

A Two-Layer Near-Optimal Strategy for Substation Constraint Management via Home Batteries

Igor Melatti, Federico Mari, Toni Mancini, Milan Prodanovic, *Member, IEEE*, and Enrico Tronci.

Abstract—Within electrical distribution networks, substation constraints management requires that aggregated power demand from residential users is kept within suitable bounds. Efficiency of substation constraints management can be measured as the reduction of constraints violations w.r.t. unmanaged demand. Home batteries hold the promise of enabling efficient and user-oblivious substation constraints management. Centralized control of home batteries would achieve optimal efficiency. However, it is hardly acceptable by users, since service providers (e.g., utilities or aggregators) would directly control batteries at user premises. Unfortunately, devising efficient hierarchical control strategies, thus overcoming the above problem, is far from easy. We present a novel two-layer control strategy for home batteries that avoids direct control of home devices by the service provider and at the same time yields near-optimal substation constraints management efficiency. Our simulation results on field data from 62 households in Denmark show that the substation constraints management efficiency achieved with our approach is at least 82% of the one obtained with a theoretical optimal centralized strategy.

NOMENCLATURE

s EDN substation
 T_s set of time slots for substation s
 U set of houses (users)
 t time slot (element of T)
 u house (element of U)
 T_u set of time slots for house u ; note that, for all notation depending on an house index u , if the house is understood, u is not shown
 $T_{u,P}$ set of time slots in T_u in which the EV is plugged-in on house u
 τ_l duration (in minutes) of time slots in T_s

Authors Melatti, Mancini and Tronci are with the Sapienza University of Rome, via Salaria 113, 00198 Rome, Italy. Author Mari is with the University of Rome “Foro Italico”, Viale del Foro Italico, 00135 Rome, Italy. Author Prodanovic is with Electrical Systems Unit of IMDEA Energy Institute, Avda. Ramón de la Sagra, Móstoles Technology Park, Madrid 28935, Spain. This work was partially supported by: Research programme S2018/EMT-4366 PROMINT-CAM from Madrid Government; Italian Ministry of University and Research under grant “Dipartimenti di eccellenza 2018–2022” of the Department of Computer Science of Sapienza University of Rome; EC FP7 project SmarHG; INdAM “GNCS Project 2019”.

τ_s time (in minutes) between two optimisation decisions on substation
 $H_s = \frac{\tau_s}{\tau_l}$ optimisation horizon for substation
 τ time (in minutes) between two optimisation decisions on houses
 $Q_{u,E}, m_{u,E}, M_{u,E}, \alpha_E, \beta_E$ maximum capacity (in kWh), minimum and maximum power rate (in kW), charge and discharge efficiency of ESS in u
 $Q_{u,P}, m_{u,P}, M_{u,P}, \alpha_P, \beta_P$ maximum capacity (in kWh), minimum and maximum power rate (in kW), charge and discharge efficiency of battery EV in u
 C_u^{low}, C_u^{high} minimum and maximum power demand (in kW) from energy contract of u
 H_i, H, H_δ initial optimisation horizon, current optimization horizon and optimization horizon changing step for computation in houses
 ζ deadline (in minutes) to complete a single optimisation computation in houses
 $P_s^{low}(t), P_s^{high}(t)$ lower and upper desired power bounds (in kW) for substation s in t
 $\tilde{Q}_{u,E}, \tilde{Q}_{u,P}$ current SoC (in kWh) of ESS and EV in u
 $\mathcal{D}_{u,P}$ deadline (in minutes) for EV complete recharge in u
 $d_u(t)$ forecasted power demand (in kW) of u in t
 $b_{u,E}(t)$ SoC (in kWh) of ESS in u at t
 $a_{u,E}(t)$ charge or discharge action (in kW) for the ESS in u at t
 $P_u^{low}(t), P_u^{high}(t)$ lower and upper power limits (in kW) for u in t
 $\Delta^{low}(t), \Delta^{high}(t)$ aggregated power (in kW) which exceeds substation lower and upper bounds in t
 $e_u(t)$ resulting power demand (in kW) in u at t
 $a_{u,E}^{ch}(t), a_{u,E}^{dis}(t)$ charge and discharge actions (in kW) for the ESS in u at t
 $a_{u,P}^{ch}(t), a_{u,P}^{dis}(t)$ charge and discharge actions (in kW) for the EV in u at t
 $z_u(t)$ overall power demand exceeding power limits (in kW) for u at t
 $y_{u,E}(t), y_{u,P}(t)$ binary variables, true if ESS and EV of u are charged in t and false otherwise
 $y_u^{low}(t), y_u^{high}(t), y_u^{in}(t)$ binary variables, true if overall power $e_u(t)$ is greater than lower limit $P_u^{low}(t)$, less than upper limit $P_u^{high}(t)$, inside lower and upper limits (resp.) in u at t

I. INTRODUCTION

In an Electrical Distribution Network (EDN), many electrical substations provide electricity to the residential users connected to such substations. On each substation s , one of the main goals for a Distribution System Operator (DSO) is Substation Constraints Management (SCM), that is, enforcing suitable desired lower and upper bounds on the aggregated power demand resulting from the houses connected to s . In fact, SCM enables savings for the DSO, e.g., in substations maintenance and energy peak production [1]. In the context of smart grids, computational services may be used to enforce *efficient* and *effective* SCM. Namely, efficiency is measured as the reduction of bounds violations w.r.t. the unmanaged aggregated power demand, i.e., the complement to 1 of the time average of the ratio between managed w.r.t. unmanaged demand outside bounds. On the other hand, SCM is effective if it minimizes user discomfort and is technically viable. The main obstacles for efficient and effective computational services for SCM are the following: 1) using Autonomous Demand Response (ADR), i.e., relying on residential users to autonomously respond to price incentives, is often ineffective, as users tend to ignore price signals [2]; 2) using Direct Load Control (DLC), i.e., active power curtailment and reactive power control, is ineffective as well, as it may lead to a loss of useful energy [3]. A promising way of achieving efficient and effective SCM is to install low-cost batteries at each user premises, and then automatically controlling them. In this way, bounds violations can be reduced by automatically shifting user demand, which also minimizes user discomfort. Although centralized control of such batteries would achieve optimal efficiency, it faces the following main obstacles: 1) reliability of communication lines, as every few minutes a command for each home battery must be sent to each user (we note that, typically, home electricity mains can transmit, but not receive); 2) above all, for security and privacy reasons, many users would not accept such a centralized solution, as a service provider (e.g., a utility or an aggregator) is demanded of controlling the storage at the user premises. Thus, we have to rely on hierarchical control strategies for home batteries.

Related work. Many single-layer, as well as hierarchical methodologies, have been proposed in the context of smart grids, with different goals. As for single-layers methodologies, in [2], [4] individualized Inclining Block Rate (IBR) and Time of Use (ToU) price policies for residential houses connected to a substation s are investigated, to perform peak shaving. In [5] a data analytical ADR management scheme for residential load is proposed to reduce the peak load demand. In [6], a privacy-aware stochastic multiobjective optimization framework that considers the objectives of both consumers and utility companies in an ADR scheme. Unfortunately, users tend to ignore price signals [2]. This motivates the goal of this paper, i.e., automatically shifting user demand.

A single-layer approach involving DLC of Heating, Ventilation and Air Conditioning (HVAC) appliances is proposed in [7]. Unfortunately, demanding DSO of controlling devices at user premises is hardly accepted by users. Furthermore, here we focus on actuating home batteries. In [8], a methodology is

presented to allow a storage aggregator to invest and operate a central physical storage unit, by virtualizing it into separable virtual capacities and selling it to users. In [9], an intelligent multi-microgrid energy management method is proposed based on artificial intelligence techniques, to protect user privacy. In our setting, we only use the information from user mains, which is already available to DSOs. In [10], a methodology for optimal residential battery operation in a single house is proposed, to minimize electricity costs. Finally, the approaches in [11], [12] focus on the scheduling of Electric Vehicles (EVs) only. However, in our setting, we are interested in SCM, which is not addressed in [9], [8], [10], [11], [12].

The methodologies described above mainly rely on the charge/discharge of home batteries to perform power demand shifting. Many other methodologies (see, e.g., [13], [14] and citations thereof) have also been proposed which rely on scheduling appliances usage, to be either automatically or manually applied. However, such approaches require either modern smart appliances, which may not be available in many houses, or rely on users manually applying the scheduling, which is ineffective [2]. In our setting, we focus on batteries as they allow both 1) more widespread applicability, as it is simpler, especially in non-modern houses, to install a home battery than many smart appliances, and 2) to always rely on a completely automatic approach.

