

# Distributed Asynchronous Greedy Control of Large Networks of Thermostatically Controlled Loads for Electric Demand Response

M. Kaheni, A. Pilloni, G. Serra Ruda, E. Usai, and M. Franceschelli\*

**Abstract**—This letter illustrates two multi-agent greedy demand-side response control schemes for networks of Thermostatically Controlled Loads. The objective is to provide simple but effective local control actions such that the overall power consumption tracks an aggregated desired profile. Compared with the existing literature the novelties are twofold. Since model-free, our schemes possess certain robustness features to the model deterioration and exogenous disturbances. Since greedy, they are of easy implementation also on cheap development boards which do not support optimization software, moreover because asynchronous do not require any network-wide synchronization event. Specifically, Algorithm 1 is very simple but it is applicable only on K-regular communication topologies. Such prerequisite is then removed in Algorithm 2 by including within its instruction list a dynamic consensus protocol to estimate the mean network power consumption. Performance analysis and numerical simulations confirm the effectiveness of the schemes.

**Index Terms**—Demand Response, Greedy Control, Multi-agent Systems, Thermostatically Controlled Loads.

## I. INTRODUCTION

THE widespread installation of *Renewable Energy Sources* (RES), and the fact they do not contribute to the power system inertia could make the future power systems and *microgrids* particularly sensitive to frequency variations. Among the options to overcome this problem, in some countries *National Electricity Distributors*, which are responsible for maintaining the *Quality-of-Service* (QoS) and the *frequency nadir* no lower than 49.2 Hz (in Europe), announced a new market for *Energy Service Providers* (ESPs), aimed to respond within *one second* to frequency variations [1] (*ten times faster* than conventional frequency control). Among other options such

The authors gratefully acknowledge the financial support of the Fondazione di Sardegna with grants: “Formal Methods and Technologies for the Future of Energy Systems (CUP:F72F2000035000)”, “IQSS-Information Quality aware and Secure Sensor networks for smart cities” (CUP:F75F21001400007), “ARGOSAT-Microsatellite cluster for the observation of optical transients in Astronomy (CUP:F7419001070007)”; and the Sardinian Regional Government for “POR SARDEGNA FSE 2014-2020-Asse III, Azione:10.5.12, “Avviso di chiamata per il finanziamento di Progetti di ricerca Anno 2017”.

M. Kaheni, A. Pilloni, G. Serra Ruda, E. Usai, and M. Franceschelli are with the Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Italy.

E-mail: mojtaba.kaheni@unica.it, alessandro.pilloni@unica.it, g.serraruda@studenti.unica.it, elio.usai@unica.it, mauro.franceschelli@unica.it.

\*Mauro Franceschelli is corresponding author.

as energy storage systems [2], studies confirmed the *Demand-side Response* (DSR) is the most cost-effective solution for the deployments and optimal management of the power flows within a microgrid. Although statistics may vary across countries, a significant part of the residential energy demand is due to the *Thermostatically Controlled Loads* (TCLs) (e.g. water heaters, radiators, air conditioners, etc.). Thus, their coordinated control could be crucial for DSR success. Studies focused on TCL-DSR programs in the United Kingdom [3], California [4], and Sardinia [5] authenticate this claim.

## A. Literature review

TCL-DSR can aim for various objectives within a microgrid. For instance, strategies for provisioning *frequency response* are [3], [5]. Specifically, TCLs monitor the microgrid frequency, then respond to deviations by decreasing (if the frequency is lower than a threshold) or increasing (if higher) their consumption by means of model-based considerations. *Peak load shaving* programs which aim to desynchronise the TCLs' consumption are [6]–[8]. Then, [9]–[12] study the so-called *load-following* programs, where the TCLs are coordinated such that the overall network demand tracks a given time-varying desired profile provided by the ESP. This letter aims at this control objective.

To control the consumption of a TCL one can act on the thermostat, or modify its temperature setpoint. Nonetheless, the *ENTSO-E* states the customer's right to set the temperature of his/her own TCL has to be reserved [13]. Thus, the DSR cannot override the TCL temperature ranges, and of course, it cannot create disservices to the users. As a consequence, strategies aimed to modify the temperature (or *hysteresis*) setpoints, thus providing an additional degree of control, such as [7], [11], are no more considered legit.

TCL-DSR programs can be seen as large-scale constrained *mixed-integer* optimization problems [8]. Moreover, to keep computationally tractable the optimizations, it is common to approximate the dynamics of TCLs as LTI systems [8], [12], [14] thanks to which is predicted when the TCL will/would be switched on/off. Those predictions are however strongly influenced either by parameters deterioration or disturbances, such as water withdrawal, season changes, etc.

To confirm this claim, it is worth to mention [15], which found that both first and second-order LTI models are significantly inaccurate in describing the TCL behaviour. It follows

that model-free approaches are desirable because inherently robust to model uncertainties.

From a control perspective, most of the recently published load-following DSR schemes are, semi- ( [10], [11]), or fully-centralized ( [16], [17]). Thus, within the control architecture there exists a central coordinator, or a set of coordinator *agents*, gathering information and making choices to be forwarded to all, or to a subset of TCLs participating in the DSR program. These architectures carry however cyber-security vulnerabilities since misbehavior on the coordinator(s) threatens the whole system performance. *Multi-agent system* (MAS) architectures where instead the decision-making is distributed among peer autonomous agents can instead significantly improve the process resilience and flexibility and the risk of cascading failures.

