

Distributed MPC applied to power demand side control

G. K. H. Larsen¹, J. Pons¹, S. Achterop¹ and J. M. A. Scherpen¹

Abstract—In the future, global energy balance of a smart grid system can be achieved by its agents deciding on their own power demand locally and the exchange of these decisions. In this paper, we model a network of households with washing machine programs that can be shifted in time so that the overall power demand is flattened. The network model describes how the information of power imbalance of individual agents can be exchanged in the system. Additionally, dynamics, washing machine constraints and power demand forecasts of each agent are included. Compared to existing smart grid models with hierarchical structures, our model, together with a market mechanism, achieves the power balance in the system in a completely distributed way. The market mechanism is a distributed MPC scheme based on dual decomposition and sub-gradient iterations. We provide results with a realistic power and washing machine demand pattern and we test scalability of the problem. Finally, we provide insights in the scalability of the algorithms.

I. INTRODUCTION

The term smart grid represents a vision for the future power grid, where individual users can also contribute to optimise the system by the means of demand response [1]. This means that the demand of the individual users in the network will be balanced dynamically and continuously to match the supply of electric power.

Domestic devices with flexibility to participate in demand response are typically devices such as refrigerators, since the temperature can be controlled to stay within a certain temperature range, and the washing machine program can be shifted in time. When the number of households gets large, this flexibility can help to drive the power demand in the grid towards a target value when the decisions for when to turn on is coordinated. The resulting optimisation problem that would need to be solved at a central computing centre has high combinatorial complexity. In the literature, in order to make the problem scalable, the problem is often solved in an hierarchical fashion where one decision making element is present [2].

We take a different approach where the optimisation is solved distributed. The idea is therefore to split the computation into smaller subproblems, that can be computed at household level. By exchanging prices with a few neighbours in the network and taking into account its own local information, the household makes the decision for when to consume electric power. This approach is expected to scale better than the centralised approach in a large network.

We model a network with washing machines whose switching time has to be coordinated with the overall power demand in the system. An information structure specifying the exchange information at each time step is introduced. The constraints from the washing machines have a binary nature. Model Predictive Control (MPC) is used to incorporate all operational constraints from the device.

Our suggested model can be used with a method that coordinates the agents' decisions in a completely distributed way. Such a control strategy using a price mechanism is described in [3], [4]. This strategy, based on dual decomposition, is described and applied to control a formation of vehicles [3]. By exchanging only prices, the vehicles hold the desired position. This is also an attractive idea for demand response at household level. In [5], the method is applied to a power balance problem, and in [6] distributed model predictive control (d-MPC) [7] version of the method is applied to incorporate constraints. The method successfully balances the power supply and demand in the network, even in the presence of non-convexities.

In this paper we show how distributed MPC can be used to coordinate washing machines in a local power network. Our model with the method is tested with realistic power demand patterns from typical Dutch households. Due to the binary nature of the constraints a Mixed Integer Quadratic Program is used to find the solution of the local problem, given price signals from information neighbours. We check scalability of the problem, in terms of computation time per household, and we advise an algorithm implementation with dynamic allocation of the households in an alternative case with distributed computation centres instead of computers present at each household.

The rest of the paper is organised as follows: Section II reviews the theory behind the distributed MPC algorithm. Section III develops our model for the network and the washing machine. Section IV clarifies some aspects of the practical implementation and shows the simulation results for 24 households in a network with realistic power demand patterns. Scalability of the distributed implementation is also considered. Finally, Section V discusses the result of the paper.

II. PRELIMINARIES

We aim to model and control a network of households with washing machines with demand that can be shifted in time, in a distributed Model Predictive Control (MPC) setting. We use MPC to handle constraints and prediction models, and we use the distributed control setting so that local controllers only have to take into account local information.

*This work is part of the HYCON2 project

¹Gunn Larsen, Jarno Pons, Sietse Achterop and Jacquelin Scherpen are with faculty of Mathematics and Natural Sciences., University of Groningen, 9747 AG, The Netherlands {g.k.h.larsen, s.achterop, j.m.a.scherpen} at rug.nl, j.l.pons@student.rug.nl

Here we give a quick review of a distributed MPC scheme for convex problems, as described in [7] and as is previously presented in [6]. The technique is based on dual decomposition and sub-gradient iterations.