As for hierarchical methodologies, in [15] a hierarchical distributed Model Predictive Control (MPC) approach is presented to solve the energy management problem in the multi-time frame and multilayer optimization strategy. In [16] a hierarchical day-ahead Demand Side Management (DSM) model is proposed, where renewable energy sources are integrated. In [17] a hierarchical approach is presented for distributed voltage optimization in high-voltage and medium-voltage EDNs. In [18] a two-layer distributed cooperative control method for islanded networked microgrid systems is described. In [19] an optimal multiobjective control methodology is discussed for power flow regulation and compensation of reactive power and unbalance in AC microgrids. In [20] a bilevel optimization framework is presented to minimize energy cost for commercial building HVAC systems. In [21] a distributed energy management strategy for the optimal operation of microgrids is described. In [22] a distributed consensus-based approach is proposed to solve the grid welfare problem by deriving an real-time pricing scheme that facilitates an automated ADR. problem by deriving a real-time pricing scheme that facilitates an automated ADR. In [23] a hierarchical MPC of smart grid systems is described to balance demand and supply. Such methodologies cannot be applied to our setting, as they do not address the problem of constraining the aggregated demand of residential users within given desired bounds. In [24], a two-layer control framework is proposed to perform peak shaving (i.e., keeping the aggregated power demand below a given upper threshold). However, in our setting, we are interested in acting on domestic batteries, which allows a more widespread use, while [24] focuses on (smart) HVAC only.

Finally, “adaptive” MPC often refers to techniques able to automatically adjust, at run-time, the model parameters [25], e.g., the weights of some constraints (see [26] and citations

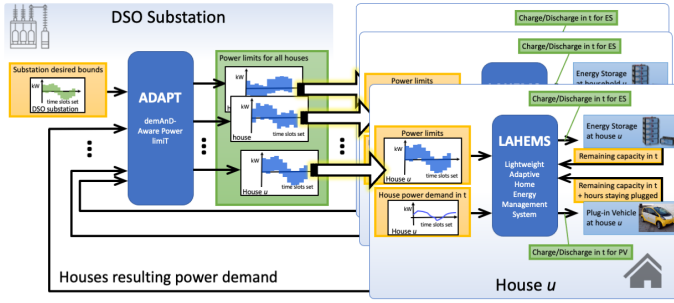


Fig. 1: The proposed architecture

thereof). In [27], a lightweight MPC scheme able to adjust its prediction horizon has been presented and evaluated on a simple industrial process plant. However, in [27] horizon changes are driven by the need of tuning the reference trajectory of the model. Instead, in our work, we vary the horizon so as to keep the house inside given power bounds.

Summing up, to achieve efficient and effective SCM, we need a framework that avoids centralized solutions (typically not accepted by users for privacy and security reasons) and rely neither on user autonomously changing their habits nor in (possibly expensive) smart appliances.

Main Contributions. To overcome the obstructions described above, in this paper we propose a novel hierarchical two-layer computational service for efficient and effective SCM. We call such a service Demand-Aware Network Constraint mAnager (DANCA, see Figure 1). In the following, we list the main contributions of our approach.

1) DANCA encloses two services, each running at different levels of the EDN (substations and houses) and with different periodicity (orders of days w.r.t. orders of minutes).

- The first layer is the DemAnD-Aware Power limit (ADAPT) service (based on [2]), which is executed independently for each EDN substation s . The ADAPT goal is to maintain the aggregated power demand of s within the desired range given in input, by computing individualized and time-dependent lower and upper bounds on the power demand of each house connected to s . The duration of such bounds must be long enough to allow users to actually shift their power demand (one day in our experiments).

- The second layer is the Lightweight Adaptive Home Energy Management System (LAHEMS) service, which must be run independently on each residential user u . The LAHEMS goal is to maintain the demand of u within the power bounds decided for u by ADAPT. Namely, LAHEMS acts as a Home Energy Management System (HEMS) which is able to control the charge and discharge of home batteries, thus shifting the demand of u to stay inside the given power bounds. To this aim, LAHEMS must compute actions on home batteries with a sufficiently short periodicity (5 minutes in our experiments), to catch up with variations in the demand of u .

2) DANCA may be either directly employed by the DSO itself or offered by a Demand Side Response Aggregator (DSRA). In this latter case, the DSO provides the DSRA with

the desired bounds on the aggregated demand of s and will pay the DSRA so that violations on such bounds are minimized. In the following, we will refer to the entity running DANCA as *DANCA provider*.

3) ADAPT only requires in input the bounds on the substation (always provided by the DSO) and the residential user power demand (already provided by the electricity main in each house), thus it is executed at the DANCA provider premises, possibly using powerful computing devices. On the other hand, LAHEMS is responsible to actuate charge/discharge of home batteries with real-time requirements, thus it is executed at each user premises. Communication between the two services takes place when ADAPT sends to LAHEMS the lower and upper bounds for the given user power demand: this happens only once a day without real-time constraints, thus it can rely, e.g., on a typical home Internet connection.

4) Running LAHEMS at each user premises entails that an inexpensive and small microcomputer with limited computational resources, i.e., small RAM and low CPU frequency (in our experiments, we used a Raspberry Pi), can be used.

5) Both ADAPT and LAHEMS are based on the MPC methodology [28], [29]. That is, with the given periodicity (1 day and 5 minutes, respectively), ADAPT and LAHEMS solve a suitable optimization problem which, depending on forecasts for the user power demand, minimizes the power outside the given bounds. From the solution to the optimization problem, ADAPT extracts the bounds for each user, while LAHEMS extracts the charge/discharge actions for batteries. The main parameter for the MPC methodology is the *receding horizon* used for the optimization problem, i.e., how many hours in the future must be considered. While ADAPT receding horizon is typically one day (as it is standard in the day-ahead energy market), for real-time-constrained LAHEMS it should be experimentally estimated in an initialization phase, which may be costly. To this aim, for each user, the detailed power demand on a past period (e.g., one year) is needed, which may be unavailable. Furthermore, such initialization could not catch up with modifications in user power demand habits, which would diminish LAHEMS effectiveness. LAHEMS solves such a problem by employing an *adaptive* algorithm, which automatically adjusts, at run-time, the receding horizon. Moreover, LAHEMS succeeds in doing this without violating real-time requirements. To the best of our knowledge, this is the first time that such an algorithm is presented.

Experimental Results. We experimentally evaluate the efficiency and effectiveness of DANCA using data collected from sensors in 62 Danish households connected to the same substation during the SmartHG project [30]. As a result: 1) DANCA is able to achieve a 50% efficiency (i.e., reduction of substation bounds violations w.r.t. the unmanaged demand). This is a near-optimal solution, as a *theoretical* optimal centralized approach on the same scenario would achieve 61% efficiency, i.e., our solution is 82% as effective as the theoretical optimal one. We remark that the results obtained in the centralized version of our approach cannot be actually achieved in our setting for the previously explained reasons

(i.e., lack of reliability of communication lines and users security and privacy reasons). 2) LAHEMS can be run on a Raspberry Pi, meeting the required hard-real-time deadlines. As for ADAPT, it may easily be run by a desktop computer.

II. PROBLEM FORMULATION AND SYSTEM ARCHITECTURE

In our setting, a set of residential houses U are connected to the same substation s . The DSO D is able to compute, basing on documentation and recorded power demand data, *desired power bounds* for the substation $P_s^{low}(t), P_s^{high}(t)$ (in kW) for suitable time slots $t \in T_s$ (in our experiments, each t lasts one hour and all $t \in T_s$ refer to the next day). Let $d_u(t)$ be the power requested to the grid by house $u \in U$ in time slot $t \in T_s$, and let $d(t) = \sum_{u \in U} d_u(t)$ be the *aggregated* power demand in t (in kW). Furthermore, let $\Delta(d, t) = \Delta^{low}(d, t) + \Delta^{high}(d, t) = \max\{P_s^{low}(t) - d(t), 0\} + \max\{d(t) - P_s^{high}(t), 0\}$ be the power (in kW) outside $P_s^{low}(t), P_s^{high}(t)$, if any, when the aggregated demand is d . If D is able to keep the overall aggregated power outside the desired substation bounds $\Delta(d) = \sum_{t \in T_s} \Delta(d, t)$ as low as possible, then it will save in substation maintenance and energy peak production [1]. Given this, we want to devise a software framework to *shift* power demand d_u of each $u \in U$, so as to obtain a power demand $e = \sum_{u \in U} e_u = \sum_{u \in U} d_u + a_u$ s.t. the aggregated power outside bounds $\Delta(e)$ is minimized over a long-enough period (e.g., one year). We want such a framework to have the following properties: 1) It must be *completely automatic*, by shifting each household power demand *without* involving residential user direct actions, to minimize user discomfort. Note that demand shifting must not entail power curtailment [3]. 2) It must be *easily applicable* to most houses, with as low hardware installations as possible. 3) It must be *technically viable*. That is, the aggregated demand which results from the power shifts must reduce the peaks outside the substation desired bounds. Moreover, we have to show that real-time requirements arising from the hardware-software interaction are met. Figure 2 (left) shows an example of our problem formulation, in which we only consider the upper bound on the substation by using input and output selected from the most demanding day in our experiments (see Section IV). Note that shifts may be positive (e.g., from 0 AM to 5 AM, where the resulting demand green curve is above the historical demand blue curve) as well as negative (e.g., from 8 AM to 11 AM). In the day depicted in Figure 2, the reduction is about 50%, measured as $1 - \frac{\Delta(e)}{\Delta(d)}$.

