Hierarchical Control for Smart Grids

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

Published in:
Proceedings of the 18th IFAC World Congress, 2011

Publication date:
2011

Document Version
Early version, also known as pre-print

Hierarchical Control for Smart Grids

Klaus Trangbaek * **Jan Bendtsen** * **Jakob Stoustrup** *

* *Department of Electronic Systems, Automation and Control, Aalborg University, Fr. Bajers Vej 7C, 9220 Aalborg, Denmark;*
{ktr,dimon,jakob}@es.aau.dk

Abstract: This paper deals with hierarchical model predictive control (MPC) of smart grid systems. The design consists of a high level MPC controller, a second level of so-called *aggregators*, which reduces the computational and communication-related load on the high-level control, and a lower level of autonomous consumers.
The control system is tasked with balancing electric power production and consumption within the smart grid, and makes active use of the flexibility of a large number of power producing and/or power consuming units. The objective is to accommodate the load variation on the grid, arising on one hand from varying consumption, and on the other hand by natural variations in power production e.g. from wind turbines.
The high-level MPC problem is solved using quadratic optimisation, while the aggregator level can either involve quadratic optimisation or simple sorting-based min-max solutions.
In this paper we compare the performance and computational complexity of these two solutions and find that the performance of the two algorithms are very similar, whereas the sorting-based algorithm is much faster than the quadratic optimisation-based algorithm, thus allowing to handle vastly larger numbers of consumers.

Keywords: Smart Grids, Model Predictive Control, Hierarchical Control

1. INTRODUCTION

One of the greatest challenges in introducing large ratios of renewable energy into existing electric power power systems, is the fluctuating and unpredictable nature of power sources that harvest energy from natural sources such as wind turbines, solar plants etc. Essentially, since electric power is difficult to store, it must be available where and when it is needed by consumers supplied by the power system. Since the renewable sources of energy are difficult to control, base load units (e.g., fossil fuel-fired co-generation plants) must be kept in reserve, to compensate for temporary shortages. The higher the percentage of renewable sources, and the more fluctuating the power demands, the harder the regulation task becomes for the base load units (see e.g., Banakar et al. (2008)).

A so-called "smart grid" is an electric power system, where both producers and consumers are equipped with control capabilities that allow them to participate in these balancing efforts, for instance by allowing local devices with large time constants to store more or less energy at convenient times and thereby adjusting the momentary consumption, see e.g., Moslehi and Kumar (2010). The obvious method to do so 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.

Obviously, local control capabilities require local measurement and feedback of current energy and power demand. Consumers equipped with such measurement and feedback capabilities are called *intelligent consumers* (IC). Considering the fact that the consumers have local control capabilities, while a systems operator or supplier at the same time has the responsibility of maintaining balance between production and demand, the control architecture naturally becomes *hierarchical* (see, e.g., Scattolini (2009) and the references therein).

Since power systems are multi-variable and subject to constraints, and future reference estimates are often known in advance, e.g., from 24-hour power consumption traces [1] , weather forecasts, etc., a natural choice for the top-level controller will in most case be some sort of model-predictive controller (MPC) – see for instance Rossiter (2003), Maciejowski (2002), or Picasso et al. (2010). Unfortunately, the computational complexity of traditional MPC scales quite poorly with the number of states in the problem ($O(n^3)$), see e.g., Edlund et al. (2009)). In the type of problems considered above, this complexity growth places significant limits on how large systems a centralised solution can handle, as also pointed out in e.g. Rao et al. (1992).

In an earlier paper (Trangbaek et al. (2010)), the authors proposed a hierarchical control architecture for this type of problem, which

- 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

The design relies on injecting a second level of so-called *aggregators*, which reduce the computational and communication-related load on the high-level control, between the high-level controller and the lower level of autonomous consumers. Such an aggregator could for example act like a simple "Virtual

[1] Such traces are typically traded on spot market sales, see for instance http://www.nordpoolspot.com/