Since the 1960s dual decomposition methods for finding the optimal control trajectories have been developed. Later these decomposition methods were also developed for dynamical large-scale optimisation problems. The original problem is replaced by several smaller subproblems. Each subproblem is isolated, except through a small interface depending on the structure of the original problem. In [4] it has been shown that dynamic price mechanisms result from the dual decomposition method for distributed optimisation of feedback systems. In [7] the method is combined with Model Predictive Control (MPC).

Consider a distributed system given by state equations

$$x(k+1) = Ax(k) + Bu(k) + w(k), \quad (1)$$

where k is the discrete time variable, $x(k) \in \mathbb{R}^n$ is the to-be-controlled vector of n users, $u(k) \in \mathbb{R}^m$ contains the m control inputs, and $w(k) \in \mathbb{R}^n$ contains the n disturbances. Information matrix A is an $n \times n$ matrix that specifies the topology and weights the information in the network, and B is an $n \times m$ input matrix. The information matrix is restricted by

$$\begin{aligned} A_{ij} &\geq 0 \\ A_{ij} &= 0 \text{ if no information flows from } i \text{ to } j, \\ \sum_j A_{ij} &:= 1, \end{aligned} \quad (2)$$

A is strongly connected.

which means that A_{ij} weights the importance of information from household j at household i , and information is conserved in the information network. We assume that each household knows the weights of all its neighbours.

Let i be a user in the network. The state, input and disturbance take its values in

$$\begin{aligned} x_i(k) &\in X_i \subseteq \mathbb{R}, \quad i = 1, \dots, n, \quad \forall k \in \mathbb{Z} \\ u_i(k) &\in U_i \subseteq \mathbb{R}, \quad i = 1, \dots, m, \quad \forall k \in \mathbb{Z} \\ w_i(k) &\in W_i \subseteq \mathbb{R}, \quad i = 1, \dots, n, \quad \forall k \in \mathbb{Z} \end{aligned} \quad (3)$$

where the sets X_i, U_i, W_i are constraining sets. The disturbances $w_i(k)$ are assumed bounded $|w_i(k)| \leq w_{max}$. Models with this boundness assumption in a dual decomposition setting are presented in [8]. There is a local cost $l_i(x_i(k), u_i(k))$ associated with each user i at every time step k , where $l_i \geq 0$ and with $l_i(0, 0) = 0$. This cost is assumed to be independent in time. The objective is to find the sequence $\{u(k)\}_{k=0}^{\infty}$ given initial values x^0 that minimise value function

$$V^\infty(x^0, u^0, \dots, u^\infty) = \sum_{k=0}^{\infty} \sum_{i=1}^n l_i(x_i(k), u_i(k)), \quad (4)$$

for the system (1) with constraints (3).

To incorporate constraints and to deal with measurements of $w(k)$ on each time step k , we formulate the problem in a MPC setting. We introduce a new discrete time variable $\tau =$

$k, \dots, N+k$ starting at time k to use over the MPC horizon N , and replace the minimisation of $V^\infty(x^0, u^0, \dots, u^\infty)$ with the minimisation of

$$V^N(x(k), \hat{u}^0, \dots, \hat{u}^N) = \sum_{\tau=k}^{N+k} \sum_{i=1}^n l_i(\hat{x}(\tau)_i, \hat{u}(\tau)_i), \quad (5)$$

which results in the *centralised MPC* problem recalculated at every k is given by

$$\min_{\hat{u}(\tau)} V^N(x(k), \hat{u}^0, \dots, \hat{u}^N), \quad (6a)$$

s.t.

$$\hat{x}(\tau)_{\tau=0} = x(k), \quad (6b)$$

$$\hat{x}(\tau+1) = A\hat{x}(\tau) + B\hat{u}(\tau) + \hat{w}(\tau), \quad \tau = 0, \dots, N, \quad (6c)$$

$$\hat{x}(\tau) \in X = X_1 \times \dots \times X_n, \quad \tau = 0, \dots, N, \quad (6d)$$

$$\hat{u}(\tau) \in U = U_1 \times \dots \times U_n, \quad \tau = 0, \dots, N, \quad (6e)$$

where $\hat{x}(\tau)$ and $\hat{w}(\tau)$ represents the predicted states and disturbances of $x(\tau)$ and $w(\tau)$. The hat notation is introduced to distinguish variables used in the finite horizon problem and variables used for the real system (1). Equation (1) is used to predict $\hat{x}(\tau)$. According to the receding horizon principle of MPC, after obtaining the finite optimal control sequence, only the first control input is implemented in (1), $u(k) = \hat{u}(\tau)_{\tau=0}$.