In our proposed framework, each house $u \in U$ is provided with a battery (and related circuitry/inverters). This allows us to implement demand shifts via charge/discharge commands to such batteries (see, e.g., [10]). This also allows us to easily apply our methodology to most houses, given the widespread availability and low costs of modern home batteries (*applicability*). As we want a fully automatic framework, we need software computing such charge/discharge commands. However, this cannot be done at the DSO premises, as having the utility directly acting on batteries at user premises would not be acceptable for users. Furthermore, commands for batteries

need to be computed at a high rate (e.g., every 5 minutes) and to be reliably delivered. To this aim, using Internet links may entail delays or even missed communication, whilst using new dedicated communication lines would be too expensive.

In order to solve such issues, we organize our framework as a *two-layer* architecture named Demand-Aware Network Constraint mAnager (DANCA) (see Figure 1). Namely, layer 1 is a centralized software service called ADAPT [2]. One instance of ADAPT has to be run *for each substation*, with a periodicity of one day. This entails that ADAPT instances are run at the DSO premises, possibly using powerful workstations. The main goal of ADAPT is to acquire power demands from all houses and compute individualized power bounds P_u^{low}, P_u^{high} for the next day. If all houses $u \in U$ are able to keep their resulting demand $e_u(t)$ inside the bounds $[P_u^{low}(t), P_u^{high}(t)]$ for all $t \in T_s$, then the aggregated power outside of the desired substation power profile $\Delta(e)$ is minimized. Note that power bounds P_u^{low}, P_u^{high} may be sent via the Internet to each house u , as such communication takes place only once a day and may be delayed. Finally, we note that ADAPT uses coarse-grained time slots (i.e., one hour). Layer 2 is a decentralized software called LAHEMS. One instance of LAHEMS must be run *on each house* $u \in U$. This entails that LAHEMS must be run on inexpensive low-resources hardware (a Raspberry Pi in our experiments). The main goal of LAHEMS is to acquire the current power demand and State of Charge (SoC) of the battery in house u , and to compute the charge/discharge actions for the battery itself. Such actions will modify the home demand $e_u(t) = d_u(t) + a_u(t)$, by either increasing it (*charge* action $a_u(t) > 0$, e.g., from 7 to 8 AM in Figure 2 (right)) or decreasing it (*discharge* action $a_u(t) < 0$, e.g., from 8 to 9 AM in Figure 2 (right)). The objective is to minimize the power outside the bounds $[P_u^{low}(t), P_u^{high}(t)]$ provided by the ADAPT service, without compressing or increasing the user demand in the full period. Furthermore, if an EV is also present, then LAHEMS may also be used to drive the EV charge/discharge (thus employing the so-called V2H). Note that: 1) The battery is always plugged-in and ready to accept charge/discharge commands. On the contrary, the EV is plugged-in only when the residential user decides to do so. 2) There are no restrictions, other than the physical ones (e.g., do not exceed the maximum power rate), on battery usage. On the contrary, the EV, once plugged-in, must be fully charged within a given deadline. 3) Both battery and EV must be equipped with a Battery Energy Manager (BEM) [31], [32], [33], [34], accepting (wireless) commands to: 1) read the current SoC; 2) charge/discharge the battery/EV. In this latter case, the BEM receives a software signal $a \in \mathbf{R}$, and the battery/EV is charged (if $a \geq 0$) or discharged (otherwise) with a kW rate until the next signal a' is received.

III. METHODOLOGY

In this section, we describe our DANCA service, by giving details of ADAPT and LAHEMS (for a high-level view, see Figure 1). In the following, for both services, we will distinguish between *configuration input* and *online input*. That is, configuration input must be given once and for all when

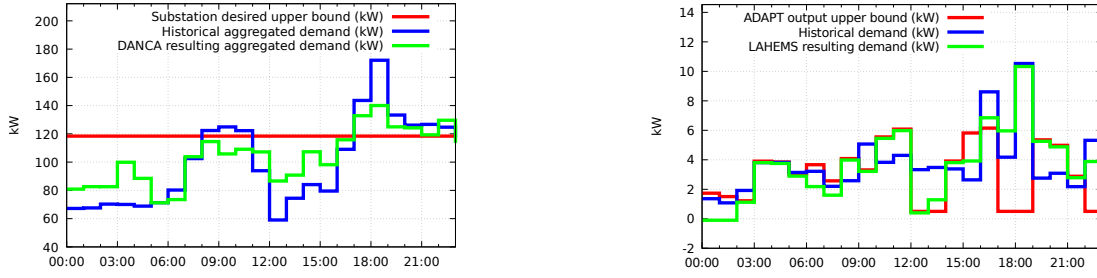


Fig. 2: Input and output for layer 1 (left) and layer 2 (right) in the most demanding day of our experiments (2014-01-29, see Section IV). Left figure shows aggregated power demand before (“historical”) and after (“DANCA resulting”) applying DANCA, right figure shows power demand of the most demanding house before (“historical”) and after (“LAHEMS resulting”) applying LAHEMS on that house

starting a service for the first time, while online input needs to be periodically acquired.

ADAPT Input and Output. The main configuration input consists of the following: 1) Duration $\tau_l \in \mathbf{R}_+$, in minutes, of the power limits output from ADAPT. 2) Period $\tau_s \in \mathbf{R}_+$, in minutes, of ADAPT invocations (i.e., ADAPT computes output power limits every τ_s minutes). Note that time slots duration τ_l must divide τ_s . This also defines the horizon length $H_s = \frac{\tau_s}{\tau_l}$ of the MPC methodology used by ADAPT. 3) For each house $u \in U$, battery maximum capacity $Q_{u,E}$ (in kWh) and battery maximum and minimum power rates $M_{u,E}$ and $m_{u,E}$, in kW (see, e.g., [2], [10]). 4) For each house $u \in U$, minimum and maximum power demand (in kW) $C_u^{low}, C_u^{high} \in \mathbf{R}_+$, as from the electricity contract.

The online input consists of the following: 1) (Ordered) set $T_s = \{t_1, \dots, t_{H_s}\}$ of the future time slots, each lasting τ_l minutes. 2) Desired bounds for the substation s to which houses in U are connected $P_s^{low}, P_s^{high} : T_s \rightarrow \mathbf{R}$ (in kW). 3) For each house $u \in U$, power demand \tilde{d}_u (in kW), as the difference between consumption (from appliances and EV) and production (from Photovoltaic Panels), taken at intervals at least τ_l . This is used to compute $d_u : T_s \rightarrow \mathbf{R}$ (in kW) as power demand forecasted for the next period T_s . Here we are interested in computing the forecast in negligible time, thus, for a given time slot t , the forecast is computed by a discounted average on the demands in the same time slot t in the past days (in our experiments, we consider 10 days in the past). For an overview of demand forecasting methods, see [35], [36]. Note that, as ADAPT cannot directly drive EVs, the power used on each house to recharge the EV is included in d_u .

Finally, the ADAPT output consists, for each house $u \in U$, of two power profiles $P_u^{low}, P_u^{high} : T_s \rightarrow \mathbf{R}$. Such power profiles will be given as input to LAHEMS. Namely LAHEMS, executed at u premises, will have to keep the resulting power demand e_u inside $[P_u^{low}(t), P_u^{high}(t)]$ as most as possible, for all time slots t . If each LAHEMS running on each house $u \in U$ succeeds in this task, then the overall aggregated power outside the desired substation bounds $\Delta(e)$ will be minimized.

LAHEMS Input and Output. In the following, we focus on a given house $u \in U$, thus we will assume index u to be understood. The main configuration input of LAHEMS consists of: 1) The starting horizon length $H_i \in \mathbf{N}$ and

horizon length changing step $H_\delta \in \mathbf{N}$ used for the Adaptive Model Predictive Control (AMPC) methodology employed by LAHEMS. 2) The period $\tau \in \mathbf{R}_+$, in minutes, of LAHEMS invocations (i.e., LAHEMS decides an action every τ minutes). We also require EV and battery actions to be computed within ζ minutes. This allows LAHEMS to correctly assume that computed actions will be held for $\tau - \zeta$ minutes. Namely, if ζ is sufficiently low, computed actions will be actually held by EV and battery for almost τ minutes. 3) EV maximum capacity Q_P (in kWh) and EV maximum and minimum power rates M_P and m_P (in kW). Furthermore, battery and EV efficiency for charge α_E, α_P and discharge β_E, β_P , respectively.

On the other hand, the online input consists of the following: 1) The (ordered) set $T = \{t_1, \dots, t_H\}$ of the future time slots. All time slots except t_1 last τ_l minutes, i.e., the frequency of changes in power limits. Duration of t_1 is defined so as t_2 starts at a multiple of τ_l . E.g., if power limits change every hour ($\tau_l = 60$) and the current time-stamp is 10:15, t_1 will last 45 minutes. 2) The power limits for u as an output from ADAPT. 3) Power demand \tilde{d} currently being requested to the grid (excluding EV, which is managed separately). Using the same techniques of ADAPT, the forecast for the demand $d : T \rightarrow \mathbf{R}$ on the next H periods of τ_l minutes is computed. 4) Current state of charge for both the battery \tilde{Q}_E and the EV \tilde{Q}_P ($\tilde{Q}_P = -1$ if it is currently not plugged-in). 5) If $\tilde{Q}_P \neq -1$, the deadline for EV recharging $\mathcal{D}_P \in \mathbf{N}$, s.t. $\mathcal{D}_P = i \geq 0$ iff the EV must be completely recharged in at most i minutes. We assume that the residential user manually specifies the deadline for the complete EV recharge when plugging the EV.

