal modeling, mixing physical modeling using equations with the use of *object-oriented* constructs, and therewith significantly easing the development of such complex prediction models [13, 39, 99, 122]. In the next section we discuss object-oriented modeling and its use for constructing object-oriented prediction models implementing models such as (4.1). In addition, we discuss how prediction models approximating these object-oriented prediction models can be derived using linearization. These models will then be used in Section 4.3 for setting up MPC control problems.

4.2 Constructing prediction models with object-oriented modeling

4.2.1 Object-oriented modeling

To face the difficulty of constructing models of complex systems, object-oriented approaches for analysis and simulation of such networks have received increasing attention [95]. In object-oriented modeling, the structure of models of complex systems are determined by defining objects for subsystems in these complex systems. The objects are used to map the structure of the model as closely as possible to the structure of the system. The objects are described in a declarative way by defining only local equations of objects and the connections between the objects. To facilitate modeling, an object-oriented approach for modeling offers inheritance and composition concepts. Inheritance offers the possibility to form new classes of objects using classes that have already been defined. The new classes take over or inherit attributes and behavior, e.g., dynamics, of the already existing classes. Extended models can then be constructed by inheriting dynamics and properties of more basic or more general models. E.g., for power networks, advanced generator objects are designed in this way by extending a basic generator objects that only contains the basic dynamics of a synchronous machine. Composition offers the possibility to combine simple objects into more complex ones. E.g., for power networks, when composing an object of a voltage regulator and an object of a turbine governor with an object of a basic generator, an object for a regulated generator with complex dynamics is obtained. Objected-oriented concepts enable proper structuring of models and generally lead to more flexible, modular, and reusable models.

4.2.2 Modeling tools

Several object-oriented approaches have been developed over the years, e.g., [13, 39, 99, 122, 136]. The approaches typically support both high-level modeling by composition and detailed component modeling using equations. Models of system components are typically organized in model libraries. A component model may be a composite model to support hierarchical modeling and may specify the system topology in terms of components and connections between the components. Using a graphical model editor, e.g., Dymola [39], a model can be defined by drawing a composition diagram, by simply positioning icons that represent the models of the components and drawing connections between the icons. Parameter values of the underlying models are then conveniently specified in dialog boxes.

Most of the object-oriented simulation software packages assume that a system can be decomposed into objects with fixed causal relations [7]. Causal relations are relations between causes and effects. E.g., if there is a causal relationship between two objects A and B, then this means that if the variables of object A change, that then the variables of object B change as a consequence of the change of the variables of object A. In a fixed causal relations this behavior is defined in one direction only. Hence, for objects A and B, the variables of object A do not change as a consequence of changes in variables of object B. In general, causality implies that the model of the system can be expressed as the interconnection of objects with an explicit state-space representation, in which algebraic relations as in (4.1) cannot be present. Often a significant effort in terms of analysis and analytical transformations is required to obtain a model in this form [39], in particular for systems in which causality is not naturally present, as is the case, e.g., in power networks. Setting the causality in a voltage-current formulation would mean that currents are expressed as function of voltages, or vice versa. Acausal modeling permits to relax the causality constraint and allows to focus on the elements and the way these elements are connected to each other, i.e., the system's topology. An environment that allows acausal modeling, is Dymola [39], which implements the object-oriented modeling language Modelica [136]. In Section 4.4 we will develop an object-oriented Modelica model for power networks using Dymola.

4.2.3 Object-oriented prediction models

Using an object-oriented modeling approach, each of the objects of a transportation network can be modeled with a mixture of differential equations, algebraic equations, and discrete logic. The model of the overall system then consists of the models for the objects and in addition algebraic equations interconnecting the individual objects.

For the object-oriented model to be useful as a prediction model that can be used by an MPC control agent, a method has to be available that can evaluate the model over a time horizon from time t_0 until t_f given the initial state of the system at time t_0. So, it should be possible to solve a so-called initial value problem that given the initial states $\mathbf{x}(t_0) \in \mathbb{R}^{n_\mathbf{x}}$, the initial inputs $\mathbf{u}(t_0) \in \mathbb{R}^{n_\mathbf{z}}$, and inputs $\mathbf{u}(t)$ specified over the full time interval, computes the outputs $\mathbf{y}(t) \in \mathbb{R}^{n_\mathbf{y}}$, for $t \in [t_0, t_\mathrm{f}]$.

Note that a medium-layer control agent in fact does not provide set-points to a lower control layer continuously, but only at discrete control cycles k_c, for $k_\mathrm{c} = \{0, 1, \ldots\}$, where control cycle k_c corresponds to continuous time $k_\mathrm{c} T_\mathrm{c}$, with T_c the control cycle time in continuous time units, as shown in Figure 4.2. A zero-order hold is used to make the trans-

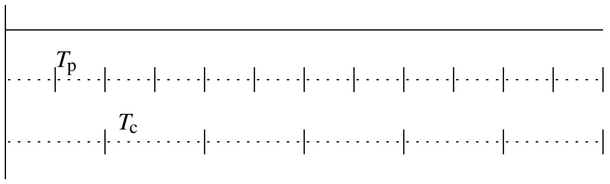

Figure 4.2: Overview of different time scales.

formation between the continuous-time input signal $\mathbf{u}(t)$ and the discrete-time input signal $\mathbf{u}(k_c)$. So, $\mathbf{u}(k_c) = \mathbf{u}_{k_c}$ becomes in continuous time:

$$\mathbf{u}(t) = \mathbf{u}_{k_c}, \quad \text{for } t \in [k_c T_c, (k_c+1)T_c).$$

Therefore, instead of specifying the continuous-time input signal $\mathbf{u}(t)$ to the prediction model M, a sequence of N_c inputs is specified to the prediction model M. The N_c inputs are collected in $\tilde{\mathbf{u}}(k_c)$ as $[(\mathbf{u}(k_c))^T, \ldots, (\mathbf{u}(k_c+N_c-1))^T]^T$, where $N_c = \frac{t_f - t_0}{T_c} + 1$ is the length of the prediction horizon in control cycles, and where for the sake of simplicity it is assumed that $t_f - t_0$ is an integer multiplier of T_c.

In general there is no analytic expression for the solution of the initial value problem. Instead, the trajectories of the variables of interest have to be approximated by numerical means to obtain values for these variables at discrete points in time. For control purposes we are typically interested in the outputs $\mathbf{y}(t)$. Assume that computing a sample of the continuous-time output $\mathbf{y}(t)$ for every T_p time units is sufficient to adequately represent the underlying continuous signals, where T_p is the length of one discrete time step, as illustrated in Figure 4.2. We then define the prediction horizon with a length $N_p = \frac{t_f - t_0}{T_p} + 1$ in discrete time steps, where for the sake of simplicity it is assumed that $t_f - t_0$ is an integer multiplier of T_p. We denote the outputs over the prediction horizon with length N_p by $\tilde{\mathbf{y}}(k_p) = [\mathbf{y}(k_p)^T, \ldots, \mathbf{y}(k_p+N_p-1)^T]^T$, where discrete time step $k_p = 0$ corresponds to continuous time $t = 0$ and discrete time step $k_p + l$ corresponds to continuous time $(k_p + l)T_p$.

Transition between t, T_p, and T_c

Below the notations $\mathbf{v}(t)$, $\mathbf{v}(k_p)$, and $\mathbf{v}(k_c)$, for some variables $\mathbf{v}$ each have to be interpreted in their own way. The notation $\mathbf{v}(t)$ refers to the variables $\mathbf{v}$ defined at continuous time t, the notation $\mathbf{v}(k_p)$ refers to the variables $\mathbf{v}$ defined at discrete time steps k_p, and the notation $\mathbf{v}(k_c)$ refers to the variables $\mathbf{v}$ defined at control cycle k_c. In particular, if the continuous-time signal $\mathbf{y}(t)$ is sampled with a sample size T_p, the signal $\mathbf{y}(T_p)$ is obtained. If the continuous-time signal $\mathbf{y}(t)$ is sampled with a sample size T_c, the signal $\mathbf{y}(T_c)$ is obtained. The variables $\tilde{\mathbf{x}}(t)$, and $\tilde{\mathbf{y}}(t)$ can be transitioned in a similar way. Furthermore, if the control inputs at control cycle $\mathbf{u}(k_c)$ are subjected to a zero-order hold, the signals $\mathbf{u}(k_p)$ and $\mathbf{u}(t)$ can be obtained. The variables $\tilde{\mathbf{u}}(k_c)$ can be transitioned in a similar way. The zero-order hold for the control inputs to make the transition between $\mathbf{u}(k_c)$ and $\mathbf{u}(k_p)$ can be implemented as:

$$\mathbf{u}(k_p + l + l_2) = \mathbf{u}(k_p + l), \text{for } l = \{0, L, 2L, \ldots, N_c - 1\}, \text{and } l_2 = \{1, 2, \ldots, L-1\}, \quad (4.2)$$

where $L = \frac{N_p}{N_c}$, and where $\mathbf{u}(k_p + l)$ at discrete time $k_p + l$ corresponds to $\mathbf{u}(k_c + \frac{l}{L})$ at control cycle $k_c + \frac{l}{L}$.

General object-oriented prediction model

Given the previous considerations, in the following we assume without loss of generality that the object-oriented prediction model of the transportation network is given by the mapping:

$$\tilde{\mathbf{y}}(k_{\mathrm{p}}) = M\left(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \tilde{\mathbf{u}}(k_{\mathrm{c}})\right), \tag{4.3}$$

where the prediction model M maps the initial states $\bar{\mathbf{x}} = \mathbf{x}(k_{\mathrm{c}})$, the previous inputs $\bar{\mathbf{u}} = \mathbf{u}(k_{\mathrm{c}} - 1)$, and the N_{c} inputs collected in $\tilde{\mathbf{u}}(k_{\mathrm{c}})$ to the N_{p} outputs collected in $\tilde{\mathbf{y}}(k_{\mathrm{p}})$. The prediction model thus includes the procedure to perform the time-domain simulation of the object-oriented model.

Remark 4.1 Here we have assumed that the initial derivatives $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(k_{\mathrm{c}})$ and initial algebraic variables $\mathbf{z}(k_{\mathrm{c}})$ can be uniquely determined when $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are given. If this is not the case, then the initial derivatives and algebraic variables have to be provided to the prediction model M as well. □

A transformed prediction model

For the interconnected individual objects modeled with differential equations, algebraic equations, and discrete-event logic, there is no direct initial value problem solver. However, the object-oriented model can be transformed into a system of synchronous differential, algebraic, and discrete equations [39], leading to deterministic behaviour and automatic synchronization of the continuous and discrete parts of the model. The continuous dynamics are modeled using a system of DAEs. For handling discrete event dynamics, the synchronous data flow principle is employed [44]. The idea of this principle is that at each time instant all active equations have to be fulfilled concurrently. The active equations at a particular time instant consist of those equations representing the continuous dynamics at that time and possibly the equations related to the discrete events at that time [118].