Next we need to obtain a distributed formulation of (6) using dual decomposition and sub-gradient iterations. The first step is to decouple state equations (6c) using dual decomposition. When $m = n$ and B is the identity matrix, we see that the right hand side in (6c) depends on neighbouring states through the information matrix A . Each user introduces a local variable $\hat{v}(\tau)$ representing the guess of expected influence from connected users. Additional equality constraints are introduced, since a guess about the neighbours' action should agree with the neighbour's reality. The system is now given by the decoupled state equations

$$\hat{x}(\tau+1) = A_D \hat{x}(\tau) + B\hat{u}(\tau) + \hat{v}(\tau) + \hat{w}(\tau) \quad (7)$$

$$\text{s.t. } \hat{v}(\tau) = A_o \hat{x}(\tau) \quad (8)$$

where $A_D = \text{diag}(A)$, $A_o = A - A_D$ and B is the identity matrix.

The constraints (8) are relaxed by introducing Lagrangian multipliers $\hat{\lambda}(\tau)$ to cost function (6a), which are interpreted as prices [3]. We obtain an *almost distributed MPC* formulation of (6). The solution is the same as for the centralised MPC under convexity assumptions, i.e. l_1, \dots, l_n are convex [7].

$$\max_{\hat{\lambda}} \min_{\hat{u}, \hat{x}, \hat{v}} \sum_{\tau=k}^{N+k} l(\hat{x}, \hat{u}) + \hat{\lambda}^T (\hat{v} - A_o \hat{x}) \quad (9)$$

$$\text{s.t. (6b), (7), (6d) and (6e) hold} \quad (10)$$

where $l(\cdot) = \sum_{i=1}^n l_i(\cdot)$, and $\hat{u}_i, \hat{x}_i, \hat{v}_i, \hat{w}_i$ and $\hat{\lambda}_i$ depend on τ . Cost function (9) can be rewritten such that the inner

minimisation over $\hat{u}_i, \hat{x}_i, \hat{v}_i$ only depends on local variables.

$$\max_{\hat{\lambda}} \sum_i \min_{\hat{u}_i, \hat{x}_i, \hat{v}_i} \sum_{\tau=k}^{N+k} l_i(\hat{x}_i, \hat{u}_i) + \hat{\lambda}_i \hat{v}_i - \left(\sum_{j \neq i} \hat{\lambda}_j A_{ji} \right) \hat{x}_i$$

s.t. (6b), (7), (6d) and (6e) hold

(11)

The inner minimisation problem is now fully distributed. Only price information from connected users are needed. However, as the problem is stated above, global coordination to obtain the right prices in the outer maximisation problem is still required. Therefore, we added the word "Almost" in the method.

In order to make the problem fully distributed, in [7] gradient iterations are included. We call the new value function in (11) $V^N(x(k), \hat{u}^0, \dots, \hat{u}^N, \hat{\lambda}^0, \dots, \hat{\lambda}^N)$. We observe that $V^N(x(k), \hat{u}^0, \dots, \hat{u}^N, \hat{\lambda}^0, \dots, \hat{\lambda}^N)$ is concave in $\hat{\lambda}$, even if the original problem is not convex [9]. The optimal price sequence can be found with the means of gradient iterations. When $\nabla_{\hat{\lambda}} V^N(x(k), \hat{u}^0, \dots, \hat{u}^N, \hat{\lambda}^0, \dots, \hat{\lambda}^N) = 0$ the constraints from (8) are met.

Prices are updated every iteration r at time-step k according to

$$\hat{\lambda}_{i,r+1}(\tau) = \hat{\lambda}_{i,r}(\tau) + \gamma_{i,r} [\hat{v}_{i,r}(\tau) - \sum_{j \neq i} A_{ij} \hat{x}_{j,r}(\tau)], \quad \tau = 0, \dots, N$$

(12)

In this way the price updates are also completely distributed, only depending on neighbouring users. Gradient iterations (12) are performed over subscripts r , and $\gamma_{i,r}$ chosen such that we converge to the optimum. A *completely distributed MPC* is obtained.