Finally, the LAHEMS output consists of commands $a_E, a_P \in \mathbf{R}$. Namely, a_E is the charge (if $a_E \geq 0$) or discharge ($a_E < 0$) command, in kW, for the battery in the current time slot. Analogously, if the EV is plugged-in (i.e., if $\tilde{Q}_P \geq 0$) then a_P is the charge/discharge command for the EV.

ADAPT Base Algorithm. Both ADAPT and LAHEMS algorithms are based on the MPC methodology. As for the system model (e.g., batteries and power demand) as well as for the underlining MPC scheme, we follow well-established approaches from the literature, e.g., [2], [10]. Every τ_s minutes, the ADAPT algorithm computes the power limits for

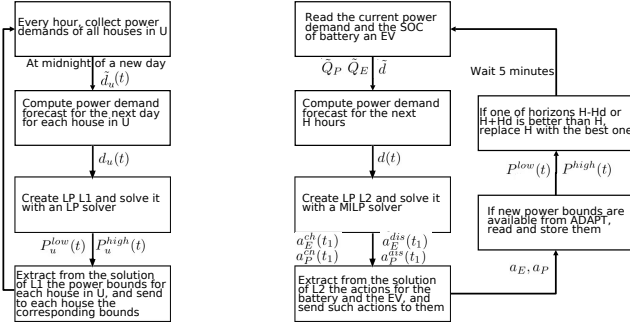


Fig. 3: Simplified control-flow diagram for ADAPT (left) and LAHEMS (right), also showing the main variables used

all houses $u \in U$. To this aim, a Linear Programming (LP) problem L_1 (designed by suitably extending [2]) with receding horizon H_s for layer 1 is generated and solved. Namely, L_1 contains $6|U||T_s| + 3|T_s| + |U|$ constraints, defined over $4|U||T_s| + |U| + 2|T_s|$ real-valued variables, which are detailed in the following.

$$\forall u \in U, t \in T_s, b_{u,E}(t+1) = b_{u,E}(t) + \frac{\tau_l}{60} a_{u,E}(t) \quad (1)$$

Constraint (1) states that the SoC $b_{u,E}(t+1)$ (in kWh) of the battery in house u at a given time slot $t+1$ is the result of applying action $a_{u,E}(t)$ (in kW) to SoC $b_{u,E}(t)$ at the previous time slot t , also considering time slot duration in minutes τ_l .

$$\forall u \in U, b_{u,E}(t_1) = b_{u,E}(t_{|T_s|} + 1) = \frac{Q_{u,E}}{2} \quad (2)$$

Constraint (2) states that the behavior of the battery must be cyclic, i.e., the starting and ending SoC of the battery (within time slots set T_s , which lasts one day in our experiments) in a given house u must be both half of the battery maximum capacity $Q_{u,E}$. In this way, there are no preferences among different executions of ADAPT in different days.

$$\forall u \in U, t \in T_s, P_u^{low}(t) \leq a_{u,E}(t) + d_u(t) \leq P_u^{high}(t) \quad (3)$$

Constraint (3) states that the *collaborative power profile* for each $u \in U$ must always be inside the bounds P_u^{low}, P_u^{high} to be output for user u . Note that, following the nomenclature in [2], “collaborative power profile” does not refer to collaborations between users, but to the fact that each user is willing to follow the price policies decided by DSO using ADAPT. Namely, such collaborative power profile is defined by applying an action on the battery $a_{u,E}(t)$ to the current demand $d_u(t)$, thus obtaining $a_{u,E}(t) + d_u(t)$.

$$\forall u \in U, t \in T_s, 0 \leq b_{u,E}(t) \leq Q_{u,E} \quad (4)$$

$$\forall u \in U, t \in T_s, m_{u,E} \leq a_{u,E}(t) \leq M_{u,E} \quad (5)$$

Constraints (4) and (5) require that the power demand shift $a_{u,E}$ for user u is within the limits of user u flexibility, i.e., within the power rate and capacity of the battery in house u (also Constraint (1) is involved, as it defines the future SoC).

$$\forall u \in U, t \in T_s, C_u^{low} \leq P_u^{low}(t) \leq P_u^{high}(t) \leq C_u^{high} \quad (6)$$

Constraint (6) requires that the output bounds P_u^{low}, P_u^{high} for user u must be within the electricity contract of u .

$$\forall t \in T_s, \Delta^{high}(t) \geq 0, \Delta^{low}(t) \geq 0 \quad (7)$$

$$\forall t \in T_s, \sum_{u \in U} P_u^{high}(t) \leq P_s^{high}(t) + \Delta^{high}(t) \quad (8)$$

$$\forall t \in T_s, \sum_{u \in U} P_u^{low}(t) \geq P_s^{low}(t) - \Delta^{low}(t) \quad (9)$$

Constraints (7)–(9) define the worst-case aggregated powers Δ^{high} exceeding P_s^{high} and Δ^{low} going below P_s^{low} . The objective function of L_1 is to minimize all such aggregated power exceeding substation bounds, i.e., $\sum_{t \in T_s} \Delta^{high}(t) + \sum_{t \in T_s} \Delta^{low}(t)$.

Finally, every τ_s minutes, the ADAPT output values for power limits $P_u^{low}(t), P_u^{high}(t)$, for all $u \in U, t \in T_s$, are computed by solving, every τ_s minutes, the LP problem L_1 via a LP solver and then extracting, from the obtained solution, the values for decision variables $P_u^{low}(t), P_u^{high}(t)$ (the corresponding control-flow diagram is shown in the left part of Figure 3).

LAHEMS Base Algorithm. In each house $u \in U$, every τ minutes, the main LAHEMS algorithm computes charge/discharge decisions on battery and/or EV, basing on the battery and/or EV current SoC, on the current household power demand, on forecasted future power demand, and on the known future power limits P_u^{low}, P_u^{high} from ADAPT. In order to compute the charge/discharge commands, a Mixed Integer Linear Programming (MILP) problem L_2 with receding horizon H for layer 2 is generated and solved. L_2 consists of two separate sets L_{2E}, L_{2P} of constraints. Constraints in L_{2E} deal with fixed battery dynamics and thus are always present. Constraints in L_{2P} are defined only when the EV is plugged-in. We have that L_{2E} is defined by $22H + 2$ constraints on $5H + 1$ continuous decision variables and $4H$ binary decision variables. On the other hand, L_{2P} is defined by at most $6H + 2$ constraints on at most $3H$ continuous decision variables and H binary decision variables, depending on $|T_P|$, being T_P the subset of time slots in T in which the EV will stay plugged-in. In the following, we describe such constraints in more detail. We recall that we focus on a given house $u \in U$, thus we will assume index u to be understood.

$$\forall t \in T, e(t) = d(t) + a_E^{ch}(t) - \beta_E a_E^{dis}(t) + \eta(t)(a_P^{ch}(t) - \beta_P a_P^{dis}(t)) \quad (10)$$

$$\forall t \in T, C^{low} \leq e(t) \leq C^{high} \quad (11)$$

Constraint (10) defines the power $e(t)$ requested or generated by the house in time slot $t \in T$ as the sum of all houses power consumption (power demand $d(t)$, charge commands for battery and EV $a_E^{ch}(t), a_P^{ch}(t)$) and power production (discharge commands for battery and EV $a_E^{dis}(t), a_P^{dis}(t)$). All such values are in kW. Battery and EV discharge commands also take into account round-trip inefficiencies $0 < \beta_E, \beta_P < 1$. Constant $\eta(t)$ is defined, for a given time slot t , as the fraction of t in which the EV is plugged-in. Constraint (11) requires such resulting power demand $e(t)$ to be inside the ranges of the household electricity contract.

$$b_E(t_1) = \tilde{Q}_E, 0 \leq b_E(t_H + 1) \leq Q_E \quad (12)$$

$$\forall t \in T, b_E(t+1) = b_E(t) + \frac{\tau_l}{60} (\alpha_E a_E^{ch}(t) - a_E^{dis}(t)) \quad (13)$$

$$\forall t \in T, y_E(t) \rightarrow a_E^{ch}(t) = 0, \neg y_E(t) \rightarrow a_E^{dis}(t) = 0 \quad (14)$$

$$\forall t \in T, 0 \leq b_E(t) \leq Q_E, 0 \leq a_E^{ch}(t) \leq M_E, 0 \leq a_E^{dis}(t) \leq m_E \quad (15)$$