### B. Statement of contributions

This letter investigates two distributed model-free TCL-DSR programs. In the literature, the most relevant and related approach (because classified to be *model-free*) is [18]. Therein, it is proposed a priority-based optimization method, aimed to smooth out the power demand by controlling TCLs. To achieve the task the algorithm exploits information on the TCL temperatures and “nominal” *duty-cycle*, whereas it dispenses the explicit use of the temperature model. Then, each TCL autonomously takes actions in accordance with its own score and the result of the optimization. In the following the main differences between [18] and our proposals are given:

- The objective of [18] is the peak shaving, while our goal is the load-following which is a more challenging task.
- The algorithm in [18], in its distributed-setting implementation, needs each agent broadcasts its own priority score to all the participating loads, thus needing a complete communication graph.
- Although classified to be model-free, [18] needs the TCL duty-cycle for solving optimizations. However, this function depends on the TCL temperature parameters and unknown disturbances (water withdrawal, season changes, aging effects, etc). This clearly makes [18], also sensitive to performance degradation since the effective duty cycle varies during operations. Moreover, it will also require periodic identification procedures as for classical model-based schemes. In contrast, our proposal does not need any physical TCL parameters.

Thus, compared to the known literature, the main contributions in this letter can be summarized as follows:

- Our algorithms are model-free and even the TCL duty cycle is not required for solving local optimizations.
- Our algorithms are fully distributed, do not require any central coordinator, and do not suffer from single-point failures. Specifically, Algorithm 1 requires a *K-regular graph* (even *directed*), whereas Algorithm 2 a common *undirected* topology.
- In our algorithms the decision-making is taken by each TCL autonomously (by exploiting only the neighbours thermostat states), and asynchronously (no need for network-wide periodic synchronization events).

- Our algorithms are greedy and thus well suited to be implemented on cheap development boards which do not support any optimization solver.
- Our algorithms are compliant with the ENTSO-E guidelines, which is instead not the case of [7], [11], [12].

## II. PROBLEM FORMULATION

Consider a MAS consisting of  $\mathcal{V} = \{1, 2, \dots, n\}$  TCLs. Each TCL is equipped with certain sensing, actuation and communication capabilities. The communication topology at time  $t_k \geq 0$  is encoded by a digraph  $\mathcal{G}(t_k) = (\mathcal{V}, \mathcal{E}(t_k))$  where  $\mathcal{E}(t_k) \subseteq \mathcal{V} \times \mathcal{V}$ . Specifically,  $(i, j) \in \mathcal{E}(t_k)$  if TCL  $i$  is enabled to receive information from  $j$ , whereas  $\mathcal{N}_i^{\text{in}}(t_k) : \{j \in \mathcal{V} \setminus \{i\} : (i, j) \in \mathcal{E}(t_k)\}$  and  $\mathcal{N}_i^{\text{out}}(t_k) : \{j \in \mathcal{V} \setminus \{i\} : (j, i) \in \mathcal{E}(t_k)\}$  are the in- and out-neighbourhood of TCL  $i$ . Then, let  $P_i \in \mathbb{R}^+$  be the rated power absorbed by the TCL when its thermostat  $h_i(t_k) \in \{0, 1\}$  is high, then

$$p_i(t_k) = P_i \cdot h_i(t_k), \quad (1)$$

approximates its absorbed power at time  $t_k \geq 0$ , with  $k \in \mathbb{N}_0$ , while dispensing the use of power sensors. For water heaters and radiators, the thermostat state is generally updated in accordance with the *reverse hysteric control*

$$h_i(t_{k+1}) := \begin{cases} 0 & \text{if } T_i(t_k) > T_i^{\text{max}}, \\ 1 & \text{if } T_i(t_k) < T_i^{\text{min}}, \\ h_i(t_k) & \text{otherwise,} \end{cases} \quad (2a)$$

$$(2b)$$

$$(2c)$$

where  $T_i(t_k) \in \mathbb{R}$  is the temperature of TCL  $i$ , and  $T_i^{\text{max}} \geq T_i^{\text{min}} > 0$  denote the hysteresis window. On the other hand, in refrigerators or cold flow conditioners the high and low conditions in (2a)-(2b) are reversed. Finally, each TCL is assumed measuring if  $T_i(t_k) \in [T_i^{\text{min}}, T_i^{\text{max}}]$  as well as the Boolean thermostat status  $h_i(t_k)$ .

Let us now define the overall (*total*) instantaneous absorbed power associated with the network of TCLs as

$$P^t(t_k) = \sum_{i=1}^n p_i(t_k), \quad (3)$$

and let the corresponding normalized value, computed with respect to the total nominal installed power, be as next

$$\zeta^t(t_k) = \frac{P^t(t_k)}{\sum_{j=1}^n P_j}. \quad (4)$$

Let us further define  $P^d(t_k) \in \mathbb{R}^+$  as the *desired* overall instantaneous power demand broadcasted, for instance, by the ESP to the MAS, and let

$$\zeta^d(t_k) = \frac{P^d(t_k)}{\sum_{j=1}^n P_j} \quad (5)$$

be, as for (4), the corresponding normalized desired power demand, then **our objective** is to locally control the status of  $h_i(t_k)$  in (2c), in a such a way  $P^t(t_k)$  tracks  $P^d(t_k)$ , or equivalently demonstrate that  $\zeta^t(t_k)$  tracks  $\zeta^d(t_k)$ .