If no discrete events would be present, and thus only a purely continuous system of DAEs is considered, a time domain simulation can be performed using the DAE solver DASSL [26, 121]. DASSL implements a variable integration step and variable order version of the backward differentiation formula [121]. Due to the variable integration step size, DASSL is in particular suited for performing simulations of dynamics involving fast and slow dynamics. Variable step size methods are well-suited for such dynamics, since these methods automatically choose a larger step size when no fast dynamics are present, and a smaller step size when they are [26]. The solver uses a predictor-corrector scheme. First, the predictor makes a guess of the solution at a new integration point. Then, the corrector determines the final solution by solving a system of algebraic equation, which is obtained after substituting the derivative with the backward differentiation formula. To use DASSL, the functions of the system of DAEs have to be specified. The Jacobian of this system of DAEs, which is used in the solution of the system of DAEs, can be supplied as a function, or it can be approximated numerically by DASSL.

To be able to adequately handle the discrete events present in the systems of our model, the solver DASSL-RT can be used. DASSL-RT is an extended version of the DASSL solver, including a root finder [121, 124]. The root finder is necessary to allow efficient simulation of the discrete events. The root finder checks mathematical indicator expressions that indicate when discrete events should be simulated. These indicator expressions are given in the same variables as the dynamics, and will therefore change values when the dynamics are simulated. If one of the indicator expressions changes sign during the simulation, the root finder will back track the solution until the time instance when the indicator expression is equal to zero. The values of the simulation at that time will be returned. At event instants mixed continuous and discrete systems of equations are then solved to determine new values for the discrete variables and possibly the continuous variables.

4.2.4 Linearized object-oriented prediction models

The prediction model M in (4.3) typically is nonlinear and non-smooth, involving the numerical solution of systems of DAEs in combination with discrete logic. Therefore, computing the predictions is a costly process. This will have its effect on the time required to compute control actions. Instead of using the object-oriented prediction model directly, we can also try to derive an approximate prediction model from the object-oriented prediction model. This will result in optimization problems that are more efficient to solve.

One way to approximate the object-oriented prediction model is by deriving a discrete-time linearized prediction model from the continuous-time dynamics represented in the system of DAEs, assuming small variations of the variables around the operation point for which the model is linearized. At each control cycle k_c, corresponding to continuous time $k_\mathrm{c}T_\mathrm{c}$ the continuous-time linearization for the system of DAEs:

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t)) \\
0 &= \mathbf{g}(\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t)) \\
y(t) &= \mathbf{h}(\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t)),
\end{aligned}$$

around $\bar{\mathbf{x}} = \mathbf{x}(k_\mathrm{c})$, $\bar{\mathbf{u}} = \mathbf{u}(k_\mathrm{c}-1)$, $\bar{\mathbf{z}} = \mathbf{z}(k_\mathrm{c})$, and $\bar{\mathbf{y}} = \mathbf{y}(k_\mathrm{c})$ is given by the system:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t) = \mathbf{A}_\mathrm{c}\mathbf{x}(t)+\mathbf{B}_\mathrm{c}\mathbf{u}(t)+\mathbf{F}_\mathrm{c} \tag{4.4}$$

$$\mathbf{z}(t) = \mathbf{C}_{\mathrm{c},z}\mathbf{x}(t)+\mathbf{D}_{\mathrm{c},z}\mathbf{u}(t)+\mathbf{G}_{\mathrm{c},z} \tag{4.5}$$

$$\mathbf{y}(t) = \mathbf{C}_{\mathrm{c},y}\mathbf{x}(t)+\mathbf{D}_{\mathrm{c},y}\mathbf{u}(t)+\mathbf{G}_{\mathrm{c},y}, \tag{4.6}$$

where

$$\begin{aligned}
\mathbf{A}_\mathrm{c} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) + \frac{\partial \mathbf{f}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1}\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right) \\
\mathbf{B}_\mathrm{c} &= \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) + \frac{\partial \mathbf{f}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1}\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \\
\mathbf{C}_{\mathrm{c},z} &= \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1}\frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \\
\mathbf{D}_{\mathrm{c},z} &= \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1}\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})
\end{aligned}$$

$$\begin{aligned}
\mathbf{C}_{c,y} &= \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) + \frac{\partial \mathbf{h}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\\
\mathbf{D}_{c,y} &= \frac{\partial \mathbf{h}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) + \frac{\partial \mathbf{h}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\\
\mathbf{F}_{c} &= -\frac{\partial \mathbf{f}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1} \left(-\mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{z}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{x}} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{u}}\right.\\
&\quad \left. + \frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{z}}\right) - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{u}} + \frac{\partial \mathbf{f}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{z}} - \mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{z}})\right)\\
\mathbf{G}_{c,y} &= -\left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1} \left(-\mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{z}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{x}} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{u}} + \frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{z}}\right)\\
\mathbf{G}_{c,y} &= -\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{z}} + \frac{\partial \mathbf{h}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{u}} - \mathbf{h}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)\\
&\qquad -\frac{\partial \mathbf{h}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}}) \left(-\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\right)^{-1} \left(-\mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{z}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{x}} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{u}}\right.\\
&\qquad \left. + \frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})\bar{\mathbf{z}}\right),
\end{aligned}$$

when $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}(\bar{\mathbf{x}},\bar{\mathbf{z}},\bar{\mathbf{u}})$ is invertible. The required Jacobians can either be derived analytically [83] or computed numerically. Using the modeling tool Dymola, the linearized model of the object-oriented model is conveniently obtained using symbolic differentiation.

Remark 4.2 It is assumed that initial algebraic variables $\mathbf{z}(k_c)$ can be uniquely determined given $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$. If this is not the case, then $\mathbf{z}(k_c)$ should be specified to the prediction model M. □

Remark 4.3 The linearized prediction model can give adequate approximations when the discrete dynamics do not have a too large impact on the dynamics, and the changes in the continuous values are not too large. If the variations are not small, mode changes have to be considered in the model, e.g., by using piecewise affine or similar models [83]. □

The continuous-time linearization can be discretized with the sampling interval T_p, to obtain the following discrete-time linearized model in the affine expressions of $\mathbf{x}(k_p)$, $\mathbf{u}(k_p)$, and $\mathbf{y}(k_p)$:

$$\mathbf{x}(k_p+1) = \mathbf{A}\mathbf{x}(k_p) + \mathbf{B}\mathbf{u}(k_p) + \mathbf{F} \tag{4.7}$$

$$\mathbf{y}(k_p) = \mathbf{C}\mathbf{x}(k_p) + \mathbf{D}\mathbf{u}(k_p) + \mathbf{G}, \tag{4.8}$$

where k_p denotes the discrete time step, and where

$$\begin{aligned}
\mathbf{A} &= e^{\mathbf{A}_c T_p}\\
\mathbf{B} &= \int_0^{T_p} e^{\mathbf{A}_c \tau} \mathrm{d}\tau \mathbf{B}_c\\
\mathbf{F} &= \int_0^{T_p} e^{\mathbf{A}_c \tau} \mathrm{d}\tau \mathbf{F}_c
\end{aligned}$$

$$\mathbf{C} = \mathbf{C}_{\mathrm{c},y}$$
$$\mathbf{D} = \mathbf{D}_{\mathrm{c},y}$$
$$\mathbf{G} = \mathbf{G}_{\mathrm{c},y}.$$

The value of T_{p} determines how well the dynamics of the discrete-time model approximate the dynamics of the continuous-time linearized model (4.4)–(4.6). With a smaller value for T_{p} the approximation will be more accurate than with a larger value for T_{p}. However, with a smaller value for T_{p} the number of variables over a prediction horizon will become larger, which yields increased computational requirements for performing a simulation over a prediction horizon.

The discrete-time prediction model for $\tilde{\mathbf{x}}(k_{\mathrm{p}}+1)$ over a prediction horizon with length N_{p} discrete time steps is given by:

$$\tilde{\mathbf{x}}(k_{\mathrm{p}}+1) = \begin{bmatrix} \mathbf{A} & & & \\ & \mathbf{A} & & \\ & & \ddots & \\ & & & \mathbf{A} \end{bmatrix} \tilde{\mathbf{x}}(k_{\mathrm{p}}) + \begin{bmatrix} \mathbf{B} & & & \\ & \mathbf{B} & & \\ & & \ddots & \\ & & & \mathbf{B} \end{bmatrix} \tilde{\mathbf{u}}(k_{\mathrm{p}}) + \begin{bmatrix} \mathbf{F} \\ \mathbf{F} \\ \vdots \\ \mathbf{F} \end{bmatrix},$$

where the empty entries represent blocks of zeros. Substituting the expression for $\mathbf{x}(k_{\mathrm{p}}+l-1)$ in the expression for $\mathbf{x}(k_{\mathrm{p}}+l)$, for $l = \{1,\ldots,N_{\mathrm{p}}-1\}$, we can rewrite these equations as:

$$\tilde{\mathbf{x}}(k_{\mathrm{p}}+1) = \tilde{\mathbf{B}}\tilde{\mathbf{u}}(k_{\mathrm{p}}) + \tilde{\mathbf{F}}(\bar{\mathbf{x}}),$$

where

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & & & & \\ \mathbf{A}\mathbf{B} & \mathbf{B} & & & \\ \mathbf{A}^2\mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{B} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{A}^{N_{\mathrm{p}}-1}\mathbf{B} & \mathbf{A}^{N_{\mathrm{p}}-2}\mathbf{B} & \mathbf{A}^{N_{\mathrm{p}}-3}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix}$$

and

$$\tilde{\mathbf{F}}(\bar{\mathbf{x}}) = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ \vdots \\ \mathbf{A}^{N_{\mathrm{p}}} \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} \mathbf{F} \\ (\mathbf{A}+\mathbf{I})\,\mathbf{F} \\ \left(\mathbf{A}^2+\mathbf{A}+\mathbf{I}\right)\mathbf{F} \\ \vdots \\ \left(\mathbf{A}^{N_{\mathrm{p}}-1}+\ldots+\mathbf{A}+\mathbf{I}\right)\mathbf{F} \end{bmatrix}.$$

