Hierarchical Model Predictive Control for Plug-and-Play Resource Distribution

Bendtsen, Jan Dimon; Trangbæk, K; Stoustrup, Jakob

Published in:
Distributed Decision Making and Control

DOI (link to publication from Publisher):
10.1007/978-1-4471-2265-4_15

Publication date:
2012

Document Version
Early version, also known as pre-print

Chapter 15
Hierarchical Model Predictive Control for Plug-and-Play Resource Distribution

Jan Bendtsen and Klaus Trangbaek and Jakob Stoustrup

Abstract This chapter deals with hierarchical model predictive control (MPC) of distributed systems. A three level hierarchical approach is proposed, consisting of a high level MPC controller, a second level of so-called *aggregators*, controlled by an online MPC-like algorithm, and a lower level of autonomous units.

The approach is inspired by smart-grid electric power production and consumption systems, where the flexibility of a large number of power producing and/or power consuming units can be exploited in a smart-grid solution. The objective is to accommodate the load variation on the grid, arising on one hand from varying consumption, on the other hand by natural variations in power production e.g. from wind turbines.

The proposed method can also be applied to supply chain management systems, where the challenge is to balance demand and supply, using a number of storages each with a maximal capacity. The algorithm will then try to balance the risk of individual storages running empty or full with the risk of having overproduction or unsatisfied demand.

The approach presented is based on quadratic optimization and possesses the properties of low algorithmic complexity and of scalability. In particular, the proposed design methodology facilitates plug-and-play addition of subsystems without redesign of any controllers.

The method is verified by a number of simulations featuring a three-level smart-grid power control system for a small isolated power grid.[1]

The three authors are with:
Department of Electronic Systems, Automation and Control, Aalborg University, Fr. Bajers Vej 7C, 9220 Aalborg, Denmark; e-mail: dimon@es.aau.dk,ktr@es.aau.dk,jakob@es.aau.dk

[1] This work is supported by The Danish Research Council for Technology and Production Sciences.

15.1 Introduction

We discuss a hierarchical setup, where an optimization-based high-level controller is given the task of following a specific externally generated trajectory of consumption and/or production of a certain resource. The high-level controller has a number of units under its jurisdiction, which consume a certain amount of the resource. The flow of resources allocated to each of these units can be controlled, but each unit must at all times be given at least a certain amount of the resource; vice versa, each unit can only consume a certain (larger) amount of the resource.

One can think of various practical examples of systems that fit with this setup; for instance a supply chain management system [2], where the challenge is to balance demand and supply, using a number of storages each with a maximal capacity. The algorithm will then try to balance the risk of individual storages running empty or full with the risk of having over-production or unsatisfied demand. Other examples include large-scale refrigeration systems (e.g., in supermarkets), where the resource is refrigerant and the consuming units are individual display cases [14]; irrigation systems, where the shared resource is water and the consuming units are adjustable field sprinklers [13]; chemical processes requiring process steam from a common source [5]; or even digital wireless communication systems, where the resource is bandwidth and the consuming units are hand-held terminals, e.g. connected to a building-wide intranet [1]. See also [6] and [3] for an example of a district heating system that shares some of the physical characteristics outlined here, although the cited papers pursued a decentralized control scheme rather than a centralized one.

Such large-scale hierarchical systems are often subject to frequent modifications in terms of subsystems that are added (or removed). This adds an important side constraint to design methodologies for controlling such systems: They should accommodate incremental growth of the hierarchical system in a way that is flexible and scalable. In essence, the design methodology should support a *plug-and-play control* architecture, see e.g. [12].

In many cases, a natural choice for the top-level controller is some sort of model-predictive controller (MPC) [11], [7], since systems of the kinds referred to above are multi-variable, subject to constraints and often involve considerable delays. Furthermore, some sort of reference estimate is often known in advance, e.g., from 24-hour electric power consumption traces, weather forecasts, purchase orders, etc. Unfortunately, the computational complexity of traditional MPC scales quite poorly with the number of states in the problem ($O(n^3)$), see e.g., [4]). Refer also to [9] for a recent contribution on MPC control for two-layer hierarchical control systems. In the type of problems considered above, this complexity growth places significant limits on how large systems a centralized solution can handle, as also pointed out in e.g. [10].

In this chapter, we propose a hierarchical control architecture that

- is based on a standard MPC solution at the top level;
- is able to accommodate new units without requiring modifications of the top-level controller;

- remains stable for an increasing number of units;
- facilitates plug-and-play addition of units at the bottom level, i.e., new units can be incorporated at the bottom level simply by registering with the unit at the level just above it.

Furthermore, the worst-case complexity is lower than for conventional centralised solutions, which means that the proposed scheme scales more 'reasonably' than the centralized solution. As will be illustrated, the involved optimization techniques give rise to quite sparse structures; this sparsity can be exploited to reduce complexity.
By a distributed resource control system we shall understand a system with the following characteristics:

- The system has a number of decentralized storages that can store a certain amount of some resource;
- Each storage can be filled or emptied at some maximal rate(s);
- A central controller has the responsibility of balancing supply and demand by use of the storages.

We illustrate the approach by a specific example, a so-called "smart grid" electric power system, where consumers can vary their power consumption within certain bounds by allowing devices to store more or less energy at convenient times [8]. The obvious method to do so physically is by exploiting large thermal time constants in deep freezers, refrigerators, local heat pumps, etc.; extra energy can be stored during off-peak hours, and the accumulated extra cooling can then be used—slowly—by turning compressors and similar devices on less frequently during peak hours. Implementing such schemes is considered a necessity for the adoption of large amounts of unpredictable renewable energy sources in the European power grid, and requires local measurement and feedback of current energy and power demand. Consumers equipped with such measurement and feedback capabilities are called *intelligent consumers*.

Structural flexibility of large-scale systems is important, since subsystems and components may be added, removed or replaced during the system's lifetime. In our example, it is easy to imagine customers wanting to sign up for a contract with a power company, such that the customer is assigned the necessary equipment to become an intelligent consumer. Thus, the top level system should be flexible enough to accommodate new consumers under its jurisdiction without it being necessary to perform significant re-tuning and/or restructuring every time new consumers appear. Furthermore, it is a basic requirement that the system is stable and provides good performance at all times.