In order for the completely distributed formulation to converge to problem (11), the inner minimisation problem of (11) and gradient iterations (12) need to be solved iteratively. The convergence might need many iterations. A stopping criterion that guarantees a sub optimal bound is therefore given in [7].

If X_i, U_i are convex sets, and the cost functions $\sum_{i=1}^n l_i(x_i(k), u_i(k))$ are convex, we are guaranteed that the centralised MPC problem (6) and the decentralised MPC problem (11) provide the same solution. Thus, the completely distributed MPC gives the optimal solution, if we iterate to convergence.

III. SYSTEM DESCRIPTION

The system consists of a network of households that make decisions for when to turn on the washing machines and exchange information according to a specified topology. Here we use the setting of Section II, but the input $u_i(k)$ will be completely determined after deciding the binary on-off state of the washing machine.

A. Network Model

We model a network of $n \in \mathbb{N}$ households, where each house $i = 1, \dots, n$ has an average electric power demand $d_i(k) \in \mathbb{R}_+$ measured over each discrete time-interval k . This demand consists of a shiftable power demand $f_i(k) \in$

\mathbb{R}_+ that is a result of the decisions the household makes, and a non-shiftable part $g_i(k) \in \mathbb{R}_+$ that is modelled as an external signal. The total demand of house i at time-step k is therefore given by

$$d_i(k) = f_i(k) + g_i(k), \quad (13)$$

where $f_i(k)$ is associated with the demand from washing machines in this paper.

The variables of Eq. (1) have the following interpretations: The state $x_i(k)$ of household i is this household's information about demand which will be shared with the neighbours. It will be a combination of the household's own and its information neighbours demand, where the weights are specified by the information matrix A , see (2). See [6] for details in of a similar setting. Further, $u_i(k) = f_i(k+1) - f_i(k)$ is the change in electric power demand from the washing machine at household i , and $w_i(k) = g_i(k+1) - g_i(k)$ is the change in the rest of electric power demand at household i . In this way, when household i makes its decision for when to turn on the washing machine, it is not only taking its own demand into consideration, but also the situation in the network it is a part of. As a result, coordination of decisions in the network is enabled.

B. Network Cost

The goal is to make the power demand of the network more flat. This means that we want to move shiftable demand away from peaks in the demand pattern. The cost in (5) is given by

$$\sum_{\tau=k}^{k+N} \sum_{i=1}^n [\hat{x}_i(\tau) - a]^2, \quad (14)$$

which will be minimised when the individual demand information $\hat{x}_i(\tau)$ is close to a target value a .

C. The Washing Machine: constraints

The task is to choose when to turn on the washing machine so that the inner decoupled problems (11) are minimised together with the price iterations (12), which means determining $u_i(k)$ given the network model and the operational constraints from the device present at each house.

Once a washing program is turned on, the washing machine has to remain on until the program is finished. The washing machine can only turn on if the machine is loaded, and it has to finish the program before a specified time $T_{i,f} \in \mathbb{N}_+$. This parameter together with the number of time-steps it takes for the program to finish $T_{i,p} \in \mathbb{N}_+$ is specified when the washing machine is loaded.

The washing machine at house i is loaded $n_i \in \mathbb{N}_+$ number of times per day, according to an external signal $\eta_{i,\text{load}}(k)$ which is 1 if it is loaded at time k and zero otherwise. The washing machine can only be re-loaded if it has finished the previous program, and the arrival time is random. Binary variables $\delta_i(k)$ and $\mu_i(k)$ specify whether the washing machine is running

$$\delta_i(k) = \begin{cases} 1 & \text{if washing machine is running,} \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

and whether it is loaded

$$\mu_i(k) = \begin{cases} 1 & \text{if the washing machine is loaded,} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

which differs from $\eta_{i,\text{load}}(k)$ that is only nonzero at one time-step. Therefore, $\mu_i(k) = \mu_i(k-1) + \eta_{i,\text{load}}(k)$ unless $\delta_i(k-1) = 1 \wedge \delta_i(k) = 0$ when $\mu_i(k)$ is reset to zero. Another constraint is that the washing machine can only run if it is loaded, i.e. $\delta_i(k) := 0$ if $\mu_i = 0$.