The discrete-time prediction model for $\tilde{\mathbf{y}}(k_{\mathrm{p}})$ over the prediction horizon of length N_{p} in discrete time steps is given by:

$$\tilde{\mathbf{y}}(k_{\mathrm{p}}) = \begin{bmatrix} \mathbf{C} & & & \\ & \mathbf{C} & & \\ & & \ddots & \\ & & & \mathbf{C} \end{bmatrix} \tilde{\mathbf{x}}(k_{\mathrm{p}}) + \begin{bmatrix} \mathbf{D} & & & \\ & \mathbf{D} & & \\ & & \ddots & \\ & & & \mathbf{D} \end{bmatrix} \tilde{\mathbf{u}}(k_{\mathrm{p}}) + \begin{bmatrix} \mathbf{G} \\ \mathbf{G} \\ \vdots \\ \mathbf{G} \end{bmatrix},$$

which after substitution of the prediction model for $\tilde{\mathbf{x}}(k_\mathrm{p})$ yields:

$$\tilde{\mathbf{y}}(k_\mathrm{p}) = \tilde{\mathbf{D}}\tilde{\mathbf{u}}(k_\mathrm{p}) + \tilde{\mathbf{G}}(\bar{\mathbf{x}}),$$

where

$$\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} & & & & \\ \mathbf{CB} & \mathbf{D} & & & \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & & \\ \vdots & \ddots & \ddots & \ddots & \\ \mathbf{CA}^{N_\mathrm{p}-2}\mathbf{B} & \mathbf{CA}^{N_\mathrm{p}-3}\mathbf{B} & \dots & \mathbf{CB} & \mathbf{D} \end{bmatrix}$$

and

$$\tilde{\mathbf{G}}(\bar{\mathbf{x}}) = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N_\mathrm{p}-1} \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} \mathbf{CIF} + \mathbf{G} \\ (\mathbf{CA} + \mathbf{CI})\,\mathbf{F} + \mathbf{G} \\ \vdots \\ \left(\mathbf{CA}^{N_\mathrm{p}-2} + \ldots + \mathbf{CA} + \mathbf{CI}\right)\mathbf{F} + \mathbf{G} \end{bmatrix}.$$

To take into account that the control inputs can not be adjusted at each discrete time step k_p, but only at each control cycle k_c, the equalities defining the zero-order hold on the (4.2) are added to the model. We can then denote the prediction model for $\tilde{\mathbf{y}}(k_\mathrm{p})$ by:

$$\tilde{\mathbf{y}}(k_\mathrm{p}) = M_\mathrm{lin}\left(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \tilde{\mathbf{u}}(k_\mathrm{c})\right), \tag{4.9}$$

where $M_\mathrm{lin} = \left[\tilde{\mathbf{D}}\tilde{\mathbf{u}}\left(k_\mathrm{p}\right) + \tilde{\mathbf{G}}\left(\bar{\mathbf{x}}\right)\right]$. The obtained discrete-time approximation can be employed as a prediction model in the MPC problem formulation of the medium-layer control agent. It approximates the object-oriented prediction model (4.3).

4.3 Supervisory MPC control problem formulation

We now use the prediction models as discussed in the previous section to formulate the MPC problems that a medium-layer control agent can use. Every T_c time units the control agent has to determine inputs and set-points for the coming T_c time units. These variables have to be chosen in such a way that costs over a prediction horizon of N_c control cycles, i.e., over a time span of $N_\mathrm{c}T_\mathrm{c}$ time units, are minimized. Let the control objectives of the control agent consist of determining inputs and set-points such that over the entire prediction horizon:

- the values of the output variables $\tilde{\mathbf{y}}(k_\mathrm{p})$ are maintained between given upper and lower bounds;
- the changes in the values of the set-points $\tilde{\mathbf{u}}(k_\mathrm{c})$ are minimized.

We formulate the MPC problems as a nonlinear optimization problem and a linear optimization problem.

4.3.1 Nonlinear MPC formulation

To formulate the MPC problem as a nonlinear optimization problem, we first transform the control objectives in a straightforward way into a nonlinear objective function as follows:

$$J(\tilde{\mathbf{y}}(k_\mathrm{p}),\tilde{\mathbf{u}}(k_\mathrm{c})) = \sum_{l=0}^{N_\mathrm{p}-1} \|\mathbf{Q}_\mathbf{y}\mathbf{y}_\mathrm{err}(\mathbf{y}(k_\mathrm{p}+l))\|_\infty$$
$$+\|\mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c})-\bar{\mathbf{u}})\|_1 + \sum_{l=1}^{N_\mathrm{c}-1} \|\mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c}+l)-\mathbf{u}(k_\mathrm{c}+l-1))\|_1, \qquad (4.10)$$

where $\bar{\mathbf{u}}$ are the set-points provided at the last control cycle, i.e., $\bar{\mathbf{u}} = \mathbf{u}(k_\mathrm{c}-1)$, $\mathbf{Q}_\mathbf{y}$ and $\mathbf{Q}_\mathbf{u}$ are penalty matrices, $\|\mathbf{v}\|_\infty$ and $\|\mathbf{v}\|_1$ denote the infinity and one norm of vector $\mathbf{v}$, respectively, and where $\mathbf{y}_\mathrm{err}(\mathbf{y}(k_\mathrm{p}))$ are the violations of the desired output bounds, the entries of which are computed as:

$$\mathbf{y}_{q,\mathrm{err}}(y_q(k_\mathrm{p})) = \begin{cases} y_{q,\mathrm{desired,min}} - y_q(k_\mathrm{p}) & \text{for } y_q(k_\mathrm{p}) \leq y_{q,\mathrm{desired,min}} \\ 0 & \text{for } y_{q,\mathrm{desired,min}} < y_q(k_\mathrm{p}) < y_{q,\mathrm{desired,max}} \\ y_q(k_\mathrm{p}) - y_{q,\mathrm{desired,max}} & \text{for } y_q(k_\mathrm{p}) \geq y_{q,\mathrm{desired,max}}, \end{cases} \qquad (4.11)$$

where v_q indicates entry q of vector $\mathbf{v}$, and $y_{q,\mathrm{desired,min}}$ and $y_{q,\mathrm{desired,max}}$ are the desired upper and lower bounds of y_q. The infinity norm is taken for minimization of the variables $\mathbf{y}_\mathrm{err}(\mathbf{y}(k_\mathrm{p}))$, such that the worst error is minimized. The one norm is used for the changes in the inputs $\mathbf{u}(k_\mathrm{c}+l)-\mathbf{u}(k_\mathrm{c}+l-1)$, such that the changes in each of the inputs are minimized.

The values of the output variables $\tilde{\mathbf{y}}(k_\mathrm{p})$ are related to the inputs $\tilde{\mathbf{u}}(k_\mathrm{c})$ through the prediction model, as specified in (4.3). Hence, the supervisory MPC control problem can be formulated as:

$$\min_{\tilde{\mathbf{y}}(k_\mathrm{p}),\tilde{\mathbf{u}}(k_\mathrm{c})} J(\tilde{\mathbf{y}}(k_\mathrm{p}),\tilde{\mathbf{u}}(k_\mathrm{c})) \qquad (4.12)$$

subject to

$$\tilde{\mathbf{y}}(k_\mathrm{p}) = M(\bar{\mathbf{x}},\bar{\mathbf{u}},\tilde{\mathbf{u}}(k_\mathrm{c})) \qquad (4.13)$$
$$\tilde{\mathbf{u}}_\mathrm{min} \leq \tilde{\mathbf{u}}(k_\mathrm{c}) \leq \tilde{\mathbf{u}}_\mathrm{max}, \qquad (4.14)$$

where $\tilde{\mathbf{u}}_\mathrm{min}$ and $\tilde{\mathbf{u}}_\mathrm{max}$ are vectors with bounds on the elements of $\tilde{\mathbf{u}}(k_\mathrm{c})$, and the variables with a bar are given. Instead of keeping the relation (4.13) for the prediction model as an explicit equality relation, this relation can be eliminated by substituting it into the objective function, since only the objective function depends on $\tilde{\mathbf{y}}(k_\mathrm{p})$. This substitution has computational advantages, since after the substitution the optimization problem has fewer variables and no nonlinear equality constraints. Hence, the MPC problem reduces to:

$$\min_{\tilde{\mathbf{u}}(k_\mathrm{c})} J\left(M\left(\bar{\mathbf{x}},\bar{\mathbf{u}},\tilde{\mathbf{u}}(k_\mathrm{c})\right),\tilde{\mathbf{u}}(k_\mathrm{c})\right) \qquad (4.15)$$

subject to

$$\tilde{\mathbf{u}}_\mathrm{min} \leq \tilde{\mathbf{u}}(k_\mathrm{c}) \leq \tilde{\mathbf{u}}_\mathrm{max}. \qquad (4.16)$$

Since the objective function of this problem includes the prediction model M and due to the definition of $\mathbf{y}_\mathrm{err}(\mathbf{y}(k_\mathrm{p}))$ the optimization problem is in general a nonconvex optimization problem subject to simple bound constraints.

Below we consider two approaches to solve the problem at hand. First, we propose to use the direct-search method pattern search as an appropriate solver for directly solving the nonlinear MPC problem. Pattern search has as advantage its more effective way of dealing with the problem at hand when compared to solvers for nonlinear optimization that require gradient or Hessian information. However, computational time requirements may be large. As an alternative, we consider solving the nonlinear MPC problem by using a linearized approximation of the problem. The advantage of this approach is that it may be more efficient in terms of computational requirements. However, the restricted validity of linearized models may jeopardize the quality of the resulting control actions. In Section 4.4 we experimentally compare the two approaches.

4.3.2 Direct-search methods for nonlinear optimization

In the MPC problem (4.15)–(4.16), evaluating the objective function is expensive due to the evaluation of the prediction model. In practice, computation time is limited and within the available computation time a solution that is as good as possible has to be determined. Many nonlinear optimization methods rely on gradient and Hessian information [18, 115]. However, the saturation and the use of the infinity norm in the objective function make that the objective function has many flat areas in which the gradient and Hessian are both equal to zero and thus not informative. Solvers that use this first-order or second-order information will therefore perform unnecessary numerical approximation of the gradient and the Hessian, involving numerous objective function evaluations. In addition, the MPC problem (4.15)–(4.16) typically has many local minima in which gradient-based solvers typically quickly can get stuck.

Instead of using gradient or Hessian-based solvers, we propose to use so-called direct-search optimization methods, which do not explicitly require gradient and Hessian information [32, 150]. The only property that these methods require is that the values of the objective function can be ranked [87]. This feature together with the feature that direct-search methods are suitable for non-smooth problems [32], make that these methods are suitable for solving the nonlinear problem (4.15).

Pattern search

For solving the nonlinear MPC problem (4.15)–(4.16), which is based on the object-oriented prediction model, we propose to use the direct-search method pattern search [87], for its straightforward implementation and its ability to yield good solutions, even for objective functions with many local minima, in combination with a multi-start method [96], to improve the probability of obtaining a solution close to a globally optimal solution. Several theoretical issues of pattern search have been discussed in [9, 10, 84, 138].