The outline of the rest of the chapter is as follows: Section 15.2 explains the problem in a general setting, while Sec. 15.3 presents the proposed algorithm for resource sharing. Section 15.4 shows that the resulting architecture remains stable for increasing numbers of units. Section 15.5 shows a simulation example of the algorithm applied to an electric 'smart grid' with a small number of consumers and, finally, Sec. 15.6 offers some concluding remarks. Note that, unless otherwise stated, all time-varying quantities (signals) are assumed to be real scalars. Vectors and ma-

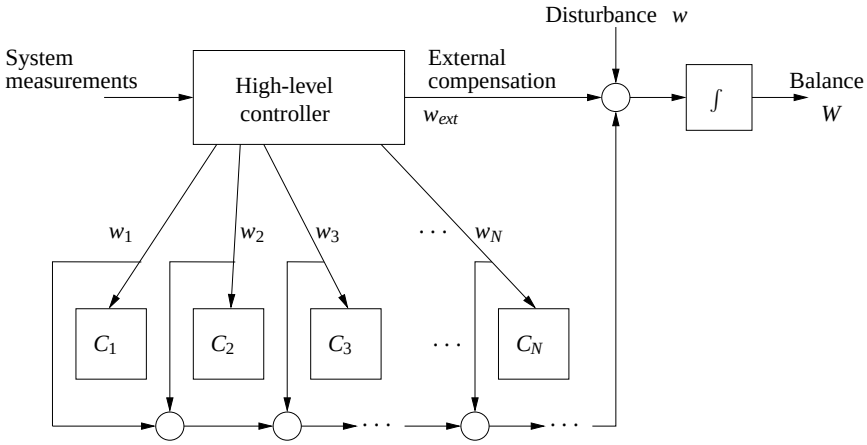

Fig. 15.1 Problem setup

trices are indicated with bold-face symbols, while sets are written in calligraphic font.

15.2 Problem Formulation

We consider a setup as depicted in Fig. 15.1. The high-level controller is given the task of following a specific externally generated trajectory of consumption and/or production of a certain resource. The objective is to maintain a certain system-level *balance* (between demand and production); the error in the balance is represented by the scalar signal $W(t)$, which must be driven to 0 as the time t tends to infinity. The demand and production must match over time, however, and the disturbance $w(t)$ is hence treated as short-time changes in the balance, whereas $W(t)$ is an integrated error signal. The high-level controller can compensate directly for the disturbance $w(t)$ by assigning some of the resource flow $w_{\text{ext}}(t)$ to this task, but at a significant cost. However, the high-level controller also has a number of units, which we will in general refer to as *consumers*, C_i, $i = 1,\ldots,N$, under its jurisdiction. Each one of these consumers consumes the resource at a certain, controllable rate $w_i(t)$. The high-level controller is able to direct time-varying resources to the consumers, but must ensure that each consumer *on average* receives a specific amount of the resource, and certain upper and lower bounds on the consumption rate, $\underline{w}_i$ and $\overline{w}_i$, may not be exceeded. By doing so, the consumption compensates for some of the disturbance $w(t)$, at a *lower cost* than the direct compensation signal $w_{\text{ext}}(t)$. That is, it is advantageous to utilise the consumers as much as possible, subject to the aforementioned constraints.

This setup could for instance be interpreted as a supply chain management system, where the challenge is to balance demand and supply by minimizing the excess supply $W(t)$. The demand and supply should in this interpretation be associated with the 'disturbance' signal $w(t)$, which can be thought of as short-term market fluctuations, supply irregularities etc. The system has a number of storages C_i available, each currently being filled at the rate $w_i(t)$. The maximal capacities and maximal filling/emptying rates of each storage should be exploited in such a way that the need for 'external compensation' $w_{\text{ext}}(t)$ is minimized. In this interpretation, w_{ext} corresponds, depending on the sign, either to having to rent external storages or to have to buy components from more expensive suppliers. Thus, the goal of the algorithm is to try to balance the risk of individual storages running empty against the risk of having over-production or unsatisfied demand.

In the following, let $\mathcal{I} = \{1, 2, \ldots, N\}$ denote an index set enumerating the consumers. The high-level controller must solve the following optimization problem at any given time t:

$$\begin{aligned} \min_{w_i, w_{\text{ext}}} \quad & \int_t^{t+T_h} \phi(W(\tau), w_{\text{ext}}(\tau), \frac{dw_{\text{ext}}}{dt}) d\tau \\ \text{s.t.} \quad & \underline{W} \leq W(\tau) \leq \overline{W} \\ & \underline{w_i} \leq w_i(\tau) \leq \overline{w}_i, \quad \forall i \in \mathcal{I} \end{aligned} \tag{15.1}$$

where $\underline{W}$ and $\overline{W}$ are constraints on the balance and $\phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ is a smooth, convex cost function of the balance error, the external resources that need to be acquired, and the changes in these resources (motivated by the fact that it is often more costly to acquire extra resources on a very short notice); ϕ is typically chosen as a linear or quadratic cost function. T_h is the prediction horizon of the controller. For simplicity, and without loss of generality, the consumption by the consumers is assumed cost-free.

Let $W_i(t)$ denote the amount of resource accumulated in C_i, and $\eta_i \geq 0$ denote a drain rate, respectively; η_i is assumed to be constant for simplicity. Each consumer is characterized by its own linear state equation:

$$\frac{dW_i(t)}{dt} = w_i(t) - \eta_i \tag{15.2}$$

which must satisfy $0 \leq W_i(t) \leq \overline{W}_i$ at all times. Note that this model implies that the consumers are mutually independent. The goal that each consumer receives a specific amount of the resource on average, may be expressed as the integral constraint

$$\frac{1}{T_{\text{res}}} \int_0^{T_{\text{res}}} |w_i(\tau) - \eta_i| d\tau = W_{i,\text{ref}} \tag{15.3}$$

where T_{res} is some appropriate time span. Obviously, we must require that $0 \leq W_{i,\text{ref}} \leq \overline{W}_i$.

Note that, since the dynamics contain only pure integrators, (15.1) can easily be approximated by a discrete time problem of the form

$$\begin{aligned} \min_{w_i, w_{\text{ext}}} \quad & \sum_{k=t/T_s+1}^{(t+T_h)/T_s} \phi(W(kT_s), w_{\text{ext}}(kT_s), w_{\text{ext}}((k-1)T_s)) \\ \text{s.t.} \quad & \underline{W} \leq W(kT_s) \leq \overline{W} \\ & \underline{w_i} \leq w_i(kT_s) \leq \overline{w_i}, \quad \forall i \in \mathcal{I} \end{aligned} \tag{15.4}$$

where T_s is the sampling time. For simplicity, we will often drop T_s from the notation in the sequel, writing e.g., $w(k)$ as shorthand for $w(kT_s)$.