Constraints (12) and (13) define the behavior of the battery, i.e., the starting SoC \tilde{Q}_E (in kWh) is read from sensors, and the SoC at time $t+1$ is obtained by adding to the SoC at time t the action taken at time slot t , multiplied by the time slot duration $|t|$. In case of a charge action, the efficiency coefficient $0 < \alpha_E < 1$ is also considered. Constraint (14) allows us to distinguish between a charge and a discharge action, which is required to apply the known battery efficiencies α_E, β_E . Furthermore, the battery physical constraints on power rate and capacity are taken into account by Constraint (15). Note that the Constraints (14) are *guarded constraints* of the form $\gamma \rightarrow L(\mathcal{X}) \leq K$ or $\neg\gamma \rightarrow L(\mathcal{X}) \leq K$, where L is a linear function, \mathcal{X} is a set of bounded variables (i.e., all variables in \mathcal{X} are defined on a suitable bounded interval), γ is a binary variable not in \mathcal{X} and K is a constant. Since all our decision variables are bounded, such constraints are translated into linear constraints as follows: $\gamma \rightarrow L(\mathcal{X}) \leq K$ is equivalent to $(\sup(L(\mathcal{X})) - K)\gamma + L(\mathcal{X}) \leq \sup(L(\mathcal{X}))$, while $\neg\gamma \rightarrow L(\mathcal{X}) \leq K$ is equivalent to $(K - \sup(L(\mathcal{X})))\gamma + L(\mathcal{X}) \leq K$. In such formulas, $\sup(L(\mathcal{X}))$ may be easily computed as L is linear and all variables in \mathcal{X} are bounded [2].

$$b_P(t_1) = \tilde{Q}_P \quad (16)$$

$$\forall t \in T_P. b_P(t+1) = b_P(t) + \eta(t) \frac{|t|}{60} (\alpha_P a_P^{ch}(t) - a_P^{dis}(t)) \quad (17)$$

$$\forall t \in T_P. y_P(t) \rightarrow a_P^{ch}(t) = 0, \neg y_P(t) \rightarrow a_P^{dis}(t) = 0 \quad (18)$$

$$\forall t \in T_P. 0 \leq b_P(t) \leq Q_P, 0 \leq a_P^{ch}(t) \leq M_P, 0 \leq a_P^{dis}(t) \leq m_P \quad (19)$$

Constraints (16)–(19) define the analogous behavior for the EV. Note that such constraints are defined on the set $T_P = \{t_1, \dots, \tilde{t}\}$ of the time slots in which the EV is actually plugged-in. This implies that Constraints (16)–(19) are only present when the EV is currently plugged-in, thus they are in $L_{2,P}$. Note that the Constraints (18) are guarded constraints (see above).

$$b_P(\tilde{t}+1) = \min\{Q_P, \tilde{Q}_P + \alpha_P M_P \mathcal{D}_P\} \min\left\{1, \frac{\sum_{t \in T} |t|}{\mathcal{D}_P}\right\} \quad (20)$$

To define the goal for EV recharging, we have to consider two aspects, both handled by Constraint (20) in $L_{2,P}$. On the one hand, the input deadline specified by the user for the EV complete recharge may be infeasible w.r.t. the current SoC (e.g., it is infeasible to completely recharge the EV from 0 kWh in 1 hour). In order to avoid L_2 to turn out infeasible only because of this, LAHEMS first computes the SoC attainable with the currently specified deadline, i.e., $\tilde{Q}_P + \alpha_P M_P \mathcal{D}_P$. On the other hand, the EV may be expected to be unplugged at time slot \tilde{t} within the current time horizon, or in a time slot that will be considered in a future MILP. In the former case, the EV must be completely charged at \tilde{t} . In the latter case, we require the final charge of the EV to be proportional to the remaining time before unplugging the EV, i.e., $\frac{\sum_{t \in T} |t|}{\mathcal{D}_P}$.

$$\forall t \in T. y^{high}(t) \rightarrow e(t) \leq P^{high}(t) \quad (21)$$

$$\forall t \in T. \neg y^{high}(t) \rightarrow e(t) \geq P^{high}(t) \quad (22)$$

$$\forall t \in T. y^{low}(t) \rightarrow e(t) \geq P^{low}(t) \quad (23)$$

$$\forall t \in T. \neg y^{low}(t) \rightarrow e(t) \leq P^{low}(t) \quad (24)$$

$$\forall t \in T. y^{in}(t) \rightarrow y^{high}(t) + y^{low}(t) \geq 2 \quad (25)$$

$$\forall t \in T. \neg y^{in}(t) \rightarrow y^{high}(t) + y^{low}(t) \leq 1 \quad (26)$$

$$\forall t \in T. y^{in}(t) \rightarrow z(t) = 0 \quad (27)$$

$$\forall t \in T. \neg y^{high}(t) \rightarrow z(t) = e(t) - P^{high}(t) \quad (28)$$

$$\forall t \in T. \neg y^{low}(t) \rightarrow z(t) = e(t) - P^{low}(t) \quad (29)$$

$$\forall t \in T. 0 \leq z(t) \leq \max\{P^{low}(t), C^{low}(t) - C^{high}(t) - P^{high}(t)\} \quad (30)$$

The objective function of L_2 minimizes the power outside the limits decided by ADAPT for the given house. To this aim, such exceeding power in time slot $t \in T$ is modeled by variable $z(t)$ (having bounds as in Constraint (30)), thus the objective function to be minimized is $\sum_{t \in T} z(t)$. By using guarded constraints (see above), the decision variables $z(t)$ are defined by:

- Constraints (21) and (22), where binary variable $y^{high}(t)$ is set to true iff the resulting power $e(t)$ exceeds the upper bound $P^{high}(t)$;
- Constraints (23) and (24), where binary variable $y^{low}(t)$ is set to true iff the resulting power $e(t)$ is below the lower bound $P^{low}(t)$;
- Constraints (25) and (26), where binary variable $y^{in}(t)$ is set to true iff the resulting power $e(t)$ is inside the bounds interval $[P^{low}(t), P^{high}(t)]$;
- Constraints (27)–(29), where, exploiting the binary variables defined in Constraints (21)–(26), the real variable $z(t)$ is defined to be 0 iff $e(t) \in [P^{low}(t), P^{high}(t)]$, and to be the power outside the bounds interval $[P^{low}(t), P^{high}(t)]$ otherwise.

Finally, every τ minutes, the LAHEMS output values for battery and/or EV charging/discharging actions are computed by solving the MILP problem L_2 via a MILP solver and then extracting, from the obtained solution, the actions for the first time slot in T . If L is infeasible, then a default action is selected, which is designed to minimize user discomfort. That is, if the EV is not currently plugged-in or is already fully charged, then no action is taken, i.e., $(a_E, a_P) = (0, 0)$. Otherwise, the EV is charged as much as possible, also discharging the battery as much as possible, i.e., $a_P = \min\{M_P, \frac{Q_P - \tilde{Q}_P}{\tau}, C^{high} - d(t_1) + \beta_E \min\{m_E, \frac{Q_E - \tilde{Q}_E}{\tau}\}\}$, $a_E = -\min\{m_E, \frac{Q_E - \tilde{Q}_E}{\tau}, \frac{a_P}{\beta_E}\}$. Instead, if L is feasible, then, for $x \in \{E, P\}$, $a_x = a_x^{ch}(t_1)$ if $a_x^{ch}(t_1) \geq 0$, and $a_x = a_x^{dis}(t_1)$ otherwise. The corresponding control-flow diagram is shown in the right part of Figure 3.

LAHEMS Adaptive Algorithm. Our adaptive algorithm is based on the fact that using a high value for the horizon H does not imply that we obtain better exceeding power minimization. This is due to: i) uncertainties stemming from power demand forecasting, which are worse for higher values of the horizon, and ii) higher computation time typically required to solve MILPs with higher horizons, as the number of both constraints and decision variables linearly depends on H . Thus, our adaptive algorithm works as follows. Instead of setting up only one MILP L with fixed receding horizon H , as it is done in the literature, LAHEMS sets up and separately solves 3 different MILPs with receding horizon $h \in \{H, H - H_\delta, H + H_\delta\}$

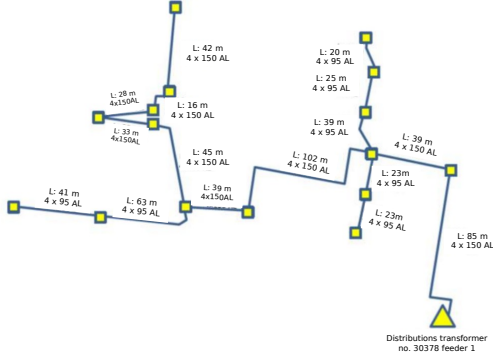


Fig. 4: Net topology of the substation and connected houses

respectively. For each of such MILPs, LAHEMS maintains the current objective function value $\hat{c}(h)$ (i.e., by accumulating the values of the MILP objective functions as computed by the MILP solver). When, for some $h \neq H$, the value for h is better than the value for the current horizon H , i.e., $\hat{c}(h) < \hat{c}(H)$, the current horizon H is updated to h and all objective function values $\hat{c}(h)$ are reset to 0. Note that only the actions computed with the current horizon H (which are computed first) are sent to batteries actuators. Namely, MILP problems with other horizons are only used to automatically adjust the current horizon, so that the exceeding power is further minimized.