Pattern search works in an iterative way. Given the solution $\mathbf{s}^{(s-1)}$ at iteration $s-1$, if a new solution $\mathbf{s}^{+}$ is found for which it holds that $J(\mathbf{s}^{+}) < J(\mathbf{s}^{(s-1)})$, then the solution at iteration s becomes $\mathbf{s}^{+}$. If such a new solution is not found, then the solution at iteration s equals the solution at the previous iteration. The new solution $\mathbf{s}^{+}$ has to be selected from a finite set of candidate solutions in a mesh $\mathcal{M}^{(s)}$ that is updated at each iteration. An iteration of pattern search for an unconstrained problem is summarized in the following steps [87]:

- A mesh $\mathcal{M}^{(s)}$ around the last solution $\mathbf{s}^{(s-1)}$ is constructed, consisting of a discrete set of candidate solutions in $\mathbb{R}^{n_s}$ in which the algorithm searches for a new solution. The coarseness of the mesh is determined by the mesh size $\gamma_{\text{mesh}}^{(s-1)} \in \mathbb{R}^+$.
- The mesh $\mathcal{M}^{(s)}$ is explored in one or two phases:
  - In the *search phase* any strategy can be used to find a solution $\mathbf{s}^+ \in \mathcal{M}^{(s)}$ for which $J(\mathbf{s}^+) < J(\mathbf{s}^{(s-1)})$, as long as a finite number of points is considered. If a solution $\mathbf{s}^+$ is found, the search was successful and the next phase is not invoked.
  - In the *polling phase* a new solution $\mathbf{s}^+$ for which $J(\mathbf{s}^+) < J(\mathbf{s}^{(s-1)})$ is searched for in a subset of solutions in $\mathcal{M}^{(s)}$, consisting of those solutions that are in the direct neighborhood of the last solution $\mathbf{s}^{(s-1)}$. This neighborhood is defined through a set of vectors called a pattern and the current solution. If a solution $\mathbf{s}^+$ is found in this neighborhood then the polling phase was successful.
- If either of the phases was successful, then $\mathbf{s}^{(s)} = \mathbf{s}^+$, the coarseness of the mesh is set to $\gamma_{\text{mesh}}^{(s)} = \gamma_{\text{exp}}\gamma_{\text{mesh}}^{(s-1)}$, with expansion factor $\gamma_{\text{exp}} > 1$, and the next iteration starts. If $\mathbf{s}^+$ was not found, then $\mathbf{s}^{(s)} = \mathbf{s}^{(s-1)}$, the coarseness of the mesh is set to $\gamma_{\text{mesh}}^{(s)} = \gamma_{\text{contr}}\gamma_{\text{mesh}}^{(s-1)}$, with contraction factor $\gamma_{\text{contr}} \in (0,1)$, and the next iteration starts.

The iterations continue until a stopping condition is satisfied, e.g., the mesh size is less than a given tolerance, the total number of objective function evaluations reaches a given maximum, or the distance between the point found at one successful poll and the point at the next successful poll is less than a given tolerance.

Approaches of pattern search for solving constraint optimization problems have been addressed in the literature, e.g., for optimization problems with bound constraints [86], linear constraints [84], and nonlinear constraints [85].

Multi-start pattern search

The combination of pattern search with multi-start for solving the control problem at control cycle k_c consists of solving the control problem from N_{init} different initial solutions, with N_{init} a positive integer. In general, the larger N_{init}, the larger the chance of obtaining a solution close to a globally optimal solution. However, in practice computation time is limited, since control set-points have to be provided at each control cycle. Therefore, our multi-start implementation involves starting from different initial solutions as long as time is available. The first initial solution is based on the (perhaps shifted) solution of control cycle $k_c - 1$, since the solution of control cycle $k_c - 1$ typically gives a good guess of the solution at control cycle k_c. The solution with the minimal objective function value after optimization with pattern search when the maximum computation time has elapsed is used as the final solution at control cycle k_c. See [96] for an overview of further characteristics of multi-start methods.

Although multi-start methods generally increase the time required to solve an optimization problem significantly, multi-start methods can typically be executed in a highly parallel fashion. In particular when a straightforward multi-start method is chosen that relies on randomly generated initial solutions, then each optimization problem involved in the multi-start method can be solved on an independent processor. For N_{init} initial solutions executed on p processors the overall execution time is then expected to improve with approximately a factor p compared to when a single processor is used.

In Section 4.4 we experimentally compare the performance of multi-start pattern search with a multi-start gradient-based optimization.

4.3.3 Linear MPC formulation

The approach proposed above for solving the nonlinear optimization problem based on multi-start pattern search may still require a significant amount of computation time. Instead of solving the nonlinear optimization problem directly, we here discuss solving an approximation of the nonlinear optimization problem by linearization. This approach has the potential to require a significantly smaller amount of computation time, although possibly at the price of reduced performance.

To obtain a linear approximation of the MPC formulation of (4.12)–(4.14), the linearized prediction model (4.9) can be used instead of the object-oriented prediction model (4.3), and a transformation of the nonlinear objective function (4.10) and the expression for $\mathbf{y}_{\mathrm{err}}(\mathbf{y}(k_{\mathrm{p}}))$ in (4.11) can be made into linear objective terms and inequality constraints. First, note that the following optimization problem:

$$\min_{\mathbf{y}_{\mathrm{err}}} \|\mathbf{y}_{\mathrm{err}}(\bar{\mathbf{y}}(k_{\mathrm{p}}))\|_{\infty}$$

where $\mathbf{y}_{\mathrm{err}}$ as defined in (4.11), for any fixed $\bar{\mathbf{y}}(k_{\mathrm{p}})$, is equivalent to the optimization problem:

$$\begin{aligned} &\min_{\mathbf{y}_{\mathrm{err}}} \|\mathbf{y}_{\mathrm{err}}\|_{\infty} \\ \text{subject to} \quad & \\ &\bar{\mathbf{y}}(k_{\mathrm{p}}) \geq \mathbf{y}_{\mathrm{desired,min}} - \mathbf{y}_{\mathrm{err}} \\ &\bar{\mathbf{y}}(k_{\mathrm{p}}) \leq \mathbf{y}_{\mathrm{desired,max}} + \mathbf{y}_{\mathrm{err}} \\ &\mathbf{y}_{\mathrm{err}} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{0}$ is a zero vector of length $n_{\mathbf{y}_{\mathrm{err}}}$. Note also that the infinity-norm based optimization problem:

$$\min_{\mathbf{v}} \|\mathbf{Q}\mathbf{v}\|_{\infty},$$

where $\mathbf{v} \in \mathbb{R}^{n_{\mathbf{v}}}$, and $\mathbf{Q} \in \mathbb{R}^{n_{\mathbf{v}} \times n_{\mathbf{v}}}$, is equivalent to the linear programming problem:

$$\begin{aligned} &\min_{\mathbf{v}, z_{\infty}} z_{\infty} \\ \text{subject to} \quad & -\mathbf{1} z_{\infty} \leq \mathbf{Q}\mathbf{v} \\ &\mathbf{Q}\mathbf{v} \leq \mathbf{1} z_{\infty}, \end{aligned}$$

where $z_{\infty} \in \mathbb{R}$, and $\mathbf{1}$ is a one vector of length $n_{\mathbf{v}}$. In addition, note that the one-norm based optimization problem:

$$\min_{\mathbf{v}} \|\mathbf{Q}\mathbf{v}\|_{1},$$

where $\mathbf{v} \in \mathbb{R}^{n_\mathbf{v}}$, $\mathbf{Q} \in \mathbb{R}^{n_\mathbf{v} \times n_\mathbf{v}}$, is equivalent to the linear programming problem:

$$\begin{aligned} &\min_{\mathbf{v},\mathbf{z}_1} \mathbf{1}^\mathrm{T}\mathbf{z}_1 \\ &\text{subject to } -\mathbf{z}_1 \leq \mathbf{Q}\mathbf{v} \\ &\qquad\qquad \mathbf{Q}\mathbf{v} \leq \mathbf{z}_1, \end{aligned}$$

where $\mathbf{z}_1 \in \mathbb{R}^{n_{\mathbf{z}_1}}$ and $\mathbf{1}$ is a one vector of length $n_{\mathbf{z}_1}$. Using these equivalences the nonlinear optimization problem as defined in (4.12)–(4.14) is transformed into:

$$\min_{\tilde{\mathbf{y}}(k_\mathrm{p}),\tilde{\mathbf{u}}(k_\mathrm{c}),\tilde{\mathbf{y}}_\mathrm{err}(k_\mathrm{p}),\tilde{\mathbf{z}}_\infty(k_\mathrm{p}),\tilde{\mathbf{z}}_1(k_\mathrm{c})} \sum_{l=0}^{N_\mathrm{p}-1} z_\infty(k_\mathrm{p}+l) + \sum_{l=0}^{N_\mathrm{p}-1} \mathbf{1}^\mathrm{T}\mathbf{z}_1(k_\mathrm{p}+l) \tag{4.17}$$

subject to

$$\begin{aligned}
&\mathbf{y}(k_\mathrm{p}+l) \geq \mathbf{y}_\mathrm{desired,min} - \mathbf{y}_\mathrm{err}(k_\mathrm{p}+l) \\
&\mathbf{y}(k_\mathrm{p}+l) \leq \mathbf{y}_\mathrm{desired,max} + \mathbf{y}_\mathrm{err}(k_\mathrm{p}+l) \\
&\mathbf{y}_\mathrm{err}(k_\mathrm{p}+l) \geq \mathbf{0} \\
&-z_\infty(k_\mathrm{p}+l) \leq \mathbf{Q}_\mathbf{y}\mathbf{y}_\mathrm{err}(k_\mathrm{p}+l) \\
&\mathbf{Q}_\mathbf{y}\mathbf{y}_\mathrm{err}(k_\mathrm{p}+1) \leq z_\infty(k_\mathrm{p}+l) \\
&\qquad\qquad \text{for } l = 0,\ldots,N_\mathrm{p}-1 \\
&-\mathbf{z}_1(k_\mathrm{c}) \leq \mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c}) - \bar{\mathbf{u}}) \\
&\mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c}) - \bar{\mathbf{u}}) \leq \mathbf{z}_1(k_\mathrm{c}) \\
&-\mathbf{z}_1(k_\mathrm{c}+l) \leq \mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c}+l) - \mathbf{u}(k_\mathrm{c}+l-1)) \\
&\mathbf{Q}_\mathbf{u}(\mathbf{u}(k_\mathrm{c}+l) - \mathbf{u}(k_\mathrm{c}+l-1)) \leq \mathbf{z}_1(k_\mathrm{c}+l) \\
&\qquad\qquad \text{for } l = 1,\ldots,N_\mathrm{c}-1 \\
&\tilde{\mathbf{y}}(k_\mathrm{p}) = M_\mathrm{lin}(\bar{\mathbf{x}},\bar{\mathbf{u}},\tilde{\mathbf{u}}(k_\mathrm{c})) \\
&\tilde{\mathbf{u}}_\mathrm{min} \leq \tilde{\mathbf{u}}(k_\mathrm{c}) \leq \tilde{\mathbf{u}}_\mathrm{max}.
\end{aligned} \tag{4.18}$$

Since we have a linear objective function with linear equality and inequality constraints, and since all variables are continuous, this MPC optimization problem is a linear programming problem, for which there exist good commercial and free solvers [103].