In order to solve the optimization problem, the high-level controller in principle requires access to all states in the system, including the internal states $W_i(t)$. This may lead to a very heavy communication load on distributed systems if the number of consumers is significant. Furthermore, the computational complexity of the optimization problem grows rapidly with the number of consumers as well. This means that adding more consumers into the system may pose significant problems in practice. Thus, a purely centralized solution to the problem may be optimal in terms of maintaining the supply/demand balance, but is not desirable from a practical point of view.

15.3 Proposed Architecture

In the following we propose a new architecture for achieving the control objective that requires significantly less system-wide communication, while at the same time being more flexible with respect to changes in the number of consumers. We now consider the modified setup in Figure 15.2, where $w(t)$ is an external disturbance and $w_a(t) = \sum_{i=1}^{N} w_i(t)$ is the cumulative rate of resource absorbed by all C_i. As mentioned in the previous section, the main objective of the high-level control is to keep the resource balance governed by

$$\frac{dW(t)}{dt} = w(t) - w_{\text{ext}}(t) - w_a(t) \tag{15.5}$$

at zero. It is assumed that the top level controller can control $w_{\text{ext}}(t)$ directly and is constrained only by a rate limit, but we would like to keep the variations, i.e., the time derivative of $w_{\text{ext}}(t)$, small as well.

Between the controller and $N_A \leq N$ subsets of the intelligent consumers, we introduce a number of so-called *aggregators* $A_j, 1 \leq j \leq N_A$. Together, these aggregators serve as an interface between the top level and the intelligent consumers. To each aggregator A_j we assign a number of consumers identified by an index set $\mathcal{J}^j \subset \mathcal{I}$, where for all $k, j = 1, \ldots, N_A$ we have $\mathcal{J}^j \cap \mathcal{J}^l = \emptyset, l \neq j$, and $\cup_{j=1}^{N_A} \mathcal{J}^j = \mathcal{I}$. Let n^j denote the cardinality of $\mathcal{J}^j$, i.e., the number of consumers assigned to aggregator A_j. The objective of each aggregator is to make sure that:

- The maximum capacity is available for the upper level at any time instance;

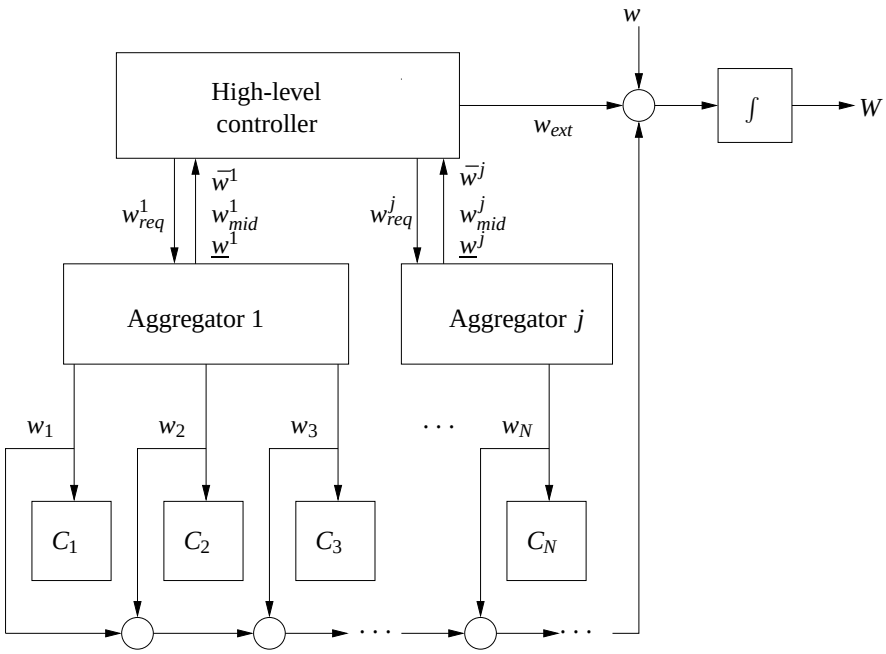

Fig. 15.2 Modified architecture

- The resources allocated to consumers are distributed roughly uniformly over the number of consumers;
- The deviation from the nominal consumption is minimized for each consumer;
- The capacity constraint for each consumer is not violated;
- The rate constraint for each consumer is not violated

As in the previous section, we approximate the continuous-time system with a discrete-time version.

The communication between the high-level controller is indicated on Fig. 15.2; each aggregator A_j provides the top level with a small number of simple parameters to specify the constraints of the consumers. In particular, the top level is informed of $\overline{w}(k)$ and $\underline{w}(k)$, which are current bounds on the cumulative resource that can be consumed by the consumers assigned to A_j, that is, bounds on

$$w_a^j(k) = \sum_{i \in \mathcal{J}^j} w_i(k)$$

that can be guaranteed over a specified horizon from time t. These limits necessarily depend on both resource storage rate and limitations among the individual consumers, and as such depend in a complicated fashion on the horizon length. Several choices can be made with respect to the length of this horizon. A simple choice is to provide the limits for one sample ahead. Various choices could be made here,

for instance providing a time-varying profile of limits over the control horizon, or the aggregators could simply provide fixed limits that can be sustained over the entire control horizon, although the latter would tend to be conservative. In addition to these limits, A_j provides $w^j_{\text{mid}}(k)$, a mid-ranging signal that informs the high-level controller of the total resource rate would currently be most helpful in bringing the intelligent consumers under its jurisdiction close to their reference resource levels $W_{i,\text{ref}}(k)$.

The aggregator level, as a whole, thus attempts to maintain $w_a(k) = \sum_{j=1}^{N_A} w^j_{\text{req}}(k)$ while the high-level controller, in turn, needs to solve the optimization problem

$$\begin{aligned} \min_{w^j_{\text{req}}, w_{\text{ext}}} \quad & \sum_{k=1}^{N_h} \phi(W(k), w_{\text{ext}}(k), w_{\text{ext}}(k-1)) + \beta \sum_{j=1}^{N_A} \sum_{k=1}^{N_h} (w^j_{\text{req}}(k) - w^j_{\text{mid}}(k))^2 \\ \text{s.t.} \quad & \underline{W} \leq W(k) \leq \overline{W} \\ & \underline{w}^j(k) \leq w^j_{\text{req}}(k) \leq \overline{w}^j(k), \quad 1 \leq j \leq N_A \end{aligned} \tag{15.6}$$

which is of significantly lower dimension than (15.1) because the number of decision variables is smaller (since $N_A < N$). The term

$$\beta \sum_{k=1}^{N_A} \sum_{k=1}^{N_h} (w^j_{\text{req}}(k) - w^j_{\text{mid}}(k))^2$$

is introduced to ensure that the high-level controller will assign resources to the aggregators such that the intelligent consumers can approach their desired levels of storage, away from their individual limits; β is a constant to be specified later.