Next we need to ensure that the washing machine completes the program $T_{i,p}$ once it has started. We introduce $t_{i,\text{on}}(k) \in \mathbb{N}_+$ to count the number of time-steps the washing machine has been running. The counter is incremented with one when $\delta_i(k) = 1$ and set to zero when $\delta_i(k) = 0$. The resulting constraints on $\delta_i(k)$ are therefore

$$\delta_i(k) = \begin{cases} 1 & \text{if } \delta_i(k-1) = 1 \wedge t_{i,\text{on}}(k-1) < T_{i,p} \\ 0 & \text{if } t_{i,\text{on}}(k-1) = T_{i,p} \end{cases} \quad (17)$$

It is also a requirement that the washing machine is finished before the given finish-time $T_{i,f}$, and for the implementation we introduce another counter $t_{i,w}(k)$ that counts the number of time-steps the machine has waited from when it was loaded. This means that it is incremented with one when $\mu_i(k) = 1$ and $\delta_i(k) = 0$, and reset to zero when $\delta_i(k) = 1$. The washing machine is forced to start running $\delta_i(k) := 1$ when the maximum waiting time is reached $(T_{i,p} - t_{i,\text{on}}(k)) \leq T_{i,f} - t_{i,w}(k)$.

Finally, the change in shiftable demand $u_i(k) = f_i(k+1) - f_i(k)$ has to get the right value when the program is running. The demand follows a pattern $e_i(k) \in \mathbb{R}_+$ when it is on. This means that

$$f_i(k) = \begin{cases} e(t_{i,\text{on}}(k)) & \text{if } \delta_i(k) = 1 \\ 0 & \text{if } \delta_i(k) = 0 \end{cases} \quad (18)$$

We will use a realistic power demand pattern $e(t_{i,\text{on}})$ for the washing machine, see Section IV.

IV. RESULT

A. Implementation

The hardware we use for the results presented in this section are four dual xeon-processor servers with a total of 32 cores.

We implement the network model in Python 2.7, and use a mixed integer quadratic program from Gurobi [10] to find the solutions of the optimisation problems. To parallelise the optimisations at each household, we used mpi4py [11], [12]. This way, each core represents one household.

The structure of the implementation is a master plus computational slaves with one household mapped to one computational slave. The computation goes in steps with complete synchronisation between steps, i.e. all computations are finished before the next time-step. See Subsection IV-C for improvements on this scheme.

The IF and AND operators in the model presented in Section III, such as (17) and (18), are included by rewriting the equations in terms of several inequality constraints.

See [13], [14] for details. In (18), auxiliary variables are introduced to indicate at which time-step in the washing program we are, so that the right value of the demand pattern $e(t_{\text{on}}(k))$ can be picked.

With the binary on-off decisions, the problem is no longer convex as the theory in Section II requires. We can expect a situation to occur when the binary decision variable oscillates between on and off. When such a situation occurs, we will fix the binary variable so that the sub-gradient iterations can converge to the optimal solution given the fixed binary variable.

B. Simulations

We use realistic power demand patterns [15] provided by the Energy research Centre of the Netherlands (ECN). The simulations use demand patterns from one evening in a November month, with a resolution of seven minutes. Each of the 20 households have a unique demand pattern, based five typical household profiles.

The information network consists of $n = 20$ households exchanging information according to

$$A = \begin{bmatrix} 0.6 & 0.2 & 0 & 0 & \dots & 0.2 \\ 0.2 & 0.6 & 0.2 & 0 & \dots & 0 \\ 0 & 0.2 & 0.6 & 0.2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0.2 & 0.6 & 0.2 \\ 0.2 & \dots & 0 & 0 & 0.2 & 0.6 \end{bmatrix}, \quad (19)$$

which means that each households has two information neighbours. They weigh themselves with a weight 0.6 and two neighbours with a weight 0.2.