4.4 Application: Voltage control in a 9-bus power network

A major source of power outages is voltage instability [142]. Voltage instability in general stems from the attempt of load dynamics to restore power consumption beyond the capability of the combined transmission and generation system after a fault. The control problem we are dealing with in this section is emergency voltage control, i.e., control to prevent a particular type of voltage instability. After a fault, e.g., a partial or total outage of a line, the generation and transmission network may not have sufficient capacity to provide the loads with power. A lower layer of decentralized control agents will try to restore the behavior of the system to an acceptable level. However, due to the reduced transmission capacity of the network the requested load demand together with the given system configuration place the network under an excessive amount of strain and voltages may start to drop. Corrective

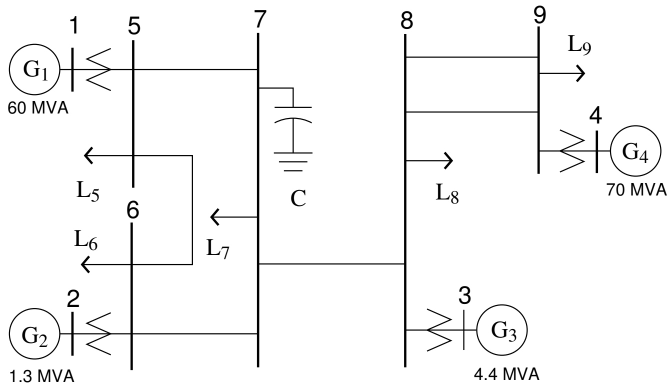

Figure 4.3: Network topology of the 9-bus dynamic network. The generators are shown with their nominal apparent power ratings.

actions have to be taken to coordinate the decentralized control agents in the lower control layer such that negative effects of this voltage instability are minimized and such that the induced transients that drive the system to collapse or cause unwanted and hazardous sustained oscillations are avoided.

Traditionally, offline static stability studies are carried out in order to avert the occurrence of voltage instability. The approach we propose in this section is an application of *online* control that takes into account both the inherent temporal dynamics and that determines the most appropriate control sequence required to reach an acceptable and secure operating point. We therefore propose the use of the MPC schemes discussed in Section 4.3 by a medium-layer control agent to determine the set-points for lower-layer control agents in such a way that negative effects due to voltage instability after faults are minimized.

In the following we describe the power network and control setup, formulate MPC problems based on an object-oriented and a linearized prediction model, and experimentally assess the performance of the medium-layer control agent using these MPC formulations.

4.4.1 The 9-bus dynamic benchmark network

We perform simulation studies on a 9-bus power network. Figure 4.3 shows the topology of the physical network. This system is an adjusted version of the 9-bus Anderson-Farmer network [46], taken from the Dynamical Systems Benchmark Library [63]. The following list contains more details on the dynamics of the network:

- Synchronous machines: The network consists of 4 synchronous machines G_1, G_2, G_3 and G_4. The synchronous machines are connected to the network via lossless step-up transformers featuring a fixed turns ratio. Synchronous machines G_2 and G_3 represent single physical unit, whereas synchronous machines G_1 and G_4 denote aggregate machines comprising several physical units. The mechanical power and the level of the excitation field can be adjusted for each machine.

- Loads: There are 5 loads, L_5, L_6, L_7, L_8, L_9. Part of the loads can be disabled by using load shedding.

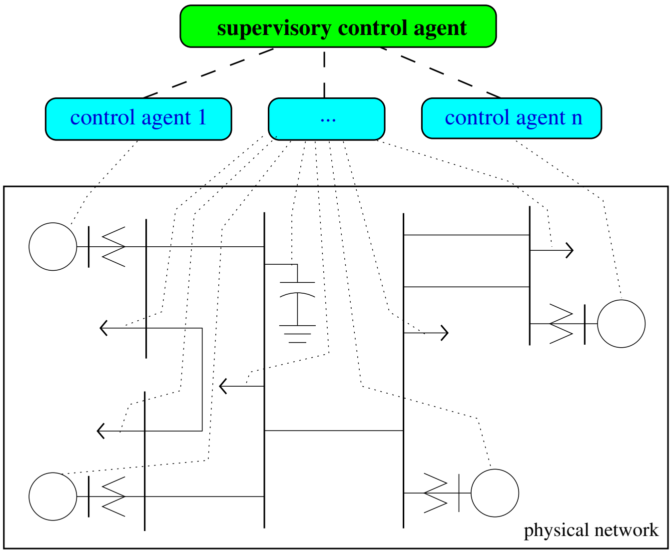

Figure 4.4: Illustration of the control of the power network.

- Capacitor bank: A capacitor bank C at bus 7 provides additional reactive power to the network to locally stabilize bus voltage magnitudes. Capacitors can be connected or disconnected from the network in discrete quantities.

- Transmission lines: The transmission lines between the buses and components transfer the power from one location to another.

Note that this power network contains very fast dynamics, due to the transmission lines, fast dynamics, due to the generators, and slow dynamics, due to the loads. Control of the physical network is done through two-layered control, consisting of a lower, primary, control layer, and a medium, secondary, control layer. Figure 4.4 illustrates this control structure.

Lower control layer

The lower control layer in the network regulates power flows and voltage levels at the bus terminals of generators. The lower control layer consists of the following elements:

- Turbine governors: All generators feature a turbine governor controlling the mechanical power acting on the shaft of the machine in order to satisfy the active power demand of the network and maintain a desired frequency. The turbine governors act on a time scale of tens of seconds. The turbine governors accept set-points for the mechanical power and frequency.

- Automatic voltage regulators: All generators feature an automatic voltage regulator (AVR) maintaining the level of the excitation field in the rotor windings at the value required to keep the bus voltage magnitude close to the desired set-point. The maximum current in the excitation system is limited. Once a machine has reached one of its limits it cannot produce additional reactive power and can therefore no longer participate in sustaining the voltage magnitudes in the network [82]. The AVR voltage references of the generators can be set in the range 0.9–1.1 p.u. The AVRs act on a time scale of seconds. The AVRs accept set-points for the voltage magnitudes of the generators' terminal buses.

- Power system stabilizers: Generators G_2 and G_3 feature a power system stabilizer (PSS) eliminating the presence of unwanted rotor oscillations by measuring the rotational speed and adding a corrective factor to the voltage magnitude reference for the AVRs. The corrective factor saturates at a lower and upper bound. Generators G_1 and G_4 feature no power system stabilizer since these generators represent multiple physical generators. The PSSs act on a time scale of tenths of seconds. The PSSs accept set-points for the frequency.

Control handles available to a medium control layer

Given the description of the network and the lower control layer, the control handles available to a higher control layer in the form of set-point and reference settings are summarized as follows:

- the voltage references for the AVRs;
- the mechanical power set-points for the turbine governors;
- the reference frequency for the turbine governors and the PSSs;
- the amount of load to shed;
- the amount of capacitor banks to connect to the grid.

Depending on the particular control problem a higher-layer control agent will adjust the values of these control handles. In particular for the voltage control problem at hand the amount of load shed and the set points of the AVRs will be taken as the available control handles.

4.4.2 Object-oriented model of the network

To construct an object-oriented model of the network, we first define several classes to describe the components in the power network. Using the definition of the classes we formalize the structure of the network. To each class we assign a set of variables and a set of equations, typically consisting of a system of DAEs. The equations of a class constrain the values of variables over time, and therefore add to the behavior of the object-oriented model. The equations of a particular class first of all typically constrain variables of that particular class. In addition, the equations of a particular class can also constrain the variables in classes from which that particular class is a subclass. After having defined the classes and the associated constraints, the classes can be instantiated into objects to form the object-oriented representation of the 9-bus network.

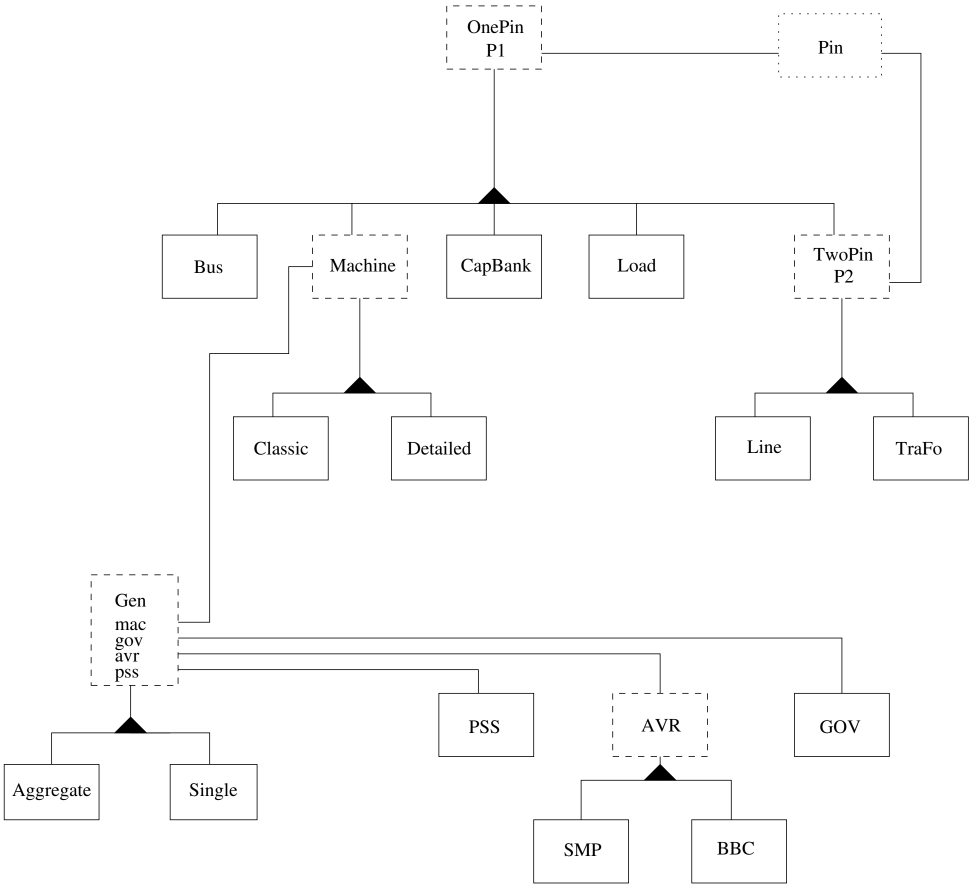

Figure 4.5: Class diagram for a power network.

Class definitions

Figure 4.5 shows the class diagram of the components that we consider. Below we motivate the definition of the classes shown in the figure.

The components in the physical network all have in common that they are connected to other components. To model this, the connector class [39] *Pin* is defined. The *Pin* connector class defines variables for the voltage magnitude and angle, and the current magnitude and angle. No additional constraints on the values of these variables are defined, however when two components are connected to each other through the *Pin* connector class, four constraints are defined that force the voltage magnitudes and angles of both components to be equal, and that force the sum of the current magnitudes and angles of both components to be zero.