That is, in periods where the load is relatively steady, the high-level controller will make w^j_{req} approach w^j_{mid}, thereby increasing the short term resource reserves for future load changes (positive or negative).

At each sample, the aggregator A_j solves the simple optimization problem

$$\begin{aligned} \min_{w_i} \quad & \sum_{i \in \mathcal{J}^j} (W_i(k+1) - W_{i,\text{ref}})^2 \\ \text{s.t.} \quad & \sum_{i \in \mathcal{J}^j} w_i(k) = w^j_{\text{req}}(k) \\ & \underline{w}_i \leq w_i(k) \leq \overline{w}_i \\ & 0 \leq W_i(k+1) \leq \overline{W}_i \end{aligned} \tag{15.7}$$

with $W_i(k+1) = W_i(k) + T_s w_i(k)$, where T_s is the sampling time.

15.4 Stability, Complexity and Performance Analysis

In this section, we shall discuss stability, complexity and performance of the architecture proposed above.

15.4.1 Stability

First, in order to assess stability, it is of course necessary to establish a sensible definition, especially as the main obstruction to the standard definition is the presence of constraints.
Intuitively, stability for a system of the type described above will mean that

- For constant inputs, all trajectories will tend to constant values;
- In steady state, a minimal number of constraints will be invoked;
- Wind-up behavior of system states is avoided for any bounded input set

In the following, we will give an outline of a procedure for constructing controllers that stabilize the system in such a way that it satisfies these properties. For ease of the presentation we will only consider one aggregator.

Suppose the system satisfies the following assumptions.

1. The external load is constant;
2. The number of intelligent consumers is non-decreasing;
3. Any new ICs that appear in the system start with an initial amount of resource in storage equal to $W_{i,\mathrm{ref}}$;
4. As long as the sum of all deviations from $W_{i,\mathrm{ref}}$ does not increase, the constraints $\underline{w}$ and $\overline{w}$ do not become narrower (i.e., $\underline{w}$ does not increase, and $\overline{w}$ does not decrease).

The last assumption is technical; we aim to choose the reference levels exactly such that the constraints are as wide as possible, thus making sure that this assumption is satisfied by design. Indeed, in order to accommodate the stability notions introduced above, we will modify the performance objective sligthly, so that we may be able to follow a standard dual mode approach to stability analysis of model predictive control with terminal constraints [7].

First of all, we note that the overall system is a linear, constrained system. Therefore, at the top level we consider the state vector

$$\mathbf{x}(k) = \begin{bmatrix} W(k) \\ w_{\mathrm{ext}}(k) - w(k) \\ W_{\Sigma}(k) \end{bmatrix}$$

where $W_{\Sigma}(k) = \sum_{i\in\mathcal{I}}(W_i(k) - W_{i,\mathrm{ref}})$ denotes the total amount of surplus resources in the ICs. Next, we define the function

$$l(k) = \mathbf{x}(k)^T \mathbf{Q}\mathbf{x}(k) + R\Delta w_{\mathrm{ext}}(k)^2$$

where $\mathbf{Q} \in \mathbb{R}^{3\times 3}$ and $R \in \mathbb{R}_+$ are constant weight factors, and $\Delta w_{\mathrm{ext}}(k) = w_{\mathrm{ext}}(k+1) - w_{\mathrm{ext}}(k)$. If $\mathbf{Q}$ is chosen as a symmetric, positive definite matrix, it is easily seen that l is a positive definite, smooth, unbounded function of $\mathbf{x}$ with minimum in $\mathbf{x} = 0$. Based on this function, we define the function

$$V(k_0) = \sum_{k=k_0+1}^{\infty} l(k) \tag{15.8}$$

along with the control optimization

$$\begin{aligned} \min_{w_{ext}, w_{req}} \quad & V(k_0) \\ \text{s.t.} \quad & W_\Sigma(k_0 + N_h) = 0 \\ & \underline{w}(k) \leq w_{\text{req}}(k) \leq \overline{w}(k) \end{aligned} \tag{15.9}$$

where (15.9) is a terminal constraint.

Given the properties of $l(k)$, we see that V can be used as a Lyapunov function, i.e., if we can ensure that it decreases every sample, the closed loop will be stable.

Assuming that the constraints are not active after $k_0 + N_h$, the optimal trajectory will be described by the dynamics $\mathbf{x}(k+1) = \tilde{\mathbf{A}}\mathbf{x}(k)$, where $\tilde{\mathbf{A}}$ can be found as the closed loop matrix resulting from a standard LQR problem. By construction, all eigenvalues of $\tilde{\mathbf{A}}$ have modulus less than one, so we can find a symmetric positive definite matrix $\mathbf{Q}$ that solves the discrete Lyapunov equation

$$\tilde{\mathbf{A}}^T \bar{\mathbf{Q}} \tilde{\mathbf{A}} = \bar{\mathbf{Q}} - \mathbf{Q}$$

We can then write

$$V(k_0) = \sum_{k=k_0+1}^{k_0+N_h} l(k) + \sum_{k=k_0+N_h+1}^{\infty} l(k) = \sum_{k=k_0+1}^{k_0+N_h} l(k) + \mathbf{x}(k_0+N_h)^T \bar{\mathbf{Q}} \mathbf{x}(k_0+N_h) \tag{15.10}$$

which means that we perform the optimization on a finite horizon with an extra weight on the terminal state. In order to realize that V is decreasing, it is enough to note that as the horizon recedes, more flexibility is added to the optimization, meaning that $\min V(k_0+1) \leq \min V(k_0) - l(k)$.