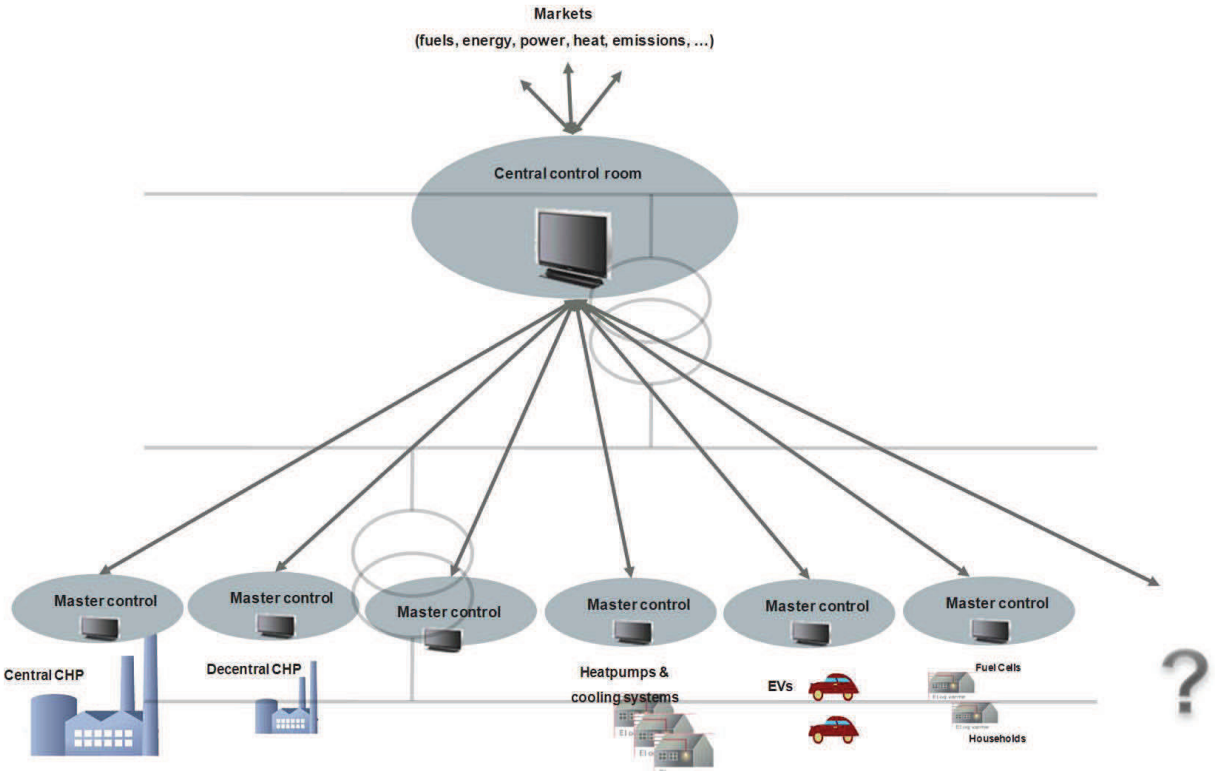

Fig. 1. A vision for smart grids: Virtual Power Plants which aggregate producing or consuming units

Power Plant", see Figure 1, i.e. an entity that seen from the cental control perspective behaves as a power plant in that it can produce (e.g. by refraining from consuming) a requested amount of power.

In Trangbaek et al. (2010) the aggregators solved a local quadratic optimisation problem, which works well for moderate numbers of intelligent consumers. However, as the number of consumers assigned to each aggregator grows large, the computational burden may grow restrictive. Therefore, in this paper, we investigate an alternative scheme based on simple min-max-based sorting algorithms in order to determine if the computational burden can be kept low while still maintaining good performance.

The outline of the rest of the chapter is as follows. Sections 2 and 3 briefly recount the simplified smart grid setting and the previous algorithm, while Section 4 presents the alternative algorithm proposed here and compares the two algorithms in terms of performance and computational burden. Finally, Section 5 offers some concluding remarks.

2. PROBLEM SETUP

The solutions described in this paper are based on a contract-based approach to Smart Grids. The underlying assumption is that an operator responsible for the balance has contracts with a number of consumers, where the operator offers reduced rates for electricity in return of direct access to consumer flexibility. Often this flexibility will relate to the operation of e.g. thermal systems (heat pumps, floor heating systems, supermarket refrigeration systems, cold stores, etc), where the significant time constants mean that a significant proportion of the consumption can easily be urged or postponed in a time range up to several hours without compromising the functionality of the thermal systems. Large-scale experiments of this type of Smart Grid approach has been carried out several places in the world, e.g. in Colorado in the United States and in Denmark in Europe. In principle, the methods of this paper could also be applied to Smart Grid approaches based on price-signalling, but due to the stochastic nature of the consumers, the parameters needed in the methods could be quite difficult to obtain in practice in a price-signalling setup.