Components like buses, machines, loads, and capacitor banks are connected to the network at one point. We therefore define the basic class *OnePin*. This class has a single variable $P1$, which refers to an object of class *Pin*, and has no further additional constraints. Components like transmission lines and transformers are connected to the network at two points. Therefore, we define the class *TwoPin* as an extension of the *OnePin* class. This class has a single variable $P2$, which refers to an object of class *Pin*. Note that by extending the *OnePin* class, the *TwoPin* class inherits access to the variables of the *OnePin* class, hence it inherits access to variable $P1$ and the *Pin* class variables associated with this variable.

Since no additional constraints are defined in the *OnePin* and *TwoPin* classes, the values for the variables involved in either of the two classes, consisting of the four variables of the $P1$ variable, cannot be determined. These classes are therefore *partial* classes [39]. Subclasses of partial classes have to be defined containing constraints to give these variables values.

Subclasses of the class *OnePin* represent those components in the network that are connected at a single point, i.e., buses, machines, loads, and capacitor banks. Therefore, classes *Bus*, *Machine*, *Load*, and *CapBank* are defined as subclasses of class *OnePin*. There are different types of machines and we therefore define as subclasses of the class *Machine* the classes *Classic* and *Detailed*. The following lists the most important characteristics of the dynamics associated with these classes, and the variables that the classes expect as control inputs or provide as outputs:

- The *Bus* class involves two constraints that force the current magnitude and angle of the pin of the bus to be zero.
- The *Classic* class is equipped with classical 2nd-order mechanical dynamics [63, 82]. The dynamics of this machine depend on the level of field voltage $u_E(t)$ and mechanical power $u_{\text{Pm}}(t)$. The value of the voltage magnitude $y_V(t)$ of the bus of the machine and the frequency deviation $y_{\Delta\omega}(t)$ are made available to other classes.
- The *Detailed* class is equipped with a detailed 6th-order model [63, 82] including the mechanical equations and the electrical transient and sub-transient dynamics of the machine, since it represents a single physical unit. The variables that the dynamics depend on are the same as for the *Classic* class. Also the values that are available to other classes are the same.
- If the original benchmark definition would be used, the *Load* class would be equipped with a static voltage dependent and constant impedance load model [76]. However, to model the loads in more detail and to obtain slow load dynamics, a 2nd-order ZIP model [59] is assigned to the *Load* class. Among others, two constraints are included describing the relation between the current angle and magnitude and the voltage angle and magnitude under different amounts of active and reactive power consumption. The class accepts as input the amount of load to shed $u_{\text{shed}}(t)$.
- The *CapBank* class is equipped with two static constraints relating the number of capacitors $u_{\text{cap}}(t)$ connected to the power network to the current magnitude and angle and the voltage magnitude and angle of its pin [63]. The class accepts as input the number of capacitors $u_{\text{cap}}(t)$ to connect to the network. This input is a variable that can take on only discrete values.

Subclasses of the class *TwoBus* represent those components in the network that are connected to two buses, i.e., transmission lines and transformers. Therefore, classes *Line* and *TraFo* are defined as subclasses of class *TwoPin*. The most important characteristics of the dynamics associated with these classes and the variables that the classes expect as control inputs or provide as outputs are:

- The *Line* class is equipped with the static equations of the π model for transmission lines [63, 82]. Four constraints relate the eight variables of the two pins.
- The *TraFo* class is equipped with the static equations of the π model for transmission lines [63, 82], but with the resistance and susceptance set to zero. Four constraints relate the eight variables of the two pins.

Each of the classes defined so far contains variables related to the particular component being modeled, i.e., input, state, algebraic, and output variables, and equations describing the behavior of the components. Moreover, each of these classes defines constraints involving the variables of the *OnePin* or *TwoPin* class.

There are also several components that do not directly connect to the power network, and that therefore are not defined as a subclass of the *OnePin* or *TwoPin* class. These components consist, e.g., of the components in the lower control layer, which determine the inputs to components directly connected to the power network. Examples of these are the AVRs, turbine governors, and possibly PSSs. Corresponding classes *AVR*, *GOV*, and *PSS* are therefore defined. For the class *AVR* subclasses *SMP* and *BBC* are defined, depicting two different types of AVRs. The most important characteristics of the dynamics of these classes, and the variables that the classes expect as control inputs or provide as outputs, are the following:

- The *SMP* class is equipped with the equations of a 3rd-order AVR [63, 82]. The *SMP* class accepts as inputs a bus voltage magnitude $u_{V,\mathrm{mac}}(t)$ of the bus of which the AVR should regulate the voltage magnitude, and a voltage magnitude set-point $u_{V,\mathrm{PSS}}(t)$. In addition, the *SMP* class accepts as input voltage magnitude set-point $u_{V,\mathrm{ref}}(t)$. The *SMP* class provides the excitation field voltage $y_E(t)$ as output. The excitation field voltage $y_E(t)$ saturates at a lower limit $y_{E,\mathrm{min}}$ and an upper limit $y_{E,\mathrm{max}}$.
- The *BBC* class is equipped with 2nd-order dynamics [63, 82]. This class has the same inputs and outputs as the *SMP* class. Also this AVR class considers saturation of the excitation field voltage $y_E(t)$.
- The *GOV* class is equipped with 3rd-order dynamics [63, 82]. The dynamics have as input a frequency deviation $u_\omega(t)$ of a machine. The *GOV* class accepts $u_{\mathrm{Torder}}(t)$ as set-point for the mechanical power. The class provides mechanical power $y_{\mathrm{Pm}}(t)$ as output.
- The *PSS* class is equipped with 3rd-order dynamics [63, 82]. It uses as input a frequency deviation $u_\omega(t)$ to determine a voltage magnitude $y_{V,\mathrm{PSS}}(t)$. The voltage magnitude $y_{V,\mathrm{PSS}}(t)$ saturates at upper bound $y_{V,\mathrm{PSS},max}$ and lower bound $\eta_{V,\mathrm{PSS,min}}$.

Having defined the classes for these individual components, it is convenient to define some classes by composition. E.g., the class *Gen* is defined as the composition of a machine with a specific lower control configuration. As subclasses we define the classes *Aggregate* and *Single*. The classes *Aggregate* and *Single* include references to specific *AVR*, *GOV*, and *PSS* classes.

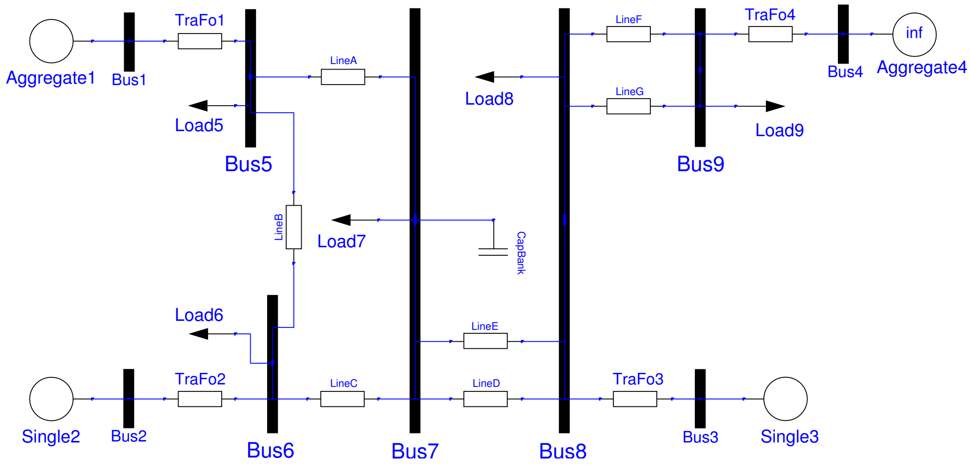

Figure 4.6: The 9-bus power network as object diagram.

Remark 4.4 In the example of the division of the power network into classes and subclasses given here only the components that will be used later have been defined. It is straightforward, however, to include many more subclasses, e.g., for describing different loads, transformers, and additional generators. In [105] several examples of additional components that can be added can be found. The classification of components into classes facilitates easy experimenting with models with different levels of detail. □

Object diagram

Given the classes and the associated dynamics, we can now instantiate the classes into objects to form an object diagram for the power network under study. Generators G_1 and G_4 are of class *Aggregate*. Generators G_2 and G_3 are of class *Single*. The loads are of class *Load*. The capacitor bank is of class *CapBank*. The buses are of class *Bus*. The transmission lines are of class *Line*, and the transformers are of class *TraFo*. Figure 4.6 shows the layout of the resulting object diagram as created in Dymola. The Dymola model can be obtained from the author on request.

4.4.3 Control problem formulation for the higher control layer

To illustrate the control problem, we consider a typical scenario with no medium-layer MPC control agent installed, in which we use the model constructed in the previous section as model of the physical network. In the scenario that we consider, the network is initially in steady state. Then, at $t_{\text{fault}} = 26.5$ s a fault of 600% impedance increase in the transformer between bus 1 and 5 occurs. Figure 4.7 illustrates the evolution of the voltage magnitudes of three representative buses. The fault occurring in the transformer between bus 1 and 5 changes the transmission capacity of the network. Due to the changed transmission capacity of the network and due to the dynamics of the loads, the voltage magnitudes start to start oscillating, despite the actions of the lower control layer. If the set-points to the control agents

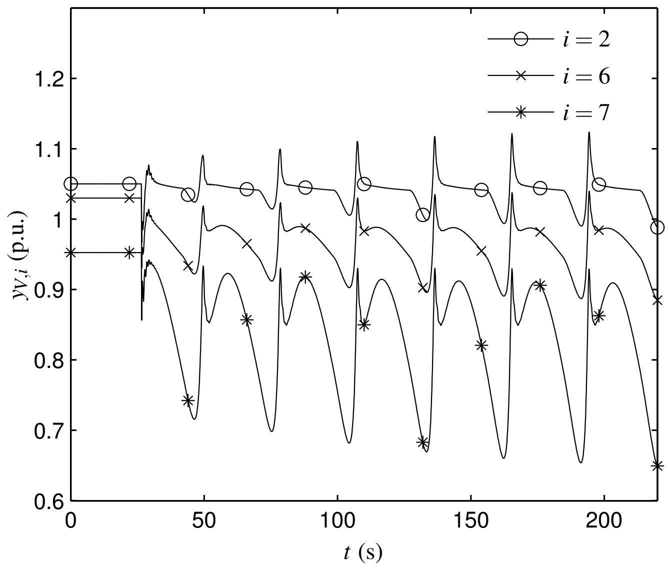

Figure 4.7: Voltage magnitude profiles $y_{V,i}$ of three representative buses, ($i = 2$, $i = 6$, and $i = 7$), for a typical scenario in which no medium control layer is present. After a fault of 600% impedance increase occurs in the transformer between buses 1 and 5 at $t_{fault} = 26.5$ s, the voltage magnitudes $y_{V,i}$ start oscillating, ultimately resulting in a network collapse.

of the lower control layer are not changed, perhaps in combination with other measures, the network ultimately collapses.