The reason that we can assume that constraints are not active at the end of the control horizon follows from Assumption 4 and the terminal constraint (15.9).
Thus, under the assumptions above, we can say the following:

- Each aggregator drives the amount of resource stored in the ICs under its jurisdiction towards their reference values.
- In steady state, the minimal number of constraints are active. This follows from the properties of quadratic optimization; if the number of active constraints is non-minimal, the quadratic cost will always become smaller by shifting load from one of the subsystems with an active constraint to one or more subsystems with inactive constraints.
- Wind-up behavior of system states is avoided for any bounded input set, since all the individual subsystems are open loop (marginally) stable.

It should finally be noted that the terminal constraint (15.9) was only included in the above to make the argumentation easier; it does not appear to be necessary in practical implementations and has not been included in the examples in Sec. 15.5.

15.4.2 Complexity

In terms of complexity, the proposed design methodology scales in the same way as quadratic optimization, which is $O(N^3)$, where N is the number of consumers.

It should be noted, however, that the optimization problem has a high degree of sparsity. This has not been exploited in the implementation applied in the simulations below, but it should be expected that the complexity could be further reduced by implementing a dedicated quadratic programming solver, which exploits the mentioned sparsity. Further considerations on complexity can be found in Sec. 15.5.2.

15.4.3 Performance

In terms of performance, the following five parameters are decisive:

- The prediction horizon;
- The total installed flexible capacity;
- The instantaneous flexible capacity;
- The total cumulative rate limitation of flexible units;
- The instantaneous cumulative rate limitation of flexible units

The prediction horizon is a crucial parameter, since the overall balance W is the result of an integration. This means that for longer horizons the potential of using flexible units becomes much larger, since the slack variables relative to the saturation limits needed for guaranteed stabilization of the integrator becomes much smaller for a longer time horizon.

The total installed flexible capacity is the sum of maximal resource storages for all units, i.e. $C_{\text{tot}} = \sum_{i=1}^{N} \overline{W}_i$. This capacity clearly scales with the number of units.

The instantaneous flexible capacity is the present unexploited part of C_{tot}. Since flexibility is exploited bidirectionally in reaction to either increasing or decreasing load, C_{tot} has to be divided between upwards and downwards movement. The dynamics of this quantity depends on the control algorithm and of the control horizon. Due to the additive nature of the quadratic programming cost function, the instantaneous capacity for the proposed algorithm scales linearly with the number of units, which is clearly optimal.

The total cumulative rate limitation of flexible units is the rate limitation experienced by the high level controller and equals $\sum_{i=1}^{N} \overline{w}_i$ for positive load gradients and $\sum_{i=1}^{N} \underline{w}_i$ for negative load gradients. This parameter scales linearly with the number of installed units.

The instantaneous cumulative rate limitation of flexible units is current rate limitation experienced by the high level controller and is equal to the sum of individual rate limits for those units, which are not in saturation. Again, due to the additive nature of quadratic programming costs, the instantaneous rate limitation scales linearly with the number of installed units. The average ratio (for a given load pat-

tern) between instantaneous and total cumulative rate limitations is controlled by the weighting factor ρ, which constitutes the trade-off between capacity limitation and rate limitation. For a given load pattern, more average capacity can be obtained at the cost of rate limitation and vice versa, but the quadratic optimization guarantees Pareto optimality.

The sampling times used at aggregator and top levels also influences performance. Since the dynamics consist entirely of pure integrators, there is no approximation in the discretization itself, but of course the flexibility in the optimisation will be smaller for a larger sampling time.

15.5 Simulation Example

The example described in this section is inspired by a vision for future Smart Grid technologies called Virtual Power Plants, which is depicted in Figure 15.3.

The main objective of the top level control is to keep the energy balance governed by

$$\frac{dE(t)}{dt} = P_{\text{ext}}(t) - P_{\text{load}}(t) - P_{\text{a}}(t) \tag{15.11}$$

at zero. $P_a = \sum_i P_i$ is the power absorbed by the intelligent consumers (ICs). P_{load} is the power absorbed by other consumers, and is considered as a disturbance here. P_{ext} is the power produced by a number of suppliers such as power plants etc.. It is assumed that the top level controller can control P_{ext} directly and restrained only by a rate limit, but we would also like to keep the time derivative small.

Each intelligent consumer is characterized by its own energy balance

$$\frac{dE_i(t)}{dt} = P_i(t) \tag{15.12}$$

which must satisfy $0 \leq E_i(t) \leq \overline{E}_i$ at all times. Furthermore, each intelligent consumer can only consume a limited amount of power $\underline{P}_i \leq P_i(t) \leq \overline{P}_i$. The aggregator serves as an interface between the top level and the ICs. It attempts to maintain $P_a(t) = P_{\text{req}}(t)$ and provides the top level with simple parameters to specify the constraints of the ICs. In particular, the top level is informed of $\overline{P}$ and $\underline{P}$, upper and lower limits on P_a that can be guaranteed over the horizon N_l. These limits depend on both power and energy storage limitations, and as such depend in a complicated fashion on the horizon length. In addition to the limits, the aggregators provide P_{mid}, a mid-ranging signal which tells the top level which P_{req} would be most helpful in bringing the ICs close to their reference energy levels $E_{ref,i}$. In periods where the load is relatively steady, the top level can then prioritize keeping the energy levels at the reference, and thereby increasing the short term reserves for future load changes.

How to choose these reference levels is again a complicated question of the considered horizon. If we consider a long horizon, then we might like to have the same energy reserve in both directions, which would lead to $E_{ref,i} = \overline{E}_i/2$. On the other

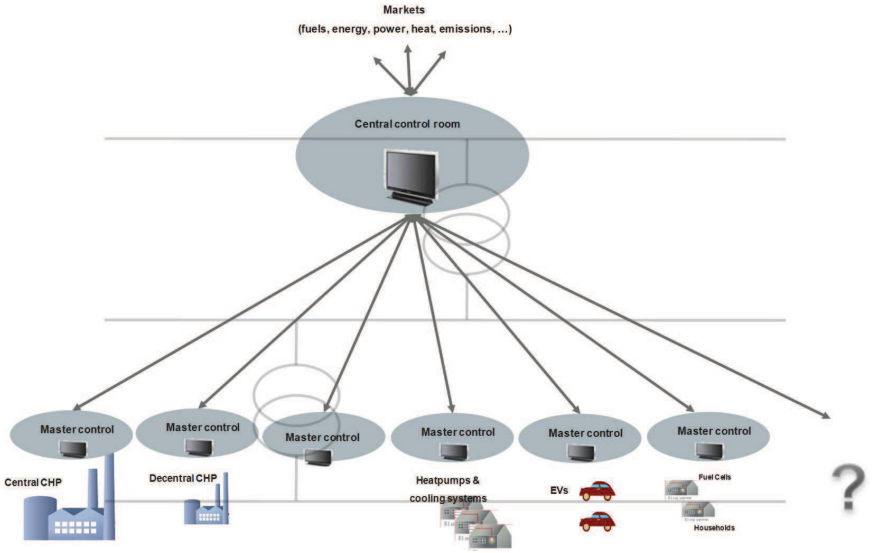

Fig. 15.3 A vision for Smart Grids: Virtual Power Plants which aggregate producing or consuming units.