We consider a setup as depicted in Figure 2, where the challenge is to balance demand and supply by keeping the grid-level energy balance governed by

$$\frac{dE(t)}{dt} = P_{\text{ext}}(t) - P_{\text{load}}(t) - P_{\text{a}}(t) \tag{1}$$

at zero. Let $\mathscr{I} = \{1, 2, \ldots, N\}$ denote an index set enumerating all the consumers in the system. $P_{\text{a}} = \sum_{i \in \mathscr{I}} P_i$ is the power absorbed by the intelligent consumers C_i at time t. P_{ext} is power that will have to be regulated by other (virtual) power plants under the portfolio controller's jurisdiction; this is assumed to be costly. P_{load} is the power produced by a number of external suppliers, such as wind turbines etc., and is considered as an uncontrollable disturbance here. It is assumed that the top level controller can control P_{ext} directly and is only restrained by a rate limit, but since rapid changes in the production output are not desirable due to extra stress on boiler units, mills etc., we would also like to keep the time derivative small.

The top level control thus optimises the following performance function over a prediction horizon N_p:

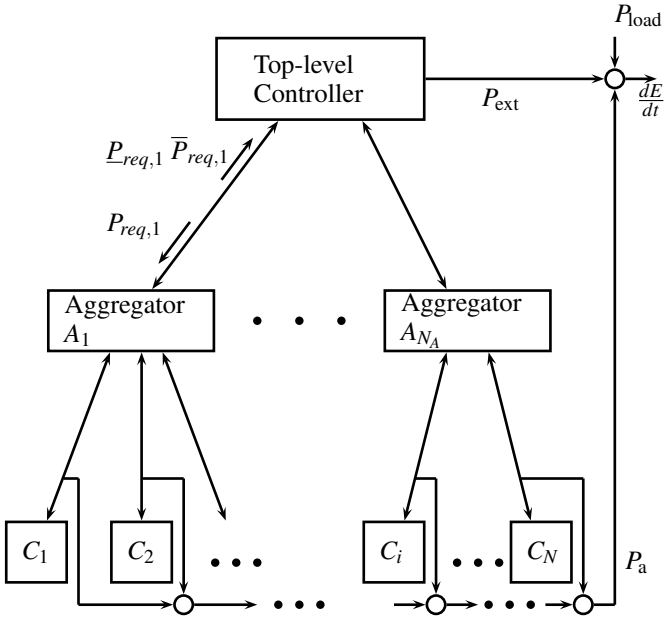

Fig. 2. Control architecture with grid-level control, aggregators and intelligent consumers

$$
\begin{aligned}
J_t = &\sum_{k=1}^{N_p} E(t+T_s k)^2 \\
&+ \beta_p \sum_{k=1}^{N_c} \left(P_{\text{ext}}(t+T_s k) - P_{\text{ext}}(t+T_s(k-1))\right)^2 \\
&+ \beta_r \sum_{j=1}^{N_A} \sum_{k=1}^{N_c} \left(P_{\text{req,j}}(t+T_s k) - P_{\text{mid,j}}(t)\right)^2
\end{aligned}
$$

with N_c samples of P_{ext} and P_{req} as decision variables. β_p and β_r are tuning weights, while T_s is the sampling time. P_{req} and P_{mid} are signals to and from the aggregator and will be explained below.

The optimisation is subject to constraints on the decision variables. Limits on the available P_{req}s are provided by the aggregators. The rate limit on the power from the power plants is modelled as follows:

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

Each intelligent consumer is characterised by its own energy balance

$$\frac{dE_i(t)}{dt} = P_i(t) \tag{2}$$

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 $\underline{P}_i \leq P_i(t) \leq \overline{P}_i$ at any given time.

In order to solve the optimisation problem, the high-level controller in principle requires access to all states in the system, including the internal states E_i. This may lead to a very heavy communication load on distributed systems. Furthermore, the computational complexity of the optimisation problem grows rapidly with the number of consumers N. This means that adding more consumers into the system may pose significant problems in practice. Thus, a purely centralised 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.