To prevent such a collapse from occurring, a higher-layer control agent should be installed with the task to [49]:

1. Maintain the voltage magnitudes between 0.9 and 1.1 p.u., i.e., sufficiently close to nominal values to ensure a safe operation of the system by keeping the voltage magnitudes sufficiently distant from low voltages.
2. Effectively achieve a steady-state point of operation, while minimizing changing of the control inputs so that a constant and appropriate set of input values is ultimately applied to the power network and the lower control layer.

For the second objective, in particular the option of shedding load is to be avoided unless absolutely necessary in order to fulfill the primary objective, as load shedding is the most disruptive countermeasure available. Since typical slow voltage collapses without a medium-layer control agent installed emerge over time spans of several tens of seconds up to several minutes [142], a control cycle time of 20 s is acceptable. It should be noted that the speed at which a voltage collapse unfolds depends on the magnitude of the fault occurring. A collapse will take place sooner with a larger fault than with a smaller fault. So, depending on the range of faults that should be adequately dealt with, the control cycle time will have to be decreased or increased.

Below we formulate the nonlinear and linear higher-layer MPC problem of the 9-bus power network, and we assess the performance of the resulting closed-loop control structure in experiments.

Nonlinear MPC problem formulation

The control problem of the supervisory control agent using the object-oriented prediction model is based on the formulation specified in Section 4.3.1. For $l = 0, \ldots, N_\mathrm{c} - 1$, the inputs $\tilde{\mathbf{u}}(k_\mathrm{c} + l)$ correspond to the AVR set-points $u_{\mathrm{AVR},i}(k_\mathrm{c} + l)$, for $i = \{1, \ldots, 4\}$, and the amounts of load to shed $u_{\mathrm{shed},i}(k_\mathrm{c} + l)$, for $i = \{5, \ldots, 9\}$. For $l = 0, \ldots, N_\mathrm{p} - 1$, the outputs $\tilde{\mathbf{y}}(k_\mathrm{p} + l)$ correspond to the voltage magnitudes $y_{V,i}(k_\mathrm{p})$, for $i = \{1, \ldots, 9\}$, at the 9 buses.

One control cycle takes 20 s, hence T_c is 20 s. Although in principle a the prediction horizon should include all important dynamics, for computational reasons a prediction horizon with a length of only 2 control cycles is taken. The continuous voltage signal is sampled every 0.5 s, hence T_p is 0.5 s, and the length of the prediction horizon N_p is therefore 40 prediction steps.

The MPC control problem is formulated as in (4.15)–(4.16), where $y_{q,\mathrm{desired,min}}$ is 0.9 p.u. and $y_{q,\mathrm{desired,max}}$ is 1.1 p.u. for each element of $\tilde{\mathbf{y}}(k_\mathrm{p} + l)$. The elements of $\mathbf{u}_\mathrm{min}$ and $\mathbf{u}_\mathrm{max}$ corresponding to AVR settings $u_{\mathrm{AVR},i}(k_\mathrm{c} + l)$ are set to 0.9 and 1.1 p.u., respectively. The elements of $\mathbf{u}_\mathrm{min}$ and $\mathbf{u}_\mathrm{max}$ corresponding to load settings $u_{\mathrm{shed},i}(k_\mathrm{c} + l)$ are set to 0 and 1, respectively, corresponding to full load shedding or no load shedding, respectively.

The cost matrix $\mathbf{Q}_\mathbf{y}$ contains on its diagonal elements $\frac{1}{N_\mathrm{p}/N_\mathrm{c}} 200$ and $\mathbf{Q}_\mathbf{u}$ contains the value 1 on the diagonal elements corresponding to AVR settings $u_{\mathrm{AVR},i}(k_\mathrm{c} + l)$, and the value 20 on the diagonal elements corresponding to load shedding settings $u_{\mathrm{shed},i}(k_\mathrm{c} + l)$. This way of penalizing the voltage bound violations, the AVR settings, and the load shedding settings ensures that the main objective of the control agent is to satisfy the voltage objectives, and that load shedding should only be chosen as a last resort.

Linear MPC problem formulation

The control problem of the supervisory control agent using the linearized prediction model is based on the formulation given in Section 4.3.3. The MPC control is formulated using (4.17)–(4.18). The length of the prediction horizon in prediction steps N_p is 40, and the length of the prediction horizon in control cycles N_c is 2. The inputs $\tilde{\mathbf{u}}(k_\mathrm{c})$ correspond to the set-points for the AVRs $u_{\mathrm{AVR},i}(k_\mathrm{c} + l)$ and the amounts of load to shed $u_{\mathrm{shed},i}(k_\mathrm{c} + l)$ over the prediction horizon. The outputs $\tilde{\mathbf{y}}(k_\mathrm{p})$ correspond to the voltage magnitudes $y_{V,i}(k_\mathrm{p} + l)$ at the 9 buses.

Similar as for the nonlinear MPC formulation, the cost matrices $\mathbf{Q}_\mathbf{y}$ and $\mathbf{Q}_\mathbf{u}$ are defined such that a weight of $\frac{1}{N_\mathrm{p}/N_\mathrm{c}} 200$ is placed on the violation of each soft constraint. The inputs are weighted with the penalty coefficients 1 and 20 for the AVR settings $u_{\mathrm{AVR},i}(k_\mathrm{c})$ and the load shedding settings $u_{\mathrm{shed},i}(k_\mathrm{c})$, respectively.

The linearized prediction model is obtained at each control cycle k_c by linearizing the object-oriented prediction model around the current state $\mathbf{x}(k_\mathrm{c})$ and the inputs applied at the preceding time instant $\mathbf{u}(k_\mathrm{c} - 1)$. The sampling interval $T_\mathrm{p} = 0.5$ s.

In the following we first focus on the performance of the control agent when it uses the nonlinear MPC formulation. We illustrate the difference between pattern search and gradient-based optimization methods, and illustrate how the proposed approach chooses adequate set points that prevent the network from collapsing. Then, we also consider the performance of the control agent, when it uses the linear MPC formulation. We illustrate how the two strategies compare.

4.4.4 Control using the nonlinear MPC formulation

Direct search versus gradient-based optimization

We compare pattern search as part of Matlab's Direct Search and Genetic Algorithms toolbox in Matlab v7.3 [97] with the derivative-based solver SNOPT v5.8 [50], as implemented in Tomlab v5.7 [65], and accessed from Matlab. SNOPT uses a sparse sequential quadratic programming method, using limited-memory quasi-Newton approximations to the Hessian of the Lagrange. In principle it requires gradient information, but this information can be approximated numerically if it is not available.

To compare the performance of the solvers, we perform 50 experiments in which a single fault occurs at varying locations in the power network (i.e., at the 4 transformers and the lines), with varying magnitudes (i.e., an impedance increase of 100% up to 800%), and at varying times (i.e., the fault time varies between second 20 and 28). The control problems of the first control cycle after a fault has been applied are solved by both pattern search and SNOPT, allowing a decision making time of 300 s[1].

In Figure 4.8 we see that SNOPT considers far more initial solutions within the given time span. The time that SNOPT requires to obtain a locally optimal solution is much lower than the time required by pattern search. This is explained by the fact that SNOPT uses much fewer prediction model evaluations per optimization, since it does not explore the search space as much as pattern search does.

Figure 4.9 shows, as decision time progresses, the average over all experiments of the best objective value of pattern search so far divided by the best objective value of SNOPT so far. This fraction is 1, if the best objective values of pattern search and SNOPT are on average the same. It is larger than 1, if SNOPT on average has a better solution, and smaller than 1 if pattern search has a better solution on average. The figure considers only points for which the fraction can be computed, i.e., both pattern search and SNOPT have finished at least one optimization problem. We observe that pattern search on average has a best objective value so far that is about a factor 5 smaller than the best objective value so far of SNOPT.

The comparison shows that pattern search, although it does not require gradient or Hessian information and is straightforward to implement, generally provides solutions that outperform the solutions provided by SNOPT.

[1] This relatively long decision making time is taken to illustrate how the performance of both solvers varies over time. In practice, multiple processors can be employed to parallelize the multi-start approach and to obtain acceptable solutions in a more realistic time frame. In addition all code can be optimized for speed and implemented in object code (currently only the SNOPT code is in object code). This is in particular important for the objective function evaluations, since these consume the most significant part of the computation time.

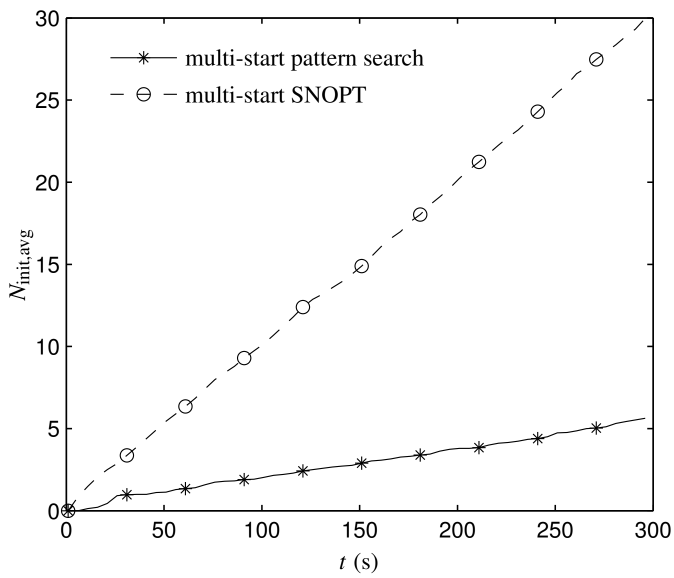

Figure 4.8: Average $N_{init,avg}$ of accumulated number of initial solutions considered by the solvers as decision time t progresses.

Control for a single scenario

To illustrate the performance of the medium-layer control agent using the nonlinear MPC formulation, we now discuss a single scenario. We reconsider the fault of 600% impedance increase at $t_{\text{fault}} = 26.5$ s in the transformer in the line from bus 1 to 5. Figure 4.7 shows the evolution of three representative buses when no medium-layer control agent is installed. We now consider using a supevisory control agent that uses the nonlinear MPC formulation. The supervisory control agent operates at $T_{\text{c}} = 20$ s using multi-start pattern search as discussed before to solve the nonlinear MPC problem. The supervisory control agent uses a prediction horizon with a length of of 40 s, and samples the voltage magnitudes from its prediction model every 0.5 s.

Figure 4.10 shows the resulting voltage magnitude profiles and Figure 4.11 shows the chosen set-points. It should be noted that the load shedding set-point is scaled to take on values between 0 and 50, corresponding to 100% load shedding and no load shedding, respectively, and that the AVR set-points for the automatic voltage regulators are scaled to take on values between 0 and 20, corresponding to 0.9 p.u. and 1.1 p.u., respectively.