In order to meet real-time requirements, LAHEMS uses a twofold strategy. First of all, LAHEMS stops MILP solver execution if it exceeds ζ minutes. In this case, the current MILP problem L is handled as if it were infeasible (i.e., minimizing user discomfort). This entails that battery actions are held for $\tau - \zeta$ minutes. Since, for small values of ζ , $\tau - \zeta \approx \tau$, this is in agreement with MILP constraints which assume computed actions for battery and EV to be held for all current time slot duration τ . Second, after having computed the actions for battery and EV, LAHEMS would be idle till the end of the current time slot duration τ (neglecting periodic tasks such as downloading new power limits or reading the current power demand). LAHEMS exploits such although idle time to compute the actions with horizons $H - H_\delta, H + H_\delta$, thus achieving horizon adjustment without overhead.

IV. EXPERIMENTAL SETUP

In this section, we describe how we organize our experiments, in order to show the feasibility of our approach. To this aim, we implemented both the ADAPT and the LAHEMS algorithm by using the Python language, and we use GNU Linear Programming Kit (GLPK) to solve MILP problems. The following results have been obtained by simulating ADAPT operation for one year on an Intel i7 2.5 GHz with 8GB of RAM, and by simulating LAHEMS operation for the same period, also considering output from ADAPT, on a Raspberry Pi Model B+ 700 MHz with 512 MB of RAM. Our experiments are organized as follows.

Key Performance Indicators. In order to evaluate our DANCA methodology, we define a set of meaningful Key Performance Indicators (KPIs), which are listed and explained in Table I.

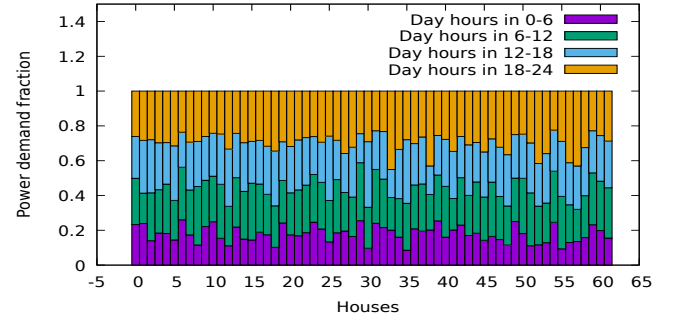


Fig. 5: Houses historical demand distribution

Substation, households and EVs. In order to accurately simulate ADAPT and LAHEMS operation for a long enough period, we need power demand data, taken at intervals of at least one hour, of each residential house connected to a given substation. To this aim, we use power demand recorded, from the beginning of September 2013 to the end of August 2014, in 62 households in a suburban area in Denmark. All such houses are connected to the same substation s (see Figure 4). Such data were recorded during the European Commission project “SmartHG” [37], [30] and consists, for each of the 62 houses, in the power demand recorded from house electricity main, with a resolution of 1 hour. We point out that Danish households use district heating for house heating and electricity for house appliances. We also use Photovoltaic Panel (PVP) energy production recorded in the same period and area. Such recorded data consider 6 kWp PVP installations, with highly seasonal productivity ranging from 200 kWh/month in December and above 1200 kWh/month from April to July. To assess the validity of our case study, we show that, with high probability, any operational scenario will be very close to one of those entailed by the houses U considered in our case study. To this end, we divide each day into 4 time slots t_1, \dots, t_4 of 6 hours each, with $t_i = [6(i-1), 6i)$ ($i = 1, \dots, 4$). Let $D_u(t_i)$ be the total electricity demand (for a whole year, in our case study) of house u within time slot t_i , let $D_u^{tot} = \sum_{i=1}^4 D_u(t_i)$ be the total (whole year) demand of house u and let $\tilde{D}_u(t_i)$ be the fraction of the demand of house u within time slot t_i , i.e., $\tilde{D}_u(t_i) = \frac{D_u(t_i)}{D_u^{tot}}$. On such a base, we define the *demand distribution* \tilde{D}_u for house u as $\tilde{D}_u = (\tilde{D}_u(t_1), \tilde{D}_u(t_2), \tilde{D}_u(t_3), \tilde{D}_u(t_4))$. Figure 5 shows the set of demand distributions $\tilde{D} = \{\tilde{D}_u \mid u \in U\}$. Our goal is to show that any *reasonable* demand distribution will not be *too different* from one of those in \tilde{D} . Accordingly, our set of *admissible* demand distributions is $D^* = \{(p_1, p_2, p_3, p_4) \mid (\bigwedge_{i=1}^4 p_i \in [0.1, 0.3]) \wedge (\sum_{i=1}^4 p_i = 1)\}$. Given a demand profile $p = (p_1, p_2, p_3, p_4) \in D^*$, we define the distance $rmse(p)$ of p from \tilde{D} as the minimum root mean square error, i.e., $rmse(p) = \frac{1}{2} \min \left\{ \sqrt{\sum_{i=1}^4 (p_i - \tilde{D}_u(t_i))^2} \mid u \in U \right\}$. Note that $rmse(p) = 0$ for $p \in \tilde{D}$. Using MonteCarlo-based statistical model checking techniques (e.g., as in [4]), we can show that, with probability at least 0.99, for a randomly selected $p \in D^*$ we have $rmse(p) \leq 0.08$. That is, with high

TABLE I: List of KPIs used for DANCA evaluation

KPI	Description
AvgSolTime	Average MILP Solving Time (in seconds), i.e., the average delay due to MILPs solution computation
MissDeadl	Missed Deadline for MILP Solving, i.e., the fraction of MILPs not solved within the $\zeta = 0.5$ mins deadline
HorChange	Horizon Changes. For user u , given $N(u)$ (number of times the adaptive algorithm decided to change the MPC receding horizon) and $T(u)$ (total number of MILPs), then $\text{HorChange} = N(u)/T(u)$
UserDiscomfort	Missed EV Deadlines. For user u , given $R(u)$ (number of times the EV was plugged) and $M(u)$ (number of times LAHEMS failed to fully re-charge the EV within the deadline), then $\text{UserDiscomfort} = M(u)/R(u)$
DemOutRed	DANCA (Hierarchical) Aggregated Demand Outside Bounds Reduction w.r.t. Historical Demand, i.e., $\text{DemOutRed} = 1 - \frac{\Delta(e)}{\Delta(d)}$ (see Section II).
DemOutRedOpt	Optimal (Centralised) Aggregated Demand Outside Bounds Reduction w.r.t. Historical Demand. Let $\Delta(c)$ be the overall aggregated ADAPT collaborative profile (see Section III) power outside the desired substation bounds. Then, $\text{DemOutRedOpt} = 1 - \frac{\Delta(c)}{\Delta(d)}$.

TABLE II: Parameters for DANCA evaluation

Param	Value	Explanation
$ U $	62	Number of houses
τ_s	1 day	Gap between two MILP solver invocations for ADAPT
τ_l	1 hour	Duration of time slots for power limits output by ADAPT
τ	5 min	Gap between two MILP solver invocations for LAHEMS
ζ	30 sec	Deadline for each MILP solver invocation in LAHEMS
H_i	6	Starting value for receding horizon in LAHEMS
H_δ	7	Changing step for receding horizon in LAHEMS
Q_E	13.5 kWh	Battery capacity on each house
M_E, m_E	3.3 kW	Battery power rate
α_1, β_1	0.9	Battery round-trip efficiencies
Q_P	16 kWh	EV capacity on each house
M_P, m_P	3.6 kW	EV power rate
α_2, β_2	0.876	EV round-trip efficiencies

probability, any admissible demand distribution will be *very close* to one of those considered in our case study. This shows that conclusions drawn from our case study can be safely generalized to other situations.

Furthermore, we virtually equip each house with EVs charging data taken from the “Test-an-EV” project [37]. Such data consists in plug-in time, SoC at plug-in time and unplug time for 184 EVs Mitsubishi i-MiEV, ranging from 2012 to 2013. The matching between a house and an EV has been done randomly. We remark that, in ADAPT experiments, EVs charging data is considered a further load for each house u , thus the power demand in a time slot t is incremented by the historical charging data of the EV connected to u . In LAHEMS experiments, we only take into account the starting time and starting SoC of each recharge, as well as its unplug time. Then, it is LAHEMS responsibility to decide charge/discharge actions for the EV.

Substation Bounds. We split our experiments into 3 *scenarios*, each corresponding to different values for the desired bounds on substation s required in input by ADAPT. In order to set up challenging scenarios for our DANCA methodology, we compute, from the historical data on houses power consumption described above (also considering EV recharging data), the daily average $A(D)$ and daily maximum $M(D)$ on the aggregated power demand, being D a given day. A scenario S is defined by setting, for each day D and all time slots t of D , $[P_s^{\text{low}}(t), P_s^{\text{high}}(t)] = [0, A(D) + S(M(D) - A(D))]$.

That is, we set the lower bound to be 0, as reverse power flows may damage any installed electrical equipment in the grid [3]. Instead, for the upper bound, we have that for $S = 0$ it coincides with the daily average, whilst for $S = 1$ it coincides with the daily maximum. The lower S , the more challenging our scenario is. In the following, we will consider 3 values for S , each defining a scenario, i.e., $S \in \{0, 0.25, 0.5\}$.