To alleviate this, we introduce a number of so-called *aggregators* $A_j, 1 \leq j \leq N_A$, between the controller and $N_A \leq N$ subsets of the intelligent consumers. 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 $\mathscr{J}^j \subset \mathscr{I}$, where for all $k, j = 1, \ldots, N_A$ we have $\mathscr{J}^j \cap \mathscr{J}^k = \emptyset, k \neq j$, and $\cup_{j=1}^{N_A} \mathscr{J}^j = \mathscr{I}$.

The aggregators serve as an interface between the top level and the ICs. Aggregator j attempts to maintain $P_{a,j}(t) = P_{\text{req,j}}(t)$ and provides the top level with simple parameters to specify the constraints of the ICs. In particular, the top level is informed of $\overline{P}_{req,j}$ and $\underline{P}_{req,j}$, upper and lower limits on $P_{a,j}$ 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 aggregator provides $P_{\text{mid,j}}$, a mid-ranging signal which tells the top level which $P_{\text{req,j}}$ 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 prioritise 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 hand, some ICs may 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. On a short horizon it would make sense to keep the first kind at a low energy level, and vice versa. Here we choose $E_{ref,i} = \overline{E}_i\overline{P}_i/(\overline{P}_i - \underline{P}_i)$, which corresponds to making the time to fully charging the same as the time to fully empty the energy reserve.

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

3. QP AGGREGATORS

We now turn to the problem of how the aggregator A_j distributes $P_{req,j}$ among the ICs. In (Trangbaek et al. (2010)) a quadratic programming (QP) approach was suggested. At each sample, at time t, aggregator j solves the simple optimisation problem

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

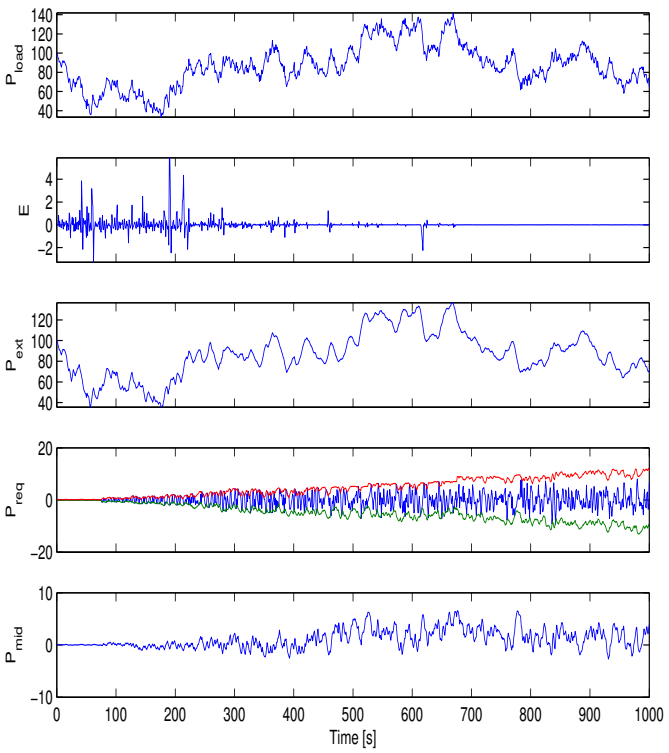

Fig. 3. Simulation example with a QP aggregator. Note how the imbalance E becomes noticeably smaller as the number of ICs increase and more flexibility becomes available.

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.

This approach has the advantage that a standard QP solver can be used.

3.1 Simulation example

A simulation of the QP scheme with one aggregator is shown in Figure 3. The controller parameters used are $T_s = 1$, $N_l = N_c = 4$, $N_p = 5$, $\beta_p = 0.1$, $\beta_r = 0.1$. The load is generated by a first order auto-regressive process with a time constant of 100 seconds. There are 20 ICs becoming available as time passes, making it possible for the aggregator to provide wider constraints on P_{req}. The ICs have $\underline{P}_i$s evenly distributed between -0.2 and -0.02, $\overline{P}_i$s evenly distributed between 0.02 and 0.2, and $\overline{E}_i$s evenly distributed between 1 and 10. 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 $\underline{P}_r(t)$ and $\overline{P}_r(t)$, 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.

The computational complexity of the QP aggregator grows with the square of number of consumers, and for large N the computational burden quickly becomes prohibitive. In the following, we suggest a method with significantly lower computational cost.

4. SORTING ALGORITHM

The sorting algorithm distributes the requested power according to which IC is furthest from the reference energy. Thus if the energy of an IC is far below the reference, and if P_{req} is positive, then this IC will be given priority to absorb power. In the following, we describe the algorithm for the case when $P_{req} > 0$. We also assume $T_s = 1$ to simplify the notation.