After the fault has appeared, the control agent is able to stabilize the voltage magnitude between 0.9 and 1.1 p.u. with a low number of set-point changes and thus achieves its objectives. The control agent obtains a total performance[2] of 98.7, and it takes the control agent in total 157.4 s to determine its control actions.

[2]The total performance is obtained by evaluating the nonlinear objective function over the full day with $T_{\text{p}} = 0.1$ s.

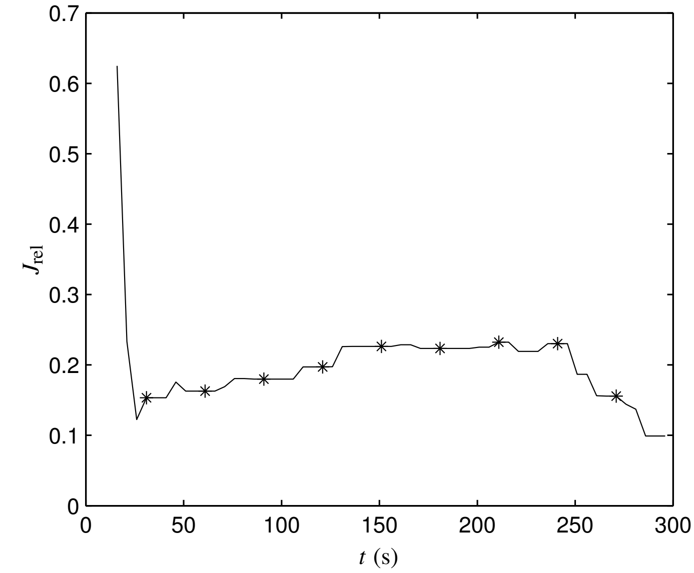

Figure 4.9: Average relative performance J_{rel} of pattern search compared to SNOPT over all experiments. The average relative performance J_{rel} at a particular time t is computated as follows: best objective value of pattern search so far divided by best objective value of SNOPT so far, averaged over all experiments.

4.4.5 Control using the linear MPC formulation

As alternative to solving the optimization problem using pattern search, we now use the linear MPC problem formulation. To solve the linear programming problems at each control cycle, we use the ILOG CPLEX v10 linear programming problem solver [71], which we access through the Tomlab v5.7 [66] interface in Matlab v7.3 [98].

We consider the following scenario. The network is in steady state, when at $t_{\text{fault}} = 26.5$ a fault appears, which increases the impedance in the transformer between buses 1 and 5 with 600%. The medium-layer control agent again operates at $T_{\text{c}} = 20$ s, and uses the linear MPC formulation with a prediction horizon with a length of 40 s, while sampling the voltage magnitudes every 0.5 s.

Figures 4.12 and 4.12 show the evolution of the voltages over the simulation and the set-points chosen by the control agent, respectively. We observe that the control agent can determine actions that stabilize the voltages at acceptable levels, despite the linearized approximation that the control agents uses for the prediction model. The control agent using the linear MPC formulation obtains a total performance of 142.4, and it takes the control agent in total 26.3 s to determine its control actions. Hence, although the control agent does not obtain an improved performance when compared to the control agent using the nonlinear MPC formulation, it does achieve stabilizing the voltage magnitudes using significantly fewer computation time.

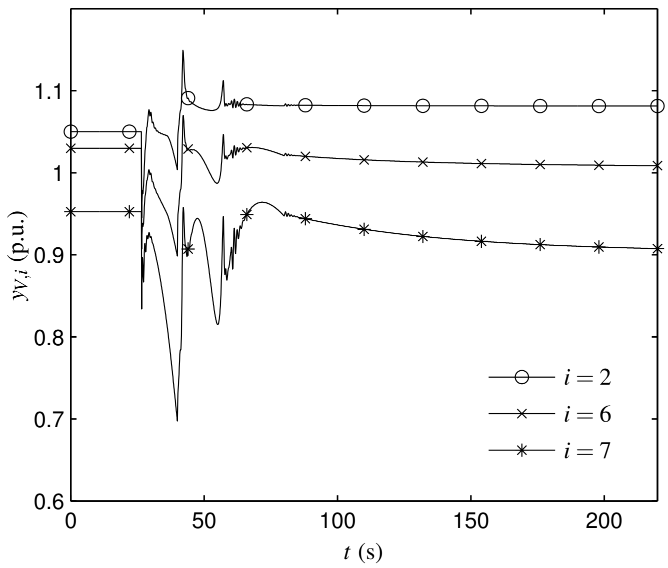

Figure 4.10: Voltage magnitude profiles for simulation including a medium-layer nonlinear MPC control agent.

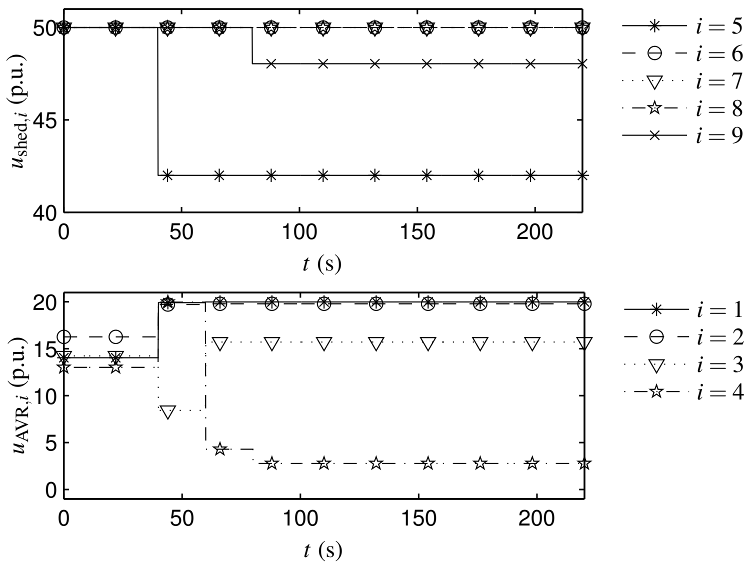

Figure 4.11: Set-points provided by the supervisory control agent for simulation including the nonlinear medium-layer MPC control agent. Load shedding values are scaled to lie within 0 and 50. AVR set-points are scaled to lie within 0 and 20.

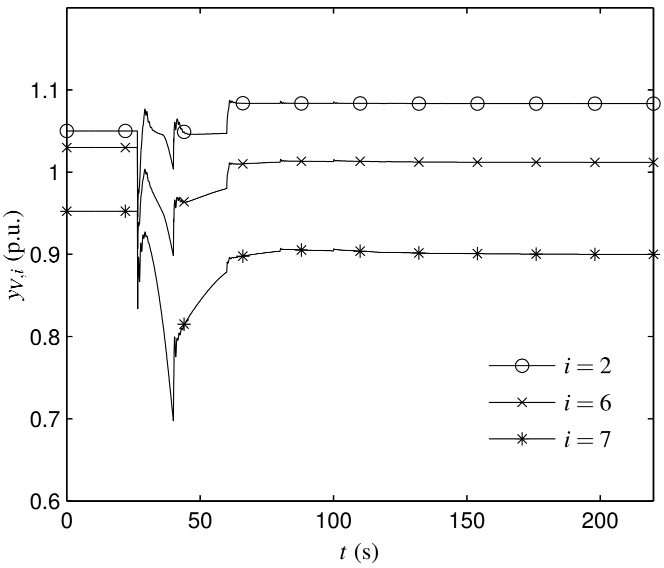

Figure 4.12: Voltage magnitude profiles for controlled simulation using the linear MPC formulation.

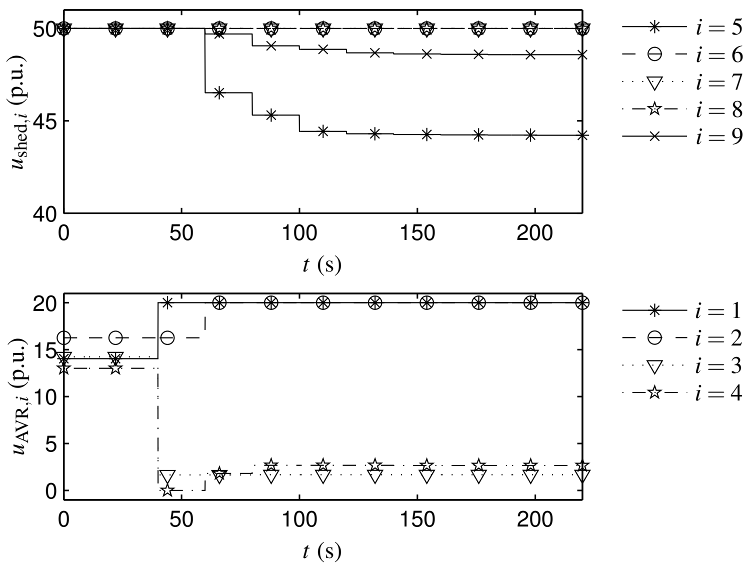

Figure 4.13: Set-points provided by the control agent using the linear MPC formulation for controlled simulation. Load shedding values are scaled to lie within 0 and 50. AVR set-points are scaled to lie within 0 and 20.

4.5 Summary

In this chapter we have discussed MPC in multi-layer control. In particular we have focused on issues related to the model that a medium-layer MPC control agent uses and discussed why object-oriented modeling is suitable for this. We have proposed an MPC strategy in which the prediction model is formulated either as an object-oriented model, allowing relatively easy construction of models of complex systems, or as a linearized approximation of such a model, allowing the use of efficient optimization problem solvers. Due to the nature of power networks, the object-oriented prediction model involves differential, algebraic, and logic relations and is nonlinear, non-smooth, and costly to evaluate.

To solve the nonlinear MPC problem of the medium-layer control agent using the constructed prediction model, we have proposed to use pattern search as optimization method. Pattern search is a direct-optimization method that does not compute or approximate gradients and/or Hessians, which are not available in analytical form in the situation considered. Moreover, due to the discrete elements, e.g., saturation, the MPC optimization problem is non-smooth, making approaches using gradient or Hessian information less suitable.

We have applied the proposed control strategy for the control agent in a medium control layer of a power network. The medium-layer control agent provides set-points to a lower control layer with the aim of preventing voltage collapses from occurring. Simulation studies on a 9-bus dynamic power network have shown the potential of the proposed approaches. For the MPC formulation based on the object-oriented model, we have illustrated the difference in performance between a gradient-based and the pattern search method and we have shown that the voltage collapses can be prevented from occurring. For the MPC problem based on the linearized model, we have compared the performance of the control for a specific example with the performance obtained by the MPC control agent using the original model. We have observed that the MPC control agent using the linearized prediction model can determine set-points that stabilize the voltage magnitudes, despite the linearized model used. Although the control actions that the MPC control agent using the linearized model chooses result in higher costs than the actions that the original MPC control agent would choose, the total computation time is significantly lower for the MPC control agent using the linearized model. It is therefore interesting to investigate further what the performance loss is due to the linearization, and for which type of disturbances the control agent using the linearized model can yield good performance.

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.