*Remark 1:* Override the thermostat is common in TCL-DSR. This is justified by the fact most of the existing policies consider *smart* TCLs. Specifically, among [3]- [11], only [8]

considers instead off-the-shelf TCLs, and thus there, the DSR is implemented by using networked smart sockets. ■

### III. MAIN RESULTS

In the following, Two model-free distributed and asynchronous greedy load-following programs operating under different and detailed networking assumptions are presented.

#### A. Load-following DSR for $K$ -regular TCL networks

Robustness to node or communication failure in MAS largely depends on the graph *connectivity*. Although connectivity can be improved by adding edges or by increasing the neighbours of each node, since an edge stands for some communications/physical links, adding edges increases the infrastructure costs. A family of well-connected yet sparse graphs, is that of  $K$ -regular graphs (connected *digraphs* where each node has the in- and out-degree equal to  $K$ ). Their disadvantage is that they are generally difficult to be computed, especially in distributed fashion. On the other hand, there exist algorithms which allow networks to self-organize their graphs into a  $K$ -regular graph, e.g. [19]. Thus, in this subsection is assumed the following.

*Assumption 1:* The graph  $\mathcal{G}(t_k)$  is  $K$ -regular, namely  $|\mathcal{N}_i^{\text{in}}| = |\mathcal{N}_i^{\text{out}}| = K, \forall i \in \mathcal{V}, K \in \mathbb{N}, t_k \geq 0$ . □

Finally, because of each TCL cannot access the network power demand  $P^t(t_k)$ , as well as is corresponding normalized consumption  $\zeta^t(t_k)$  in (4), we further define the following, *local*, in-neighbourhood normalized consumption

$$\zeta_i^\ell(t_k) = \frac{p_i^\ell(t_k)}{P_i^\ell(t_k)} = \frac{\sum_{j \in \{i\} \cup \mathcal{N}_i^{\text{in}}} P_j(t_k)}{\sum_{j \in \{i\} \cup \mathcal{N}_i^{\text{in}}} P_j}. \quad (6)$$

We are now in position to present our Algorithm 1.

---

#### Algorithm 1 (Implemented within each TCL $i \in \mathcal{V}$ )

---

- 1: **Initialize**  $h_i(t_k), T_i(t_k), P_i, \zeta_d(t_k)$  and  $\forall t_k$  **do**
  - 2: **Gather**  $P_j$  and  $h_j(t_k)$  for all  $j \in \mathcal{N}_i^{\text{in}}(t_k)$
  - 3: **Update**  $P_i^\ell(t_k) = \sum_{i \cup \mathcal{N}_i^{\text{in}}(t_k)} P_j^\ell(t_k)$  and  $p_i^\ell(t_k) = \sum_{i \cup \mathcal{N}_i^{\text{in}}(t_k)} P_j h_j(t_k)$
  - 4: **Let**  $\zeta_i^\ell(t_k) = p_i^\ell(t_k) / P_i^\ell(t_k)$  **as in** (6)
  - 5: **Compute**  $\zeta_i^{\ell'}(t_k) = (p_i^\ell(t_k) + P_i(1 - 2h_i(t_k))) / P_i^\ell(t_k)$
  - 6: **If**  $T_i(t_k) \in [T_i^{\text{min}}(t_k), T_i^{\text{max}}(t_k)]$  **AND**  $|\zeta_i^{\ell'}(t_k) - \zeta^d(t_k)| < |\zeta_i^\ell(t_k) - \zeta^d(t_k)|$  **then**  $h_i(t_k^+) = 1 - h_i(t_k)$
  - 7: **else**  $h_i(t_k^+) = h_i(t_k)$
  - 8: **endif**
- 

Algorithm 1 is **model-free**, and **asynchronous**, in the sense that each TCL calls Algorithm 1 with respect to its own clock, and no network-wide synchronization events are required. Moreover, during its execution, the amount of memory storage required at node  $i$  is independent from  $n$ , and equal to  $6 + K$  variables, where  $K = |\mathcal{N}_i^{\text{in}}|$ , thus it is scalable to large  $n$ . Last, if  $\mathcal{G}(t_k)$  is sufficiently connected ( $K$  is large), then the Mean Square Error (MSE) defined as

$$\text{MSE} = \frac{1}{T_f} \sum_{t_k=0}^{T_f} (P^d(t_k) - P^t(t_k))^2, \quad (7)$$

is reduced. The numerical results of Fig. 3 show a  $K \geq 40$  minimizes such error.

Further note Algorithm 1 controls the thermostat status  $h_i(t_k)$  in such a way the normalized in-neighbourhood consumption (6) is kept close to the desired setpoint (5). Moreover, instruction “6” guarantees at each iteration the tracking error either decreases, or at most does not increase, namely  $|\zeta_i^\ell(t_k^+) - \zeta^d(t_k)| \leq |\zeta_i^\ell(t_k) - \zeta^d(t_k)|$ , where  $t_k^+ > t_k$  represents the instant of time after the algorithm execution at time  $t_k \geq 0$ . Let us now show that if  $\zeta_i^\ell(t_k)$  tracks  $\zeta^d(t_k) \forall i$ , then  $P^t(t_k)$  is tracking  $P^d(t_k)$ .