hand, some ICs have a much higher $\overline{P}$ than $-\underline{P}$, and are therefore much better at providing a positive than negative absorption, while others are better at providing negative absorption. With a short horizon it would make sense to keep the first kind at a low energy level, and vice versa. Here, we choose

$$E_{ref,i} = \overline{E}_i \frac{\overline{P}_i}{\overline{P}_i - \underline{P}_i}$$

which corresponds to making the time to fill the energy reserves equal to the time to fully empty it.

At each sample, at time t, the aggregator solves the simple optimization problem

$$\begin{aligned} \min_{P_i} \quad & \sum \quad (E_i(t+T_s) - E_{i,\text{ref}})^2, \\ & \text{s.t.} \\ \sum P_i \quad &= \quad P_{\text{req}}, \\ \underline{P}_i \quad &\leq \quad P_i(t) \leq \overline{P}_i, \\ 0 \quad &\leq \quad E_i(t+T_s) \leq \overline{E}_i \end{aligned}$$

with $E_i(t+T_s) = E_i(t) + T_s P_i$, thereby distributing the power in a way that brings the energy levels as close to the reference as possible in a quadratic sense.

The top-level control optimises over a prediction horizon N_p. It minimizes the performance function

$$J_t = \sum_{k=1}^{N_p} E(t+T_s k)^2 + \beta_p \sum_{k=1}^{N_c} (P_{\text{ext}}(t+T_s k) - P_{\text{ext}}(t+T_s(k-1)))^2 + \beta_r \sum_{k=1}^{N_c} (P_{\text{req}}(t+T_s k) - P_{\text{mid}}(t))^2$$

with N_c samples of P_{ext} and P_{req} as decision variables.

The optimization is subject to constraints on the decision variables. There is a rate limit on the power from the power plants:

$$\underline{P}_{\text{ext}} \leq P_{\text{ext}}(t+T_s k) - P_{\text{ext}}(t+T_s(k-1)) \leq \overline{P}_{\text{ext}}$$

As mentioned, the aggregator provides limits on P_a that can be sustained over a horizon N_l. These limits are conservative in the sense that if P_{req} is for instance negative for the first part of the horizon, then a positive P_{req} higher than $\overline{P}$ may be feasible for the rest. However, in order to simplify the top level computations, the constraint $\underline{P}(t) \leq P_{\text{req}}(t+kT_s) \leq \overline{P}(t)$ is imposed over the whole horizon. A

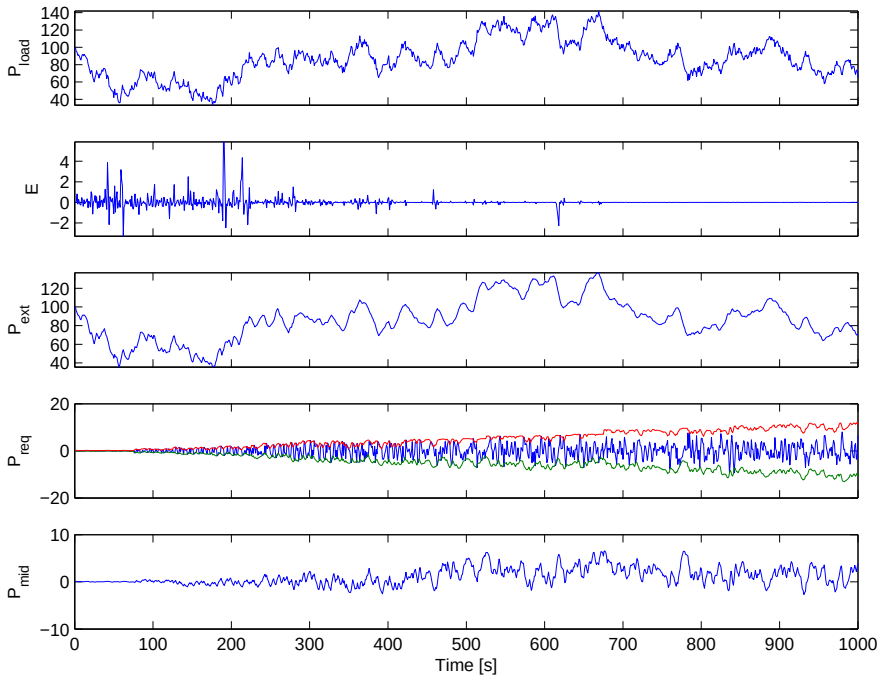

Fig. 15.4 Simulation example.

simulation of this scheme is shown in Fig. 15.4. The controller parameters used are $T_s = 1$, $N_l = N_c = 4$, $N_p = 5$, $\beta_p = 0.1$, $\beta_r = 10^{-4}$. The load is generated by a first order auto-regressive process with a time constant of 100 seconds, driven by zero-mean Gaussian white noise with unit variance. There are 20 ICs with parameters shown in Table 15.1 becoming available as time passes, making it possible for the aggregator to provide increasingly wider constraints on P_{req}. The result is that the energy balance can be controlled much better while also using a smoother P_{ext}. The requested consumption P_{req} is shown together with $\underline{P}(t)$ and $\overline{P}(t)$, computed by the aggregator. It is noted how the constraints widen as more ICs become available, but will shrink when the reserve is being used. P_{mid} is computed as the P_{req} that would bring the energy levels to the reference in N_l samples, ignoring power limits.