We again consider Aggregator A_j. First a sorted list of

$$E_{bal,i} = E_{ref,i} - E_i,\ i \in \mathscr{J}^j \tag{3}$$

is generated, highest first. Also it is computed how much power can be absorbed and at the same time bring the ICs closer to the eference,

$$W_+ = \sum_i \min(\overline{P}_i, \max(0, E_{bal,i})). \tag{4}$$

If there is excess capacity, i.e. $W_+ > P_{req}$, this can be used for bringing down the E_i that are above $E_{ref,i}$. Therefore, we also compute

$$W_- = \sum_i \max(\underline{P}_i, \min(0, E_{bal,i})). \tag{5}$$

We now initialise the power to be transferred in this way as

$$P_m := \max(0, \min(W_+ - P_{req}, -W_-)). \tag{6}$$

Thus, the total positive absorption to be found is initially

$$P_t := P_{req} + P_m. \tag{7}$$

We now go through the sorted list of ICs, starting with the highest $E_{bal,i}$.

If $E_{bal,i} > 0$ and $W_+ > P_{req}$ then we take

$$P_i = \min(E_{bal,i}, \overline{P}_i, P_t) \text{ and update } P_t := P_t - P_i. \tag{8}$$

If $E_{bal,i} > 0$ and $W_+ \leq P_{req}$ then we take

$$P_i = \min(\overline{E}_i - E_i, \overline{P}_i, P_t) \text{ and update } P_t := P_t - P_i. \tag{9}$$

If $E_{bal,i} < 0$ then we take

$$P_i = \max(E_{bal,i}, \underline{P}_i, -P_m) \text{ and update } P_m := P_m + P_i. \tag{10}$$

When all ICs have been treated we will have $P_t = P_m = 0$.

The algorithm in (3)-(10) can be seen as approximately minimising the infinity norm of E_{bal}, although simplicity has been preferred over optimality in some parts.

4.1 Simulation example

Figure 4 compares the behaviours of an aggregator using the sorting algorithm with a QP aggregator. The parameters are the same as in Section 3.1, except that there only four ICs, with

$$\overline{E}_1 = 0.1,\ \overline{E}_2 = \overline{E}_3 = 0.8,\ \overline{E}_4 = 0.2,$$
$$\overline{P}_1 = 0.1,\ \overline{P}_2 = \overline{P}_4 = 0.2,\ \overline{P}_3 = 0.02,$$
$$\underline{P}_1 = -0.1,\ \underline{P}_3 = \underline{P}_4 = -0.2,\ \underline{P}_2 = -0.02.$$

In the simulation, the load steps from 100 to 101. The resulting P_{req}, shown in the second plot, contains requests for power absorption that will help maintain a smooth P_{ext}, but is also influenced by the mid-ranging signal P_{mid} shown in the third plot. The last four plots show the energy levels in the ICs. In all the plots, solid lines are for the QP aggregator, dashed lines are for the sorting algorithm. The dotted lines in the second plot show the power limits provided by the aggregators to the