*Theorem 1:* Let Assumption 1 be satisfied. If  $\zeta_i^\ell(t_k) = \zeta^d(t_k)$  holds  $\forall i \in \mathcal{V}$ , then  $P^t(t_k) = P^d(t_k)$ .

*Proof:* Firstly note that, independently from the considered digraph  $\mathcal{G}(t_k)$ , the following identity holds

$$\sum_{i=1}^n \left( \sum_{j \in \mathcal{N}_i^{\text{in}}} p_j(t_k) \right) = \sum_{i=1}^n |\mathcal{N}_i^{\text{out}}| p_i(t_k). \quad (8)$$

From (8) and (3), and by simple manipulations one may further obtain the following one-to-one relationship holds

$$\sum_{i=1}^n \left( \sum_{j \in \mathcal{N}_i^{\text{in}}} p_j(t_k) \right) - \sum_{i=1}^n (|\mathcal{N}_i^{\text{out}}| - 1) p_i(t_k) = P^t(t_k). \quad (9)$$

Then, by exploiting Assumption 1, we have  $|\mathcal{N}_i^{\text{out}}| = K, \forall i \in \mathcal{V}$ . Thus, (9) is rewritten as next

$$P^t(t_k) = \sum_{i=1}^n \left( \sum_{j \in \mathcal{N}_i^{\text{in}}} p_j(t_k) \right) - (K - 1) \sum_{i=1}^n p_i(t_k). \quad (10)$$

Consider now (5) and (6). Since in the theorem's statement is assumed  $\zeta_i^\ell(t_k) = \zeta^d(t_k)$ , then (10) is rewritten as

$$P^t(t_k) = \sum_{i=1}^n \left( \zeta^d(t_k) \sum_{j \in \mathcal{N}_i^{\text{in}}} P_j \right) - (K - 1) \sum_{i=1}^n p_i(t_k). \quad (11)$$

Then, by simple manipulations it further results

$$P^t(t_k) = \zeta^d(t_k) \sum_{i=1}^n \sum_{j \in \mathcal{N}_i^{\text{in}}} P_j - (K - 1) P^t(t_k). \quad (12)$$

Now, by applying the same argument we used for  $\sum_{i=1}^n p_i(t_k)$  in (9), to  $\sum_{i=1}^n \sum_{j \in \mathcal{N}_i^{\text{in}}} P_j$ , it further results

$$\sum_{i=1}^n \sum_{j \in \mathcal{N}_i^{\text{in}}} P_j = \sum_{i=1}^n P_i + (K - 1) \sum_{i=1}^n P_i = K \sum_{i=1}^n P_i. \quad (13)$$

Finally, by replacing (13) into (12), yields

$$P^t(t_k) = \zeta^d(t_k) K \sum_{i=1}^n P_i - (K - 1) P^t(t_k), \quad (14)$$

which after simplifications can be rewritten as next

$$P^t(t_k) = \zeta^d(t_k) \sum_{i=1}^n P_i. \quad (15)$$

Then, from (15) and (5), one can further note that  $P^t(t_k) = P^d(t_k)$ , thus concluding this proof. ■

*Remark 2:* Since algorithms for forming regular graphs are

generally sub-optimal, thus it may happen few nodes may have either  $K + 1$  or  $K + 2$  neighbours, see e.g. [19]. If so, Assumption 1 would no longer be satisfied, and therefore the identity used in (9) is no longer valid. Nevertheless, if  $K$  is big enough such that  $K + 1 \approx K$ , then also  $P^t(t_k) \approx P^d(t_k)$ . Thus, the results of Theorem 1 do not change significantly. ■

*Remark 3:* Consider sparse  $K$ -regular graphs makes Algorithm 1 resilient to failures and malicious attacks. Specifically, let  $K'$  be the number of compromised agents sending wrong/false data. It is reasonable to assume  $K' < K$ . Thus, these  $K'$  agents will influence only a few in-neighbourhood, each for a weight that is  $1/K$ . Thus, the majority of agents will instead continue to work correctly. ■

### B. Load-following DSR for undirected mesh TCL networks

Building large  $K$ -regular networks from an arbitrarily mesh network is problematic in practice, not only for the reasons discussed in Section III-A. In fact, it may also happen that due to the different population densities some nodes would have a larger degree than others. Thus, to achieve the load-following goal on mesh networks, in this subsection it is assumed the following.

*Assumption 2:* The interaction graph  $\mathcal{G}(t_k)$  is connected and undirected, and the network size  $n \in \mathbb{N}$  is known. □

*Remark 4:* In our load-following DSR, the knowledge of  $n$  is not restrictive since TCLs participating in the DSR should, in general, “login” to a web-server to receive  $P^d(t_k)$  by the ESP. Thus, at least the ESP knows  $n$  and can send this information to the agents while preserving the privacy of their power consumption. In addition, even if  $n$  is not available for any reason, distributed network size estimation algorithms, such as [20], can be used to estimate  $n$ . ■