All other experimental parameters are shown in Table II. Simulation of ADAPT in a given day \mathcal{D} is carried out by taking as input the historical data on power demand of all houses $u \in U$ in day $\mathcal{D}-1$, as well as the substation bounds for day \mathcal{D} as discussed above. For each house $u \in U$, simulation of LAHEMS in time slot t is carried out by taking as input the power limits output by ADAPT for u in T_s , the power demand of u in t from historical data, and the estimated SoC of battery and EV in t . Such SoC is computed from the SoC and the charge/discharge action of the previous time slot $t-1$, using the first time slot of (13) and (17).

V. EXPERIMENTAL RESULTS

Given the experimental setting described in Section IV, in this section we show our experimental results. Namely, in Figure 6 we present, for each KPI of LAHEMS (see Table I), the corresponding statistics as Box-and-Whiskers plots [38]. Since each KPI is measured for every house, the plots in Figure 6 provide, for each KPI, the statistics (i.e., average, interquartile range, outliers) on all houses. Furthermore, Table III shows the results for the overall evaluation of the DANCA service. In the following, we discuss the results of the given KPIs.

Demand Outside Bounds Reduction and Users Discomfort. The main result of our DANCA service consists in the fact that we achieve a high reduction on the peaks outside substation desired bounds, without user discomfort. Table III shows that the demand peaks outside the substation desired bounds (considering both demands higher than the upper bound and lower than the lower bound) are halved in all our experimental scenarios. This is achieved with a Missed EV Deadlines (UserDiscomfort) equal to 0 in all houses, i.e., deadlines for EV recharging are never missed. Furthermore, the theoretical optimal reduction achieved in ideal conditions by the ADAPT collaborative profile computed in a centralised way is about 60% in all scenarios, thus our near-optimal approach is at least 82% as efficient as the theoretical optimal one.

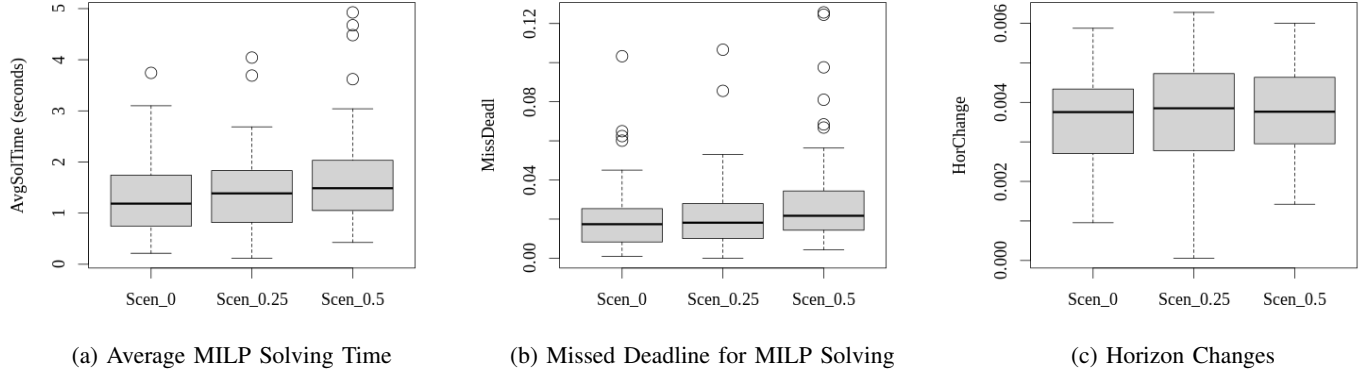


Fig. 6: Box-and-Whisker plots [38] for the main KPIs of LAHEMS. In each plot, the box boundaries are the farthest points that are not outliers (i.e., that are within 1.5 the interquartile range), the line inside the box is the statistical median, and the circles are data more than 1.5 times the interquartile range from the end of a box.

We remark that these results have been achieved by using the same inexpensive hardware that would have been needed by a non-MPC-based strategy. For the sake of comparison, consider a simple strategy where there is only one layer, consisting in a computational service acting on each house. The input of such service, call it Simple Demand-Aware Network Constraint mAnager (SDANCA), is the same as LAHEMS, but instead of the power bounds computed by ADAPT, SDANCA is directly fed with the desired substation bounds P_s^{low}, P_s^{high} and the number of houses $|U|$. Given this, SDANCA first computes the house power bounds as $\frac{P_s^{low}}{|U|}, \frac{P_s^{high}}{|U|}$. Then, SDANCA computes battery and EV actions in a greedy way, i.e., the battery is charged (within its physical limits) when the current demand is below the upper bound $\frac{P_s^{high}}{|U|}$, and discharged otherwise. Such a simple strategy, while requiring the same hardware as DANCA, achieves a reduction of substation constraint management violation of about 38%, compared to 50% obtained with our approach.

Computation Time. We recall that LAHEMS is designed to compute each battery and/or EV action in at most $\zeta = 30$ seconds. First of all, we point out that MILP deadlines violations are very few, as shown in Figure 6b. Namely, Figure 6b shows that, for most houses, at most 5% (and 2% on average) of all MILP problems solved require more than 30 seconds to be solved, thus missing the real-time deadline. Furthermore, also considering statistical outliers, the percentage of missed deadlines is always below 13% in all houses. On the other hand, Figure 6a shows that most MILP solver invocations, on average, require much lower computation time than 30 seconds. Namely, for most houses, the average of the times needed to solve all MILP problems (given there is one MILP solved every 5 minutes over one year, such average is computed on about 105,000 values on each house) turns out to be between 1 and 3 seconds in all scenarios. Furthermore, also considering the statistical outliers, for all houses the average MILP solving time is at most 5 seconds. Such results show that LAHEMS is indeed a lightweight application as required and fully meets the requirements for real-time computation

TABLE III: Results for DANCA evaluation

s	DemOutRed	DemOutRedOpt	$\frac{DemOutRed}{DemOutRedOpt}$
0	0.5	0.61	0.82
0.25	0.53	0.63	0.83
0.5	0.48	0.58	0.83

on the target low-resource device (Raspberry Pi). Finally, as for ADAPT computation time, it typically requires at most 1 second, which is negligible w.r.t. ADAPT periodicity (1 day). This is not surprising, as the MILP problems defined in ADAPT are actually LP problems (i.e., they do not involve binary decision variables).

Adaptive Algorithm Effectiveness. In order to show the effectiveness of the adaptive algorithm employed by LAHEMS, Figure 6c shows the results for the Horizon Changes (HorChange) KPI (see Table I). Namely, in all scenarios and all houses, there are 4 horizon changes every 1000 MILP solver invocations on average. This shows that our adaptive algorithm is effective, as it is able to adapt to very different conditions, depending on the scenario and the current home behaviour.

VI. CONCLUSIONS

In this paper, we presented Demand-Aware Network Constraint mAnager, a two-layer computing service that is able to enforce aggregated power demand constraints on Electrical Distribution Network substations. Demand-Aware Network Constraint mAnager is composed of two services, both based on the Model Predictive Control methodology: DemAnD-Aware Power limiT, operating once a day at the substation level and at utility premises, and Lightweight Adaptive Home Energy Management System (also employing a novel *adaptive* Model Predictive Control), operating once every 5 minutes at user premises on hardware with limited computational resources. More in detail, such services act as a hierarchical controller: DemAnD-Aware Power limiT sets

up the long-term goal for Lightweight Adaptive Home Energy Management System, which directly controls local home batteries via charge/discharge commands to meet such goals. Users privacy is also preserved, as only their overall demand is sent to the Distribution System Operator, as it already happens with home mains, and home batteries are not actuated by the Distribution System Operator.

Using power demands recorded in 62 houses in Denmark by the EU project SmartHG [30], we experimentally showed that Demand-Aware Network Constraint mAnager is able to reduce aggregated demand bounds violations w.r.t. the unmanaged demand by about 50% on average (w.r.t. 61% reduction obtained by a theoretical optimal centralized solution). This is achieved while meeting real-time requirements on the available hardware, both at the substation and at the houses level.

Our Demand-Aware Network Constraint mAnager framework currently focuses on satisfying the substation (feeder) power bounds. As future work, we plan to investigate how to extend it to enforce other network-level restrictions, e.g., on power flow.