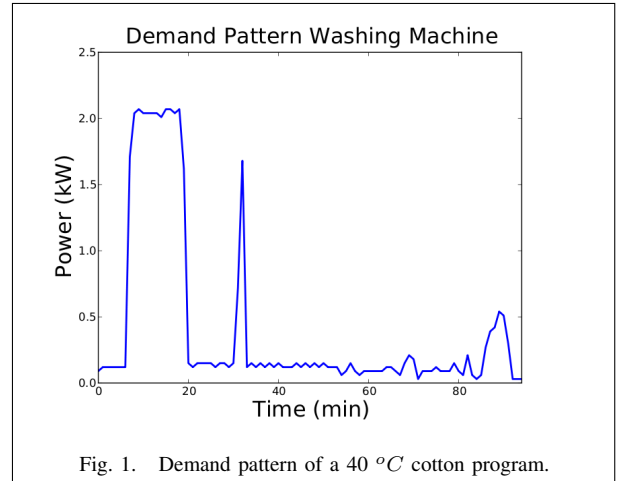


Fig. 1. Demand pattern of a 40 °C cotton program.

The power demand patterns of the washing machine is obtained from consumption patterns provided by the Flexines project [16]. Flexines measured the electricity consumption per minute for 12 different programs of the Whirlpool Texas 1400 washing machine. In the simulations presented here, a cotton wash program at 40 °C is chosen. This demand pattern $e(t_{\text{on}})$ is shown in Fig. 1.

We perform the simulations for one evening in November month, and assume that 10 of the 20 households have one wash to be done this evening, and we randomly load the washing machines during the first 2 hours and 20 minutes

of the evening. As a benchmark case, the households then immediately turn on the machine, see Fig. 2. The green solid line is the total demand in the network, the blue stippled line is the unshiftable demand and the red dotted line is the shiftable demand from the washing machines. We call this the uncontrolled case, and we will compare the cost

$$\sum_{k=0}^{T_{\text{end}}} \sum_{i=1}^n [x_i(k) - a]^2, \quad (20)$$

to the situation where the d-MPC scheme is used to determine when to start the machines. The resolution of the simulation Δk is 7 minutes, the simulation time T_{end} is 7 hours, the prediction horizon N is 4 hours and 40 minutes, and after the machine is loaded it must finish before $T_{i,f} = 4$ hours and 40 minutes. The $\gamma_{i,r}$ in (12) is 0.001 and the stopping criterion for the sub-gradient iterations is $|\hat{\lambda}_{i,r}(\tau) - \hat{\lambda}_{i,r-1}(\tau)| < 0.05$ for all $i = 1, \dots, 20$ and all $\tau = k, \dots, k + N$. The target value for each household is chosen to be approximately the average load per household over the simulation, i.e. $a = 500$ Watt. See the blue stippled line in Fig. 2.

Fig. 3 shows the situation with the d-MPC scheme presented in Section II. Compared to Fig. 2 where all washing machines turns on during the first 2 hours and 20 minutes, the load from the washing machines are now spread out over the two periods where the overall demand in the network is lowest, i.e. $k = 0 - 120$ minutes and $k = 220 - 320$ minutes. In fact, when the unshiftable demand is high, then the demand from the washing machines is low, and thus the washing machines are shifted to the moment that the unshiftable demand is somewhat lower. This is also reflected in the cost (20). In the benchmark case the cost was 248.56 kWatt², while in the d-MPC case the cost was 231.61 kWatt². All programs are finished before the specified finish-time.

Fig. 4 shows one of the households in the network with a relative high unshiftable demand, see the blue stippled line. The demand information, the green solid line, illustrates how the household share information with neighbours. From $k = 140$ minutes the demand information decreases even though the unshiftable demand and the shiftable demand of the household stays fairly constant. This information sharing with neighbours is the reason why the demand information is negative at $k = 210$ minutes, when the unshiftable demand decreases. The washing machine turns on shortly after at $k = 230$ minutes.

In the simulation presented above, the average computation-time for one time-step k was 6 seconds and the maximum computation-time for one time-step k was 15 seconds. This is well within the simulation time resolution Δk of 7 minutes. Using the centralised setup, the problem was not possible to solve using the software we used.