To consider mesh networks an extra cost in terms of information each agent has to deliver and compute must be paid. Specifically, to achieve our goal we need to provide each TCL with an estimation  $P_e^t(t_k)$  of (3). To do that, and similarly with [8], we could thus embed within our DSR program a *dynamic average consensus protocol* to estimate  $P^t(t_k)$ . Due to its simplicity in implementation and tuning, the fact it is randomized, and that the convergence is only required the graph is connected and undirected, here it is considered the dynamic consensus protocol proposed in [21]. It is worth mentioning that the iterations of the dynamic average consensus algorithm need to occur at a greater frequency than those of Algorithm 2 to ensure a time-scale separation between the two algorithms. Specifically, let  $z_i^r$  be the  $r$ -th state of a  $m$ -stages cascaded consensus filter of the form

$$\begin{aligned} z_i^1(\kappa + 1) &= \delta p_i(t_k) + (1 - \delta) z_i^1(\kappa) - \epsilon \sum_{j \in \mathcal{N}_i^{\text{in}}} (z_i^1(\kappa) - z_j^1(\kappa)) \\ z_i^2(\kappa + 1) &= \delta z_i^1(\kappa) + (1 - \delta) z_i^2(\kappa) - \epsilon \sum_{j \in \mathcal{N}_i^{\text{in}}} (z_i^2(\kappa) - z_j^2(\kappa)) \\ &\vdots \\ z_i^m(\kappa + 1) &= \delta z_i^{m-1}(\kappa) + (1 - \delta) z_i^m(\kappa) - \epsilon \sum_{j \in \mathcal{N}_i^{\text{in}}} (z_i^m(\kappa) - z_j^m(\kappa)) \end{aligned} \quad (16)$$

where  $p_i(t_k) = P_i \cdot h_i(t_k)$  in (1) is the filter's input,  $z_i^m(\kappa)$  is its output at the  $\kappa$ -th iteration. Finally, let the next tuning

rules be satisfied

$$\epsilon < (2K_{\max})^{-1}, \quad \delta < 1 - \epsilon K_{\max}, \quad (17)$$

where  $K_{\max} = \max_i \{k_i\} \leq N$  is the maximum node's degree, then  $z_i^m(k)$  will converges to the mean total consumption of the network, namely to  $P^t(t_k)/n$ . Notice that the distributed dynamic consensus on average algorithm only estimates averaged quantities, thus we need  $n$  to reconstruct  $P^t(t_k)$ .

We are now in position to present our Algorithm 2.

---

#### Algorithm 2 (Implemented within each TCL $i \in \mathcal{V}$ )

---

- 1: **Initialize**  $h_i(t_k), P_i, T_i(t_k), P^d(t_k), n$  **and**  $\forall t_k$  **do**
  - 2: **Estimate by means of** (16)  $P_e^t(t_k) = n \cdot z_i^m(\kappa)$
  - 3: **If**  $T_i(t_k) \in [T_i^{\min}(t_k), T_i^{\max}(t_k)]$   
**AND**  $P_e^t(t_k) > P^d(t_k) + \gamma$  **then**  $h_i(t_k^+) = 0$
  - 4: **else if**  $T_i(t_k) \in [T_i^{\min}(t_k), T_i^{\max}(t_k)]$   
**AND**  $P_e^t(t_k) < P^d(t_k) - \gamma$  **then**  $h_i(t_k^+) = 1$
  - 5: **else**  $h_i(t_k^+) = h_i(t_k)$
  - 6: **endif**
- 

Notice that in Algorithm 2, each TCL firstly executes the dynamic consensus (16), which implementation, following [21], can be randomized, and where at each iteration  $\kappa$ , a set of random edges is selected from its neighbourhood. Then, by exploiting  $n$ ,  $P^t(t_k)$  is estimated. Finally, as for Algorithm 1, the thermostat state  $h_i(t_k)$  is controlled in such a way  $P^t(t_k)$  is kept close to the desired demand  $P^d(t_k)$ . Finally,  $\gamma > 0$  is a threshold related to the total power consumption estimation accuracy, formalized in Theorem 2.

*Remark 5:* Since the proposed algorithms may override the thermostat status if only  $T_i \in [T_i^{\min}(t_k), T_i^{\max}(t_k)]$ , namely in (2c), then if  $P^d(t_k)$  at a given time is exceptionally low while many TCLs have a low temperature then a tracking error, possibly large, could be observed. This is a consequence that the priority, in accordance with [13], is given to the QoS of the TCL owners. Nevertheless, as time passes, TCLs coordinate itself to track  $P^d(t_k)$ . ■

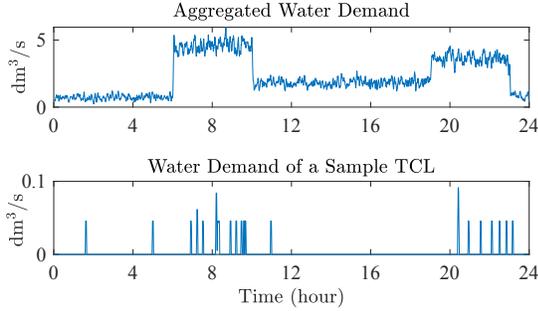
*Remark 6:* It is worth mentioning that there may exist several TCLs planning solutions that meet the temperature constraint,  $T_i \in [T_i^{\min}(t_k), T_i^{\max}(t_k)]$ , and the solution obtained by our proposed algorithms may not be the optimal one with respect to the MSE index defined in (7). However, we have relaxed many restrictions such as the computational burden and the need for a mixed-integer optimization solver, the requirement of a central coordinator, and the use of any model information such as the temperature model dynamics or the duty-cycle knowledge. ■

We now show that Algorithm 2 does not cause any increment of the load-following tracking error during its executions.