Table 15.1 Parameters for 20 consumers in simulation

i	1	2	3	4	5	6	7	8	9	10
$\overline{E}_i$	1.0	4.0	4.0	3.0	6.0	10.0	1.0	4.0	9.0	10.0
$\underline{P}_i$	-1.7	-1.4	-0.2	-1.3	-1.6	-1.3	-0.7	-1.9	-1.1	-1.1
$\overline{P}_i$	1.4	0.8	1.8	0.3	0.9	1.1	1.2	0.2	0.2	0.2

i	11	12	13	14	15	16	17	18	19	20
$\overline{E}_i$	9.0	1.0	2.0	10.0	6.0	1.0	9.0	8.0	2.0	9.0
$\underline{P}_i$	-0.2	-1.0	-1.6	-1.3	-0.3	-1.1	-1.9	-0.2	-0.9	-1.6
$\overline{P}_i$	1.1	1.2	1.6	1.9	0.4	0.9	1.8	0.8	0.6	0.5

The energy balance of the ICs is shown in Fig. 15.5. The energy constraints and reference are shown by dashed lines. It can be seen that additional consumers are "plugged in", the system automatically incorporates these new consumers and these new resources are exploited throughout the control hierarchy in order to improve the power balance at the top level.

15.5.1 Performance Study

The aggregators perform two functions, approximation and aggregation. The main purpose is the aggregation of several consumers into a simple virtual big consumer, thereby simplifying the task at the top level. The approximation, simplifying the power limits into an average over the horizon, is only necessary in order to facilitate the aggregation. In fact, if each aggregator has only one consumer, then the computation at the top level is not simplified at all. In the next section, the effects of aggregation on the performance will be studied, but before that the question arises of how much the conservative approximation affects performance compared to a centralised scheme as in Fig. 15.1, where the high-level controller directly controls the consumers. We compare two control schemes:

Centralized controller: The top level controller optimises a standard MPC objective

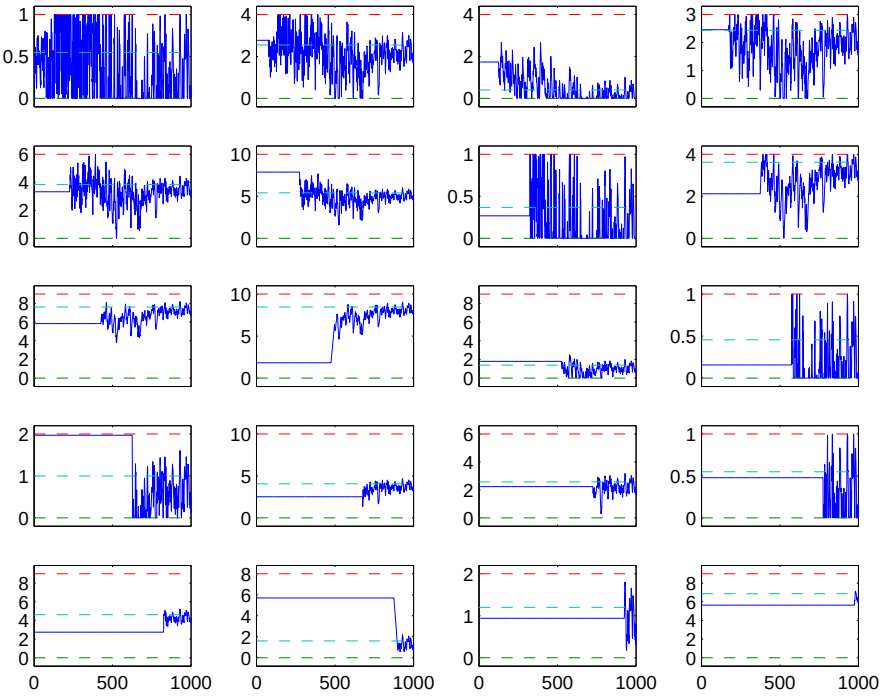

Fig. 15.5 Simulation with aggregators.

$$\min_{P_i} \sum_{k=t+1}^{t+N_p} (E(k)^2 + \beta_p (P_{ext}(k) - P_{ext}(k-1))^2 + \beta_e \sum (E_i(k) - E_{ref,i})^2)$$

for $i = 1, \ldots, N$ over the consumption rates of all consumers over a horizon N_p. The control is not fully optimal, since the horizon is finite. Therefore, the last term is used for keeping the energy levels close to the references if the reserves are not needed immediately.

Approximating control: The scheme described in the previous section, but each aggregator handles only one consumer. In this way, the comparison will reflect the effects of the approximation.

We perform simulations on a system with a small number of consumers, N. The consumer power limits are evenly distributed between $\pm 0.4/N$ and $\pm 2.4/N$, the maximum energy levels between 0.8 and 4.8. $\underline{P}_{\text{ext}} = -0.5$, $\overline{P}_{\text{ext}} = 0.5$. The load follows the same behavior as in the above example.

The approximating control has the same parameters as in the above example. The centralized controller has the same control horizon and performance weights, and $\beta_e = 10^{-4}$.

For each particular N, 100 simulations of 200 samples are performed. For each simulation the ratio of variances is computed. Figure 15.6 shows the mean of these

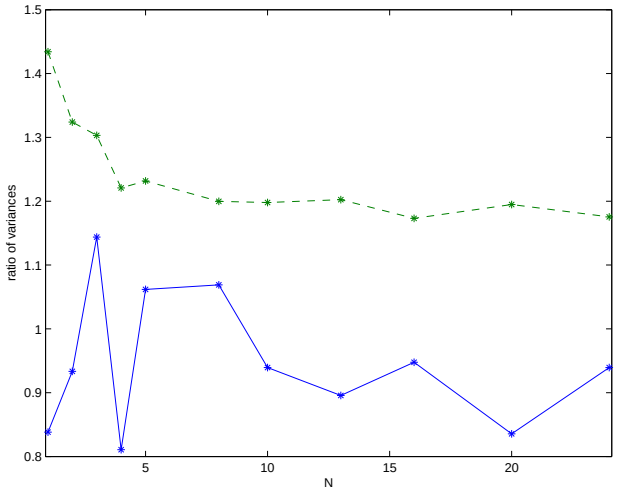

Fig. 15.6 Performance comparison with a centralised control for increasing number of aggregators. Solid: ratio of variances of balance. Dashed: ratio of variances of derivative of external power.

ratios as N grows. The solid line shows the ratio between the variance of E when using the approximating control and when using the centralized control. The dashed line shows the same but for the variance of $P_{\text{ext}}(k) - P_{\text{ext}}(k-1)$. It is noted that the ratios are quite close to 1, meaning that the performance of the approximating control is almost as good as for the centralized control. Importantly, the ratios seem to decrease towards 1 as N grows. It was not feasible to perform the simulations for higher N, as the computational complexity grew too high. The result leads us to conjecture that the approximation only has a small effect on the performance, and that this effect is unimportant for a large number of consumers.