For scalability it is important to know that the computation time is not exploding. To check the scalability of the d-MPC optimisation, the average computation-time per time-step k , and the average number of sub-gradient iterations

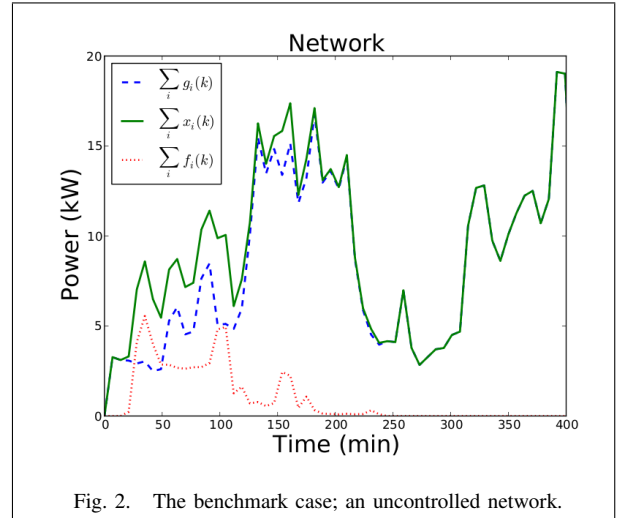


Fig. 2. The benchmark case; an uncontrolled network.

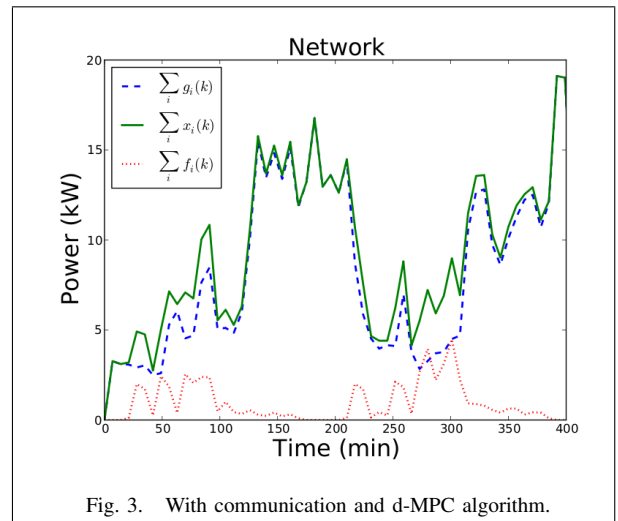


Fig. 3. With communication and d-MPC algorithm.

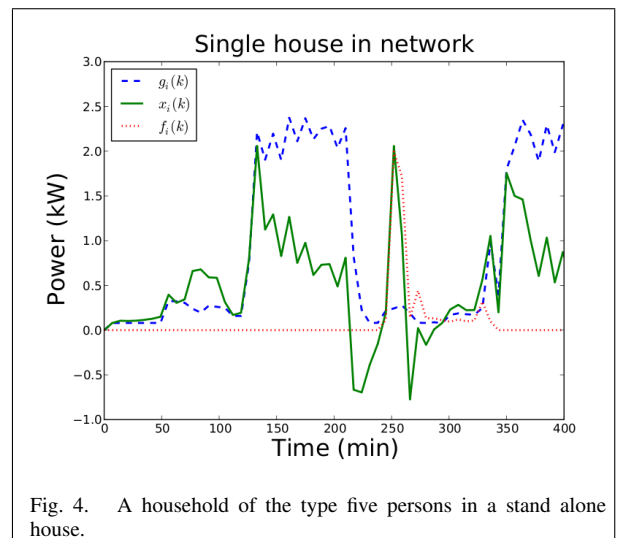


Fig. 4. A household of the type five persons in a stand alone house.

per optimisation per time-step k are given in Table I. All n households are loaded with one wash.

TABLE I
SCALABILITY OF THE D-MPC PROBLEM.

n	8	16	32
Average time per k (s)	5.7	5.9	6.4
Average # of iter	32.7	33.5	37.1

C. Scalability of the algorithms

In Subsections IV-A and IV-B, we assume that the household network and the computational network are the same. Because each household has a similar computational requirement, this solution scales quite well. However, the situation can be more flexible, with for example geographically scattered small computation centres across the network or only some households also being one computational node, whereas some others are clustered on a node. The only requirement is that the number of computational cores in total is high enough in order for a time step k in the d-MPC scheme to finish within the time resolution of the system Δk .

The computational load of each household varies a lot, i.e. it is high when the washing machine is loaded and low when there is no shiftable demand. With the implementation in the previous subsection, the computation-time for a household with a loaded washing machine was approximately 50 times longer than a household without shiftable demand.