*Theorem 2:* Consider a network consist of  $n$  TCLs. Let Assumption 2 hold and let  $P^t(t_k)$  and  $P^t(t_k^+)$  be the overall absorbed power by the TCLs at time  $t_k$  and  $t_k^+$  (immediately after an iteration of Algorithm 2), respectively. Let  $z_i^m(k)$  be the estimation of  $P^t(t_k)/n$  by dynamic consensus protocol in (16) and  $P^d(t_k)$  represent the desired load profile at time  $t_k$ . If  $|z_i^m(k) - P^t(t_k)/n| < \gamma/2n \forall i \in \mathcal{V}$ , and  $\gamma \geq \max_{i \in \mathcal{V}} \{P_i\}$ ,

**Table 1** Water heater model parameters

$\rho$	Water density	1	[kg/dm <sup>3</sup> ]
$c_p$	Water specific heat	4186	[J/(C°kg)]
$R_i$	Thermal resistance	0.0488	[m <sup>2</sup> C°/W]
$S_i$	Tank surface	0.536	[m <sup>2</sup> ]
$V_i$	Tank volume	100	[dm <sup>3</sup> ]
$P_i$	Heater power	1500	[W]



**Fig. 1.** Top: Aggregated daily hot water demand for the considered network of water heaters. Bottom: Daily water demand for a generic water heater.

then

$$|P^t(t_k^+) - P^d(t_k)| \leq |P^t(t_k) - P^d(t_k)|.$$

*Proof:* First notice that as discussed in [21], the dynamic consensus protocol (16) can track the average TCLs power consumption with the desired accuracy by tuning design parameters. Thus, recalling that  $\gamma$  in Algorithm 2 is *lower bounded*, since the accuracy of the dynamic consensus protocol in [21] is *upper bounded*, it is always possible to find a large enough  $\gamma$  that satisfies  $|z_i^m(k) - P^t(t_k)/n| < \gamma/2n$ .

According to the definition of  $P_e^t(t_k)$  in Step 2 of Algorithm 2,  $|z_i^m(k) - P^t(t_k)/n| < \gamma/2n$ , leads to  $|P_e^t(t_k) - P^t(t_k)| < \gamma/2$ , from which we observe that the term  $[P_e^t(t_k) > P^d(t_k) + \gamma]$  in Step 3 is logically true only if

$$P^t(t_k) > P^d(t_k) + \gamma/2. \quad (18)$$

Since the result of applying a control action in Step 3 can be only  $h_i(t_k^+) = 0$ , therefore only two events are feasible.

**Case 1** ( $h_i(t_k) = h_i(t_k^+) = 0$ ): In this case  $P^t(t_k^+) = P^t(t_k)$ , therefore  $|P^t(t_k^+) - P^d(t_k)| = |P^t(t_k) - P^d(t_k)|$  and the desired result is achieved.

**Case 2** ( $h_i(t_k) = 1$  and  $h_i(t_k^+) = 0$ ): In this case  $P^t(t_k^+) + P_i = P^t(t_k)$ . Thus, noticing that  $\gamma = \max_{i \in \mathcal{V}} \{P_i\}$ ,  $P^t(t_k^+) + \gamma \geq P^t(t_k)$ , and from (18), one finally obtains

$$P^t(t_k^+) \geq P^d(t_k) - \gamma/2. \quad (19)$$

Noticing that  $P^t(t_k^+) < P^t(t_k)$ , (18) and (19) yield to  $|P^t(t_k^+) - P^d(t_k)| < |P^t(t_k) - P^d(t_k)|$ . By a similar argument, one can reach to same result for Step 4. The proof is thus concluded. ■

*Remark 7:* Although privacy issues go beyond the scope of this work, it is worth mentioning both our algorithms keep private the  $T_i(t_k)$  measurements and share only  $h_i(t_k)$  and  $P_i$  to their neighbourhood for the decision making. Moreover, note that in the presence of malicious interference, due to the local temperature constraints within each algorithm, in

the worst case, it may happen that the tracking fails, but no disservices to the TCL's users will be observed. ■

#### IV. NUMERICAL EXAMPLE

To evaluate the proposed algorithms a network of  $n = 1000$  perturbed TCLs is considered. Specifically, each TCL is modelled as a water heater, which temperature  $T_i(t_k)$ , following [22], is updated accordingly with the next discrete-time dynamic model

$$T_i(t_{k+1}) = A_k T_i(t_k) + B_k \left( \alpha_i T_i^r + \beta_i(t_k) T_i^{\text{in}} + \gamma_i h_i(t_k) \right),$$

$$A_k = e^{-(\alpha_i + \beta_i(t_k))\Delta t}, B_k = \frac{(1 - e^{-(\alpha_i + \beta_i(t_k))\Delta t})}{\alpha_i + \beta_i(t_k)}, \quad (20)$$

$$\alpha_i = \frac{S_i}{\rho c_p R_i V_i}, \quad \beta_i(t_k) = \frac{w_i(t_k)}{V_i}, \quad \gamma_i = \frac{c_p P_i}{\rho V_i}, \quad (21)$$