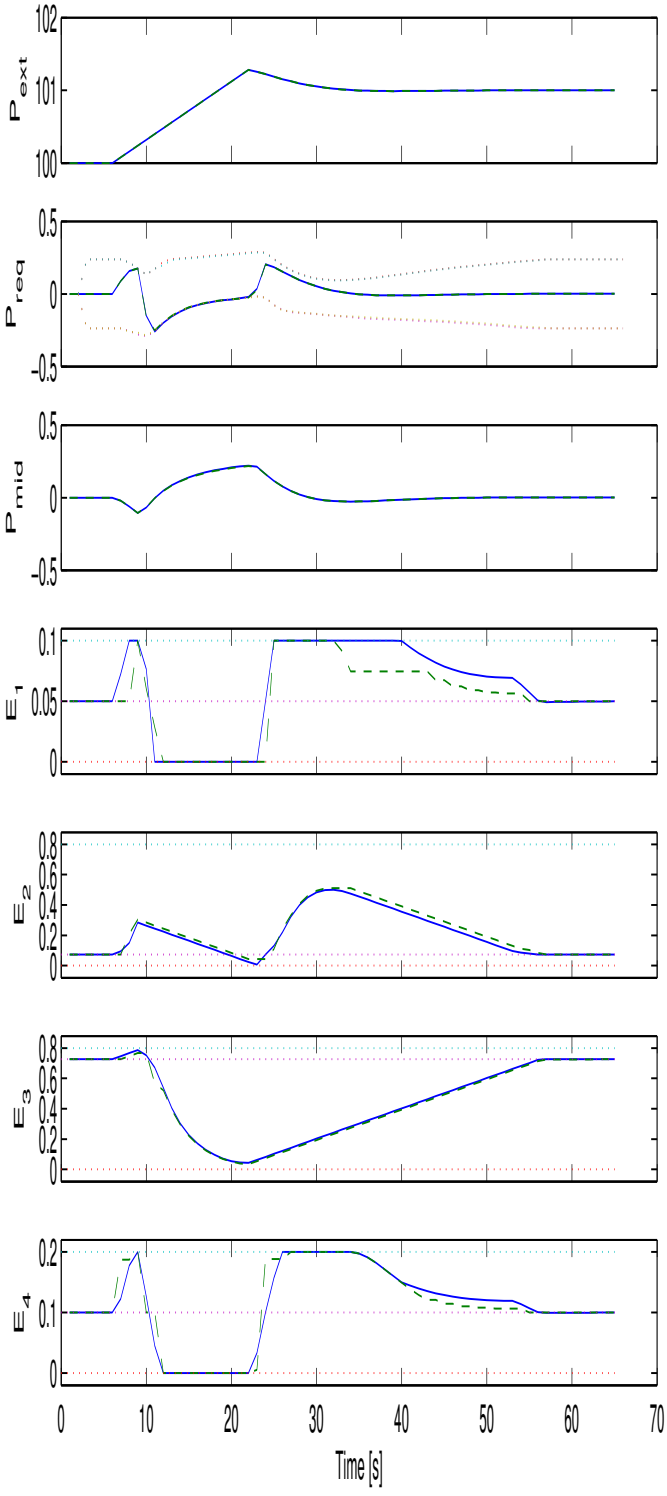

Fig. 4. Simulation example with a QP aggregator (solid) and the sorting algorithm (dashed). Note the slight difference in how the power is distributed; the sorting algorithm focuses on the extreme cases, while the QP algorithm distributes power more smoothly.

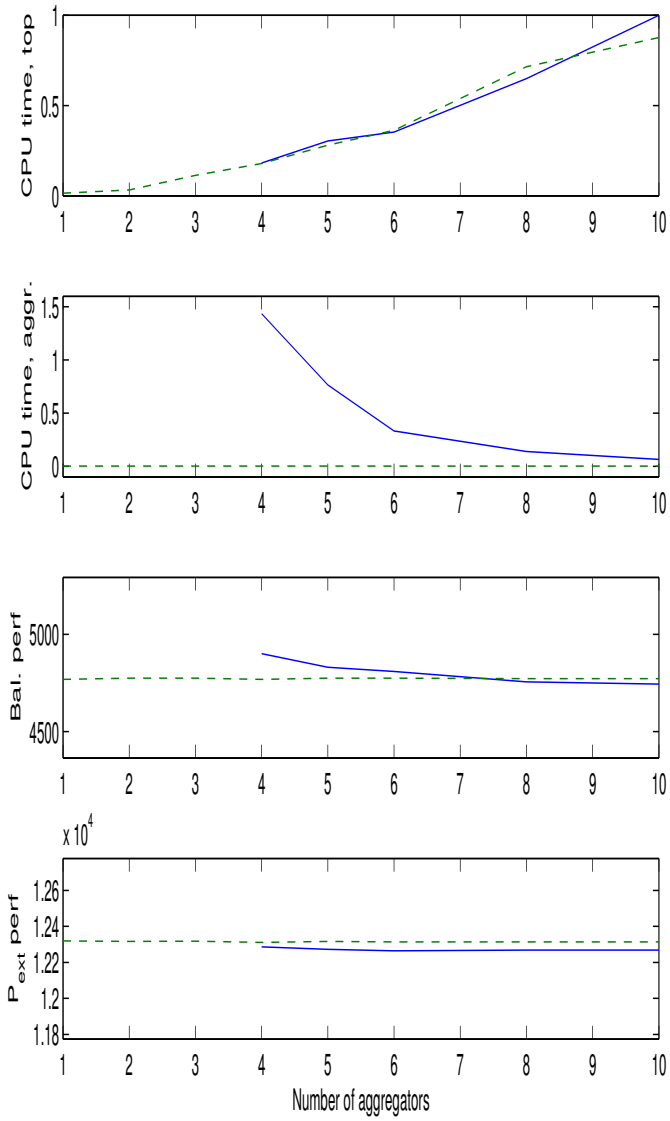

Fig. 5. CPU times for simulation with varying numbers of aggregators. Solid lines: QP. Dashed lines: Sorting algorithm.