In the case where the available processors in the network is limited, it is important to make optimal use of the available processors. We therefore need to make the mapping of households to cores dynamic. Concretely the mapping can change each cycle in the computation. This means that the computation for a household sometimes has to migrate from one core to the next. In this migration we only have to migrate the execution segment [17] of the program, because the code segment is the same everywhere.

In a large network, the relative number of nodes involved in migration is relatively small, and because the size of the execution segment is not large (here approximately 4KB), the process migration adds relatively little to the overall computational load of the system. Also, using multi-cores, the master can make the mapping so that most migration communication is local to processors.

Each household can predict when a period of high computational load arrives, since it is known when a washing machine is loaded. This has to be communicated to the master which then can change a mapping and initiate the process migrations.

V. DISCUSSION

The results in this paper show that d-MPC via dual-decomposition and sub-gradient iterations is a suitable design approach for embedding demand response in the smart grid. By exchanging price information with a few neighbours in the network, the turn-on-time of washing machines in the network was coordinated with the overall demand in

the network. Due to the distributed nature of the approach, the problem is expected to scale well as the number of households get large. This was also seen in Section IV where the scalability of the problem was tested.

The theory of Section II guarantees that the solution for the centralised and the distributed approaches are the same for convex problems. However, the non convexity of the on-off constraints change the nature of the problem. We successfully included these constraints, resulting in a network with lower costs than when no demand side control is implemented. This also accords with our earlier observations when using the approach to embed distributed generation in the power network [6].

A further study with focus on a larger variation of devices suitable for demand response present at the households is suggested, and it has to be connected to local production devices. Another interesting topic, is further development of the scalability considerations for the algorithms as discussed in Subsection IV-C.

REFERENCES

- [1] A. Khattak, S. Mahmud, and G. Khan, "The power to deliver: Trends in smart grid solutions," *Power and Energy Magazine, IEEE*, vol. 10, no. 4, pp. 56–64, Jul 2012.
- [2] A. Molderink, V. Bakker, M. Bosman, J. Hurink, and G. Smit, "Management and control of domestic smart grid technology," *Smart Grid, IEEE Transactions on*, vol. 1, no. 2, pp. 109–119, Sep 2010.
- [3] A. Rantzer, "On prize mechanisms in linear quadratic team theory," in *Proc. 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, Dec 2007, pp. 1112–1116.
- [4] —, "Dynamic dual decomposition for distributed control," in *Proceedings of American Control Conference*, St. Louis, USA, Jun 2009, pp. 884–888.
- [5] G. K. H. Larsen, J. M. A. Scherpen, N. D. van Foreest, and E. Doornbos, "Distributed control in a network of households with μ chp," in *Proc. 18th IFAC World Congress*, Milan, Italy, Aug 2011, pp. 5320–5325.
- [6] G. Larsen, N. van Foreest, and J. Scherpen, "Distributed control of the power supply-demand balance," *Smart Grid, IEEE Transactions on*, vol. PP, no. 99, pp. 1–9, 2013.
- [7] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *Proceedings of the 49th IEEE Conference on Decision and Control 2010*, Atlanta, GA, USA, 2010, pp. 7272–7277.
- [8] A. Grancharova and T. Johansen, "Distributed quasi-nonlinear model predictive control by dual decomposition," in *Proceedings of the 18th IFAC World Congress*, Milan, Italy, Aug 2011, pp. 1429–1434.
- [9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
- [10] *Gurobi Optimizer 5.0 Reference Manual*, 2012.
- [11] <http://mpi4py.scipy.org/>, 2012.
- [12] <http://www.openmpi.org/>, 2012.
- [13] J. L. Pons, "Distributed control of electricity demand - the application of distributed model predictive control on a network of washing machines," Master's thesis, University of Groningen, 2012.
- [14] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar 1999.
- [15] J. Paauw, B. Roossien, M. Aries, and O. Guerra Santin, "Energy pattern generator; understanding the effect of user behaviour on energy systems," *First European Conference Energy Efficiency and Behaviour*, 2009.
- [16] <http://flexines.org/>, 2011.
- [17] A. Tanenbaum and M. van Steen, *Distributed Systems 2nd edition*. Pearson, 2007.