where  $h_i(t_k)$  plays as the DSR control input, that is adjusted to keep  $T_i(t_k)$  between  $T_i^{\text{min}} = 50^\circ\text{C}$  and  $T_i^{\text{max}} = 60^\circ\text{C}$ . Then,  $T_i^r = 20^\circ\text{C}$  is the *room* temperature,  $w_i(t_k)$  in (21) is an unknown disturbance aimed to model the cold water refill process within the water heater after a water withdrawal, and finally  $T_i^{\text{in}} = 15^\circ\text{C}$  is the *inlet* cold water temperature. The remaining model parameters are listed in Table III-B. During the simulation, the TCL's temperatures are randomly initialized such that  $T_i(0) \in [T_i^{\text{min}}, T_i^{\text{max}}]$ , and only the 50% of them, at the start-up, have  $h_i(0) = 1$ . It is also worth mentioning that  $w_i(t_k)$  is an ad-hoc designed stochastic process aimed to model higher hot water demand at the peak hours, as depicted in Fig. 1. Finally, calls to the proposed randomized algorithms are modelled as a Poisson point process with a mean rate of 10 calls per second.

Regarding the developed tests, since Algorithm 1 requires a  $K$ -regular network, whereas Algorithm 2 an undirected topology, for the sake of comparison, undirected randomly generated  $K$ -regular graphs are considered. Specifically, *Case 1* consider  $K = 100$  and *Case 2* a  $K = 10$ . Finally, the Algorithm 2 parameters are set as next

$$\begin{aligned} \text{Case 1 } (K = 100) : & \quad m = 3, \quad \epsilon = 0.0045, \quad \delta = 0.1 \\ \text{Case 2 } (K = 10) : & \quad m = 3, \quad \epsilon = 0.045, \quad \delta = 0.1 \end{aligned}$$

whereas the mean number of interactions  $\kappa$  of the dynamic consensus algorithm given in (16) is 10 per second.

The network response to the case studies is shown in Fig. 2. Specifically, the left-subplots consider a sinusoidal desired demand of the form  $P^d(t_k) = 0.75 - 0.1 \sin(2\pi t_k/\tau)$  with period a  $\tau$  of 12 hours (purple signal), whereas in the right-subplots a constant set-point,  $P^d = 0.75$  MW is taken into account. The blue signals denote the network consumption in the uncontrolled case, whereas the red and yellow lines are the consumption under the control of Algorithm 1 and Algorithm 2. As it can be seen, both algorithms provide good tracking despite the constraint on  $T_i(t_k)$ , which must be kept within the range  $[T_i^{\text{min}}, T_i^{\text{max}}]$ . In fact, the tracking is missed only at the end of the first peak, because the expected consumption was very high, and half of the TCLs came from a "off" condition, at the start-up. Finally, note that the tracking accuracy of Algorithm 1 reduces as the network connectivity decreases. This is fairly reasonable because of the very low

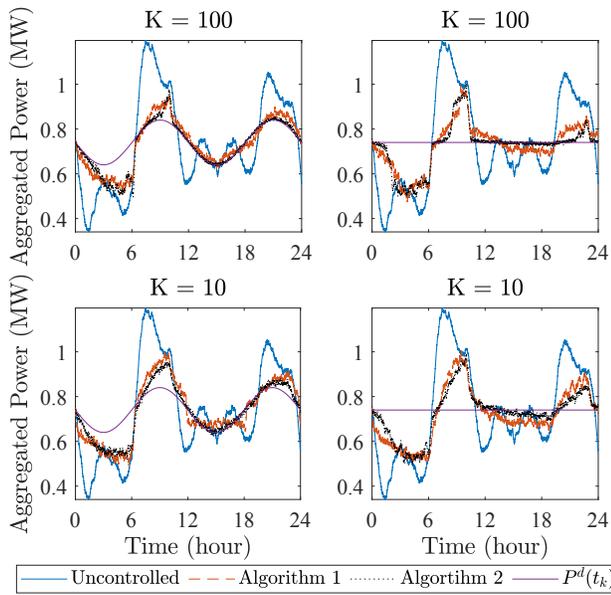


Fig. 2. Network response under different DSR policies, and different  $K$ -regular communication networks (one per row, resp., 100 and 10), and different desired demand profiles:  $P^d(t_k) = 0.75 - 0.1 \sin(2\pi t_k/\tau)$  MW (left column) and  $P^d = 0.75$  MW (right column).

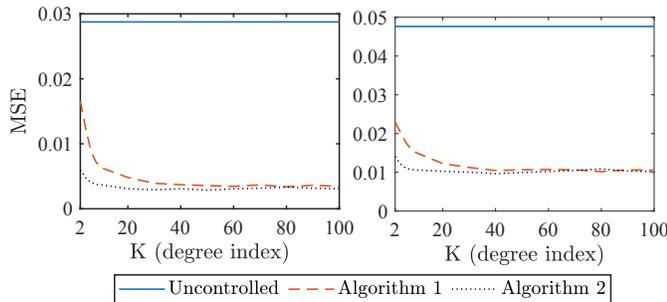


Fig. 3. MSE (7) comparison for different  $K$ -regular graphs, and different desired demand profiles:  $P^d(t_k) = 0.75 - 0.1 \sin(2\pi t_k/\tau)$  MW (left column) and  $P^d = 0.75$  MW (right column).

number of executions of the algorithm we selected and the small neighbourhood.

In Fig. 3 is shown in how our steady state accuracy vary as  $K$  increase, for test which length is of  $T_f = 24$  hours These numerical results show a  $K \geq 40$  minimizes MSE (7).