top level. It is almost impossible to distinguish these for the two aggregator types. Indeed, the performance for the balance is almost identical, even though there are noticeable differences in how the power is distributed among the ICs.

4.2 Complexity and performance

In the following, we compare the performance and computational complexity for the two aggregation methods, by examining 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 between 0.01 and 0.13, and $\bar{P}_i$s and $-\underline{P}_i$s evenly distributed between 0.01 and 0.06. The other parameters are $T_s = 1$, $N_l = N_c = 4$, $N_p = 5$, $\beta_p = 1$, $\beta_r = 0.01$. 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 is used, only N_A is changed. Figure 5 shows the result. The

top plot shows the (scaled) time consumption of the top level controller. This grows with N_A^3 and is approximately the same for the two aggregator types. The small differences are caused by the power limits provided by the aggregator types being slightly different.

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 for the QP, and for a small number of aggregators becomes prohibitively large. On the other hand, for the sorting algorithm, the time consumption is negligible. In fact, it is possible to handle a million ICs on an ordinary PC.

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. More importantly, the performance for the two aggregator types is almost identical. In other words, the QP aggregator can be replaced by the sorting algorithm to gain considerable lower computational burden without losing performance.

5. DISCUSSION

This paper presented a novel scheme for hierarchical model predictive control (MPC) of smart grid systems. The design consists of a high level MPC controller, a second level of so-called *aggregators*, which facilitates scalability by reducing the computational and communication-related load on the high-level control, and a lower level of autonomous consumers.

The high-level MPC problem is solved using quadratic optimisation, while the aggregator level can either involve quadratic optimisation or simple sorting-based min-max solutions. We compared the computational complexity of the two approaches through simulation studies with randomised intelligent consumers.

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, we found that the QP aggregators themselves can also become too complex. Allowing for more than one aggregator alleviates some of the burden. 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.

The simple sorting algorithm, while very simple, yielded almost as good performance as the QP-based approach, which may be an indication of the fact that the top-level controller is solving almost identical optimisation problems for the two approaches. The computational burden of the sorting-based approach is much lighter, however, and allows for vastly greater numbers of intelligent consumers in the system. Our simulation studies indicate that memory consumption is the limiting factor, not computational complexity, which is of course interesting in terms of actual implementation.

So far, the setup assumes very simple intelligent consumer models. Future work involves replacing these simple models with more detailed ones, as well as identifying the minimum amount of information that needs to be transmitted from the consumers in order for the aggregator level to function properly.

ACKNOWLEDGEMENTS

This work was supported by The Danish Research Council for Technology and Production Sciences.

REFERENCES

Banakar, H., Luo, C., and Ooi, B.T. (2008). Impacts of wind power minute-to-minute variations on power system operation. *IEEE Transactions on Power Systems*, 23, 150–160.

Edlund, K., Sokoler, L.E., and Jørgensen, J.B. (2009). A primal-dual interior-point linear programming algorithm for MPC. In *Proc. of 48th IEEE Conference on Decision and Control*, 351–356. Shanghai, China.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall.

Moslehi, K. and Kumar, R. (2010). A reliability perspective of the smart grid. *IEEE Transactions on Smart Grid*, 1(1), 57–64.

Picasso, B., De Vito, D., Scattolini, R., and Colaneri, P. (2010). An MPC approach to the design of two-layer hierarchical control systems. *Automatica*, 46(5), 823–831.

Rao, C.V., Campbell, J.C., Rawlings, J.B., and Wright, S.J. (1992). Control of induction motor drives. In *International Conference on 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.

Rossiter, J.A. (2003). *Model-based predictive control*. CRC Press.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control a review. *Journal of Process Control*, 19, 723–731.

Trangbaek, K., Bendtsen, J., and Stoustrup, J. (2010). Hierarchical model predictive control for resource distribution. In *Proc. of 49th IEEE Conference on Decision and Control*.