15.5.2 Complexity Study

The aggregators serve as a simplifying interface to the relatively complex top level control, and as such even a configuration with one aggregator is computationally less complex than letting the top level control work on a full model. However, for a large number of ICs, the aggregator can also itself become too complex. It is therefore necessary to allow for more than one aggregator. This also provides the top level with more detailed information and can therefore be expected to yield better performance. On the other hand, more aggregators will make the top level control more complex, so there is a trade-off between complexity at the top and aggregator levels and also with respect to performance.

Here, we examine the effects of the number of aggregators, N_A, through a simulation example. We consider a situation with 800 ICs with E_is evenly distributed

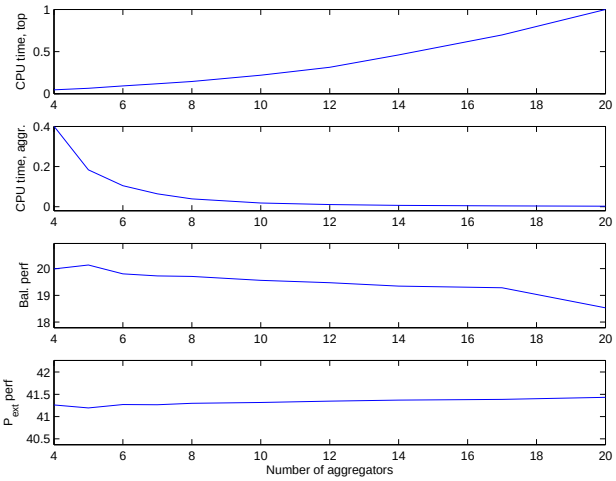

Fig. 15.7 CPU times for simulation with varying numbers of aggregators.

between 0.01 and 0.14, and $\overline{P}_i$s and $-\underline{P}_i$s evenly distributed between 0.01 and 0.06. The other parameters are $T_s = 1$, $N_l = N_c = 4$, $N_p = 5$, $\beta_p = 1$, $\beta_r = 10^{-4}$. The load is generated by a discrete time process $(1-0.99q^{-1})(P_{load}(k)-100) = e(k)$, where q^{-1} is the delay operator and e is white Gaussian noise with variance 16.

In all the simulations the same 400 sample load sequence was used, only N_A was changed (Fig. 15.7) shows the result. The top plot shows the (scaled) time consumption of the top level controller. This grows with N_A^3. The second plot uses the same scaling and shows the average time consumption of each of the aggregators. As the number of ICs handled by each aggregator is inversely proportional to the number of aggregators, this consumption is inversely proportional to N_A^3. It is noted that the computational complexity of the top level and of each of the aggregators is approximately equal with around 6 aggregators, so this may be a sensible choice.

The variance of the balance E and of the derivative of P_{ext} are shown in the next two plots. As expected, more aggregators give better performance, but the difference is rather small.

15.6 Discussion

In this chapter a design methodology for a three level hierarchical control architecture is proposed. The emphasis is on systems that accumulate the production and/or consumption of resources through the levels, exemplified by irrigation systems, sewer systems, or power production and consumption systems.

The presented solution is based on MPC-like algorithms, based on online quadratic programming solvers. The algorithmic complexity is very low and approxi-

mately scales with the number of units in the system to the power of 1.5, even without exploiting a significant sparsity of the optimization problems involved.

The approach has the specific feature that it facilitates online modifications of the topography of the controlled system. In particular, units at the lower level can be added or removed without any retuning of any controllers. This plug-and-play control property is enabled by the modular structure of the involved cost functions of the optimizations.

The proposed methodology is exemplified by a simulation of a control system for a small electrical power production and consumption system, where the power flexibility of a number of consumers is exploited. For this example, a significant improvement of flexibility at the grid level is obtained.

References

1. Beckmann, C.J.: Modeling integrated services traffic for bandwidth management in next-generation intranets. In: Proc. 1997 5th Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (1997)
2. De Kok, A., Graves, S.: Supply chain management: Design, coordination and operation. In: A. De Kok (ed.) Handbooks in Operations Research and Management Science: Vol. 11. Elsevier (2003)
3. De Persis, C., Kallesoe, C.S.: Quantized controllers distributed over a network: An industrial case study. In: Proc. Mediterranean Control Conference '09. Thessaloniki, Greece (2009)
4. Edlund, K., Sokoler, L.E., Jørgensen, J.B.: A primal-dual interior-point linear programming algorithm for MPC. In: Proc. 48th IEEE Conf. Decision and Control (CDC2009), pp. 351–356. Shanghai, China (2009)
5. Karlsson, K., Engstrom, I.: Controlling power and process steam. International Power Generation **26**, 29 (2003)
6. Knudsen, T.: Incremental data driven modelling for plug-and-play process control. In: Proc. 47th IEEE Conf. Decision and Control (CDC2008). Cancun, Mexico (2008)
7. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall (2002)
8. Moslehi, K., Kumar, R.: A reliability perspective of the smart grid. IEEE Transactions on Smart Grid **1**(1), 57–64 (2010)
9. Picasso, B., De Vito, D., Scattolini, R., Colaneri, P.: An MPC approach to the design of two-layer hierarchical control systems. Automatica **46**(5), 823–831 (2010)
10. Rao, C.V., Campbell, J.C., Rawlings, J.B., Wright, S.J.: Control of induction motor drives. In: Proc. Int. Conf. Energy Effective and Smart Motors - Development Perspectives and Research Needs. Aalborg University, Institute of Energy Technology, Aalborg University, Institute of Energy Technology, Pontoppidanstræde 101, 9220 Aalborg Ø (1992)
11. Rossiter, J.A.: Model-based predictive control. CRC Press (2003)
12. Stoustrup, J.: Plug & play control: Control technology towards new challenges. European J. Control **15**(4), 311–330 (2009)
13. Zhang, Y., Zhang, J., Guan, J.: Study on precision water-saving irrigation automatic control system by plant physiology. In: Proc. IEEE Conf. Industrial Electronics and Applications, pp. 1296–1300 (2009)
14. Zheng, L., Larsen, L.F.S., Izadi-Zamanabadi, R., Wisniewski, R.: Analysis of synchronization of supermarket refrigeration system- benchmark for hybrid system control. Kybernetes (2009)

Acknowledgement: This research was supported by LCCC—Linnaeus Grant VR 2007-8646, Swedish Research Council.