REFERENCES

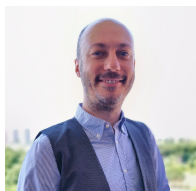
- [1] M. Uddin, M. F. Romlie, M. F. Abdullah, S. A. Halim, A. H. A. Bakar, and T. C. Kwang, "A review on peak load shaving strategies," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3323 – 3332, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364032117314272>
- [2] B. P. Hayes, I. Melatti, T. Mancini, M. Prodanovic, and E. Tronci, "Residential demand management using individualized demand aware price policies," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1284–1294, 2017. [Online]. Available: <https://doi.org/10.1109/TSG.2016.2596790>
- [3] D. Dongol, T. Feldmann, E. Bollin, and M. Schmidt, "A model predictive control based peak shaving application of battery for a household with photovoltaic system in a rural distribution grid," *Sustainable Energy, Grids and Networks*, vol. 16, pp. 1 – 13, 2018.
- [4] T. Mancini, F. Mari, I. Melatti, I. Salvo, E. Tronci, J. K. Gruber, B. Hayes, M. Prodanovic, and L. Elmegaard, "Parallel statistical model checking for safety verification in smart grids," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2018, pp. 1–6.
- [5] A. Jindal, M. Singh, and N. Kumar, "Consumption-aware data analytical demand response scheme for peak load reduction in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8993–9004, 2018.
- [6] C. E. Kement, H. Gultekin, and B. Tavli, "A holistic analysis of privacy-aware smart grid demand response," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7631–7641, 2021.
- [7] O. Erdinc, A. Tascikaraoglu, N. G. Paterakis, and J. P. S. Catalao, "Novel incentive mechanism for end-users enrolled in dlc-based demand response programs within stochastic planning context," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1476–1487, 2019.
- [8] D. Zhao, H. Wang, J. Huang, and X. Lin, "Virtual energy storage sharing and capacity allocation," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1112–1123, 2020.
- [9] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1066–1076, 2020.
- [10] N. Zhang, B. D. Leibowicz, and G. A. Hanasusanto, "Optimal residential battery storage operations using robust data-driven dynamic programming," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1771–1780, 2020.
- [11] B. Zhou, K. Zhang, K. W. Chan, C. Li, X. Lu, S. Bu, and X. Gao, "Optimal coordination of electric vehicles for virtual power plants with dynamic communication spectrum allocation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 450–462, 2021.
- [12] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Fog-computing-based energy storage in smart grid: A cut-off priority queuing model for plug-in electrified vehicle charging," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3470–3482, 2020.
- [13] D. Setlhaolo, X. Xia, and J. Zhang, "Optimal scheduling of household appliances for demand response," *Electric Power Systems Research*, vol. 116, pp. 24–28, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378779614001527>
- [14] A. Jindal, B. S. Bhambhu, M. Singh, N. Kumar, and K. Naik, "A heuristic-based appliance scheduling scheme for smart homes," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3242–3255, 2020.
- [15] A. Saad, T. Youssef, A. T. Elsayed, A. Amin, O. H. Abdalla, and O. Mohammed, "Data-centric hierarchical distributed model predictive control for smart grid energy management," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4086–4098, 2019.
- [16] D. Li, W.-Y. Chiu, H. Sun, and H. V. Poor, "Multiobjective optimization for demand side management program in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1482–1490, 2018.
- [17] Y. Chai, L. Guo, C. Wang, Y. Liu, and Z. Zhao, "Hierarchical distributed voltage optimization method for hv and mv distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 968–980, March 2020.
- [18] X. Wu, Y. Xu, X. Wu, J. He, J. M. Guerrero, C. Liu, K. P. Schneider, and D. T. Ton, "A two-layer distributed cooperative control method for islanded networked microgrid systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 942–957, 2020.
- [19] D. I. Brandao, W. M. Ferreira, A. M. S. Alonso, E. Tedeschi, and F. P. Marafão, "Optimal multiobjective control of low-voltage ac microgrids: Power flow regulation and compensation of reactive power and unbalance," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1239–1252, 2020.
- [20] M. Razmara, G. R. Bharati, M. Shahbakhti, S. Paudyal, and R. D. Robinett, "Bilevel optimization framework for smart building-to-grid systems," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 582–593, 2018. [Online]. Available: <https://doi.org/10.1109/TSG.2016.2557334>
- [21] W. Shi, X. Xie, C. Chu, and R. Gadh, "Distributed optimal energy management in microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1137–1146, 2015.
- [22] D. H. Nguyen, S.-I. Azuma, and T. Sugie, "Novel control approaches for demand response with real-time pricing using parallel and distributed consensus-based admn," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7935–7945, 2019.
- [23] K. Trangbaek, J. Bendtsen, and J. Stoustrup, "Hierarchical control for smart grids," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6130 – 6135, 2011, 18th IFAC World Congress. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667016445866>
- [24] P. Paudyal, P. Munankarmi, Z. Ni, and T. M. Hansen, "A hierarchical control framework with a novel bidding scheme for residential community energy optimization," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 710–719, 2020.
- [25] S. Thangavel, S. Lucia, R. Paulen, and S. Engell, "Dual robust nonlinear model predictive control: A multi-stage approach," *Journal of Process Control*, vol. 72, pp. 39 – 51, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0959152418304050>
- [26] M. Guay, V. Adetola, and D. DeHaan, *Robust and Adaptive Model Predictive Control of Nonlinear Systems*. Institution of Engineering and Technology, 2015.
- [27] M. Short and F. Abugchem, "A microcontroller-based adaptive model predictive control platform for process control applications," *Electronics*, vol. 6, no. 4, 2017. [Online]. Available: <http://www.mdpi.com/2079-9292/6/4/88>
- [28] C. Chen, J. Wang, Y. Heo, and S. Kishore, "MPC-based appliance scheduling for residential building energy management controller," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1401–1410, 2013.
- [29] S. Lucia, M. J. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen, "Predictive control, embedded cyberphysical systems and systems of systems - A perspective," *Annual Reviews in Control*, vol. 41, pp. 193–207, 2016.
- [30] V. Alimguzhin, F. Mari, I. Melatti, E. Tronci, E. Ebeid, S. Mikkelsen, R. Hylsberg Jacobsen, J. Gruber, B. Hayes, F. Huerta, and M. Prodanovic, "A glimpse of smarthg project test-bed and communication infrastructure," in *Digital System Design (DSD), 2015 Euromicro Conference on*, Aug 2015, pp. 225–232.
- [31] V. Pop, H. J. Bergveld, D. Danilov, P. P. L. Regtien, and P. H. L. Notten, *Battery Management Systems: Accurate State-of-Charge Indication for Battery-Powered Applications*. Springer, 2008.
- [32] L. He, L. Kong, S. Lin, S. Ying, Y. J. Gu, T. He, and C. Liu, "RAC: reconfiguration-assisted charging in large-scale lithium-ion battery systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1420–1429, 2016. [Online]. Available: <https://doi.org/10.1109/TSG.2015.2450727>

- [33] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018.
- [34] K. Li, F. Wei, K. J. Tseng, and B.-H. Soong, "A practical lithium-ion battery model for state of energy and voltage responses prediction incorporating temperature and ageing effects," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6696–6708, 2018.
- [35] C. Yu, P. W. Mirowski, and T. K. Ho, "A sparse coding approach to household electricity demand forecasting in smart grids," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 738–748, 2017. [Online]. Available: <https://doi.org/10.1109/TSG.2015.2513900>
- [36] T. Li, Y. Wang, and N. Zhang, "Combining probability density forecasts for power electrical loads," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1679–1690, 2020.
- [37] European Commission project SmartHG. (2015). [Online]. Available: <http://smarthg.di.uniroma1.it/>
- [38] J. W. Tukey, *Exploratory Data Analysis*, ser. Behavioral Science: Quantitative Methods. Reading, Mass.: Addison-Wesley, 1977.



Igor Melatti was born in Treviso, Italy. He received the Master Degree in Computer Science and the PhD in Computer Science and Applications from the University of L'Aquila, Italy in 2001 and 2005 respectively. He has been a PostDoc at the University of Utah and at the Sapienza University of Rome. He is currently an Associate Professor at the Computer Science Department of Sapienza University of Rome. His current research interests comprise: formal methods, automatic verification algorithms, model check-

ing, software verification, cyber-physical systems, automatic synthesis of reactive programs from formal specification, systems biology, smart grids.



Federico Mari was born in Treviso, Italy. He received the Master Degree and the PhD in Computer Science from the Sapienza University of Rome, Italy in 2006 and 2010 respectively. He is currently an Assistant Professor (tenure track) of Computer Science at the Department of Movement, Human and Health Sciences of the University of Rome Foro Italico (from 2019), where he serves as Rector's delegate for ICT. His primary research interest is in formal methods and artificial intelligence applied to smart

grids, health and sport sciences.



Enrico Tronci is a Full Professor at the Computer Science Department of Sapienza University of Rome (Italy). He received a master's degree in Electrical Engineering from Sapienza University of Rome and a Ph.D. degree in Applied Mathematics from Carnegie Mellon University. His research interests comprise: formal verification, model checking, system level formal verification, hybrid systems, embedded systems, cyber-physical systems, control software synthesis, smart grids, autonomous demand and

response systems for smart grids, systems biology



Milan Prodanovic (M01) received the B.Sc. degree in electrical engineering from the University of Belgrade, Serbia, in 1996 and the Ph.D. degree from Imperial College, London, U.K., in 2004. From 1997 to 1999 he was engaged with GVS engineering company, Serbia, developing UPS systems. From 1999 until 2010 he was a research associate in Electrical and Electronic Engineering at Imperial College, London, UK. Currently he is a Senior Researcher and Head of the Electrical Systems Unit at Institute IMDEA

Energy, Madrid, Spain. His research interests include design and control of power electronics interfaces for distributed generation, micro-grids control and active management of distribution networks.



Toni Mancini has a Ph.D. in Computer Science Engineering and is Associate Professor at the Computer Science Department of Sapienza University of Rome, Rome, Italy. His research interests comprise: artificial intelligence, formal verification, cyber-physical systems, control software synthesis, systems biology, smart grids.

This article appears in IEEE Transactions on Industrial Electronics, 2021. DOI: 10.1109/TIE.2021.3102431