## V. CONCLUSION

The major advantage of the proposed DSR programs is that both are model-free, thus they are inherently robust to model uncertainties and unknown disturbances. Both are asynchronous, thus dispensing the need to implement periodic network-wide synchronization events. Finally, it is worth mention both are simple and well-suited to be implemented even on hardware with limited capabilities. A natural extension of this study is to reduce the need for some local measurements, such as TCLs' temperatures, as well as the formal characterization of the steady-state accuracy of our algorithms by means of probabilistic arguments. Finally, also the study of the

practical concerns related to the presence of malicious agents and the data privacy issues appear of particular interest.

## REFERENCES

- [1] National Grid, "Enhanced frequency response," [Online]. Available: [www2.nationalgrid.com/Enhanced-Frequency-Response.aspx](http://www2.nationalgrid.com/Enhanced-Frequency-Response.aspx), 2015.
- [2] M. Kaheni, E. Usai, and M. Franceschelli, "A distributed optimization and control framework for a network of constraint coupled residential BESSs," *IEEE Int. Conf. Autom. Sci. and Eng.*, pp. 2202–2207, 2021.
- [3] V. Trovato, I. M. Sanz, B. Chaudhuri, and G. Strbac, "Advanced control of thermostatic loads for rapid frequency response in great britain," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2106–2117, 2017.
- [4] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Potentials and economics of residential thermal loads providing regulation reserve," *Energy Policy*, vol. 79, pp. 115–126, 2015.
- [5] F. Conte, S. Massucco, F. Silvestro, E. Ciapessoni, and D. Cirio, "Stochastic modelling of aggregated thermal loads for impact analysis of demand side frequency regulation in the case of sardinia in 2020," *Int. Journal of Electrical Power & Energy Syst.*, vol. 93, pp. 291–307, 2017.
- [6] M. Franceschelli, A. Gasparri, and A. Pisano, "Coordination of electric thermal systems for distributed demand-side management: A gossip-based cooperative approach," *European Control Conf.*, pp. 623–630, 2016.
- [7] D. Angeli and P.-A. Kountouriotis, "A stochastic approach to "dynamic-demand" refrigerator control," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 3, pp. 581–592, 2012.
- [8] M. Franceschelli, A. Pilloni, and A. Gasparri, "Multi-agent coordination of thermostatically controlled loads by smart power sockets for electric demand side management," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 731–743, 2021.
- [9] S. H. Tindemans, V. Trovato, and G. Strbac, "Decentralized control of thermostatic loads for flexible demand response," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1685–1700, 2015.
- [10] J. Hu, J. Cao, T. Yong, J. M. Guerrero, M. Z. Q. Chen, and Y. Li, "Demand response load following of source and load systems," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1586–1598, 2017.
- [11] M. Ghanavati and A. Chakravarthy, "Demand-side energy management by use of a design-then-approximate controller for aggregated thermostatic loads," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1439–1448, 2018.
- [12] L. A. D. Espinosa, A. Khurram, and M. Almassalkhi, "Reference-tracking control policies for packetized coordination of heterogeneous DER populations," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2427–2443, 2021.
- [13] ENTSO-E, *Demand Connection Code*, 2015. [Online]. Available: [Online]. Available: [www.entsoe.eu/major-projects/network-code-development/demand-connection](http://www.entsoe.eu/major-projects/network-code-development/demand-connection)
- [14] S. P. Nandanoori, I. Chakraborty, T. Ramachandran, and S. Kundu, "Identification and validation of virtual battery model for heterogeneous devices," in *IEEE Power Energy Soc. Gen. Meet.m*, 1–5, 2019.
- [15] Z. Xu, R. Diao, S. Lu, J. Lian, and Y. Zhang, "Modeling of electric water heaters for demand response: A baseline PDE model," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2203–2210, 2014.
- [16] S. Xiang, L. Chang, B. Cao, Y. He, and C. Zhang, "A novel domestic electric water heater control method," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3246–3256, 2020.
- [17] L. Yao and H.-R. Lu, "A two-way direct control of central air-conditioning load via the internet," *IEEE Trans. Power Deliv.*, vol. 24, no. 1, pp. 240–248, 2009.
- [18] K. Kircher, A. Aderibole, L. K. Norford, and S. Leeb, "Distributed peak shaving for small aggregations of cyclic loads," *IEEE Trans. Power Deliv.*, 2022.
- [19] A. Y. Yazıcıoğlu, M. Egerstedt, and J. S. Shamma, "Formation of robust multi-agent networks through self-organizing random regular graphs," *IEEE Trans. Netw. Sci.*, vol. 2, no. 4, pp. 139–151, 2015.
- [20] D. Deplano, M. Franceschelli, and A. Giua, "Dynamic min and max consensus and size estimation of anonymous multi-agent networks," *IEEE Trans. Autom. Control*, 2021.
- [21] M. Franceschelli and A. Gasparri, "Multi-stage discrete time and randomized dynamic average consensus," *Automatica*, vol. 99, pp. 69–81, 2019.
- [22] L. Paull, H. Li, and L. Chang, "A novel domestic electric water heater model for a multi-objective demand side management program," *Electric Power Syst. Research*, vol. 80, no. 12, pp. 1446–1451, 2010.