D3.5  Distributed Artificial Intelligence
Final Version

Deliverable

Nespoli, Lorenzo: Institute for Applied Sustainability to the Built Environment
University of Applied Sciences and Arts of Southern Switzerland

Medici, Vasco: Institute for Applied Sustainability to the Built Environment
University of Applied Sciences and Arts of Southern Switzerland

2020-05-15

This project has received funding in the framework of the joint programming initiative ERA-Net Smart Energy Systems’ focus initiative Smart Grids Plus, with support from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 646039.
INTERNAL REFERENCE

Deliverable No.: D3.5
Deliverable Name: Distributed Artificial Intelligence
Lead Partner: SUPSI
Work Package No.: 3
Task No. & Name: T3.5: Development of a distributed artificial intelligence
Document (File): D3.5.docx
Issue (Save) Date: 2020-05-15

DOCUMENT SENSITIVITY

- Not Sensitive
  Contains only factual or background information; contains no new or additional analysis, recommendations or policy-relevant statements

- Moderately Sensitive
  Contains some analysis or interpretation of results; contains no recommendations or policy-relevant statements

- Sensitive
  Contains analysis or interpretation of results with policy-relevance and/or recommendations or policy-relevant statements

- Highly Sensitive Confidential
  Contains significant analysis or interpretation of results with major policy-relevance or implications, contains extensive recommendations or policy-relevant statements, and/or contain policy-prescriptive statements. This sensitivity requires SB decision.

DOCUMENT STATUS

<table>
<thead>
<tr>
<th>Date</th>
<th>Author(s)</th>
<th>Person(s)</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-05</td>
<td></td>
<td>Nespoli, Lorenzo; Medici, Vasco</td>
<td>SUPSI</td>
</tr>
<tr>
<td>Approval by</td>
<td>2020-05</td>
<td>Matthiess, Benjamin</td>
<td>ZSW</td>
</tr>
</tbody>
</table>
CONTENTS

1 Introduction ................................................................................................................................................. 4
2 Distributed Artificial Intelligence For Grid Control ...................................................................................... 4
  2.1 Distributed Control for Business as Usual and DSO-Planned .............................................................. 4
  2.2 Distributed Coordination ......................................................................................................................... 8
REFERENCES ..................................................................................................................................................... 9

DISCLAIMER

The content and views expressed in this material are those of the authors and do not necessarily reflect the views or opinion of the ERA-Net SG+ initiative. Any reference given does not necessarily imply the endorsement by ERA-Net SG+.

ABOUT THE NEMoGRID PROJECT

The NEMoGrid Project is mainly focused on the definition of innovative business models that could ease the penetration of renewables into the distribution grid, with an emphasis on the definition of a peer-to-peer strategy based on the blockchain technology. The new business models will encourage the active participation of citizens and the assumption of their new role of prosumers, by allowing them to enter new markets as players. Among the tested scenarios, the most innovative one will be based on a peer-to-peer market. In this case, new decentralized platforms based on the blockchain technology will allow zero marginal cost transactions. In order to test the new business models effectiveness, a simulation framework will be developed. Each scenario will be evaluated based on a number of KPIs. Existing demo sites in Rolle and Lugaggia (CH), Björklinge (SE) and Wüstenrot (DE) will be used to validate the business model that gives the best simulation results. Real loads will be controlled by the algorithms developed in the simulation phase. Technical developments within NEMoGrid will be supported with user research, gathering empirical data on prosumers decisions and interactions. The results will be used to develop an adoption model and to continuously refine the simulations.

>> www.nemogrid.eu
1 INTRODUCTION

In order to make the decision process autonomous, the system must rely on some sort of distributed intelligence. Here, we describe the main features of the algorithms developed for the different use cases, as well as their communication logics. We recall the considered use-cases:

- Business as usual: loads are locally controlled with decentralized algorithms maximizing the self-consumption.
- DSO-Planned: loads are locally controlled with decentralized algorithms maximizing the self-consumption.
- Voltage-based tariffs: The idea behind voltage tariffs was that the difference between houses’ electric main and the feeder could be used as an indication for local congestions. Thus, tariffs based on voltage can be used to alleviate local congestions on the grid. Unfortunately, this kind of tariffs discriminates users based on their location and are not likely to be implemented in practice in many legislations. We changed this use case to power-based tariffs, that is, a tariff partially based on the power at the household’s main. As the voltage-based tariffs, the final effect is to perform peak shaving. The main difference is that, while the voltage-based tariffs achieve this based on the local state of the grid, the power-based tariffs achieve this apriori.
- P2P market: the aim is to coordinate the users to perform peak shaving, with computing the market equilibrium every 15 minutes. Due to the drop-out of Slack.it from the consortium, which should have developed a secure block chain based trading platform, we switched from a double action based coordination mechanism to one relying on distributed control and game theory, which is guaranteed to find optimal equilibrium points (in the Nash equilibrium sense).

The initial aim of the project was the coordination of electric batteries and thermal loads (boilers and heat-pumps). Due to delays in the porting of the detailed building thermal models developed in WP 3.2 to simpler controllable ones, we postponed the heat pumps control modeling and focused on the control of distributed electric boilers. In the next chapter, the control algorithms for batteries and electric boilers are explained.

2 DISTRIBUTED ARTIFICIAL INTELLIGENCE FOR GRID CONTROL

2.1 DISTRIBUTED CONTROL FOR BUSINESS AS USUAL AND DSO-PLANNED

In the business as usual case, each agent uses its own device to reduce its overall costs, without any kind of coordination. This means that agents equipped with a PV power plant will try to maximize their self-consumption. Both batteries and boilers are controlled through a model predictive control (MPC) approach: at each timestep of the simulation, the controller solves an optimization problem using consumption and production forecasts for the next day-ahead. Once the optimal solution has been found, the algorithms actuate only the first control action, and the procedure is repeated.
Battery control

The battery controller is supposed to be interfaced with the battery energy management system, returning an estimation of the battery’s state of charge and injected and withdrawn power, into and from the battery. In this setting, the battery can be considered as a one state fully observed system and applying the MPC is straightforward.

The formulation of the battery control algorithm we have used is based on the work published in [01]. Here we report the full formulation for the pure economic controller. Called \( u \) the matrix of decision variables, whose first and second columns represent the battery charging and discharging power, respectively, and \( y, \ s_{ch}, \ s_{ds} \) being three auxiliary variables, we seek at solving the following problem:

\[
\begin{align*}
\begin{aligned}
u^*, \ y^* &= \arg\min_{u, y} \|s_{ch}\|^2 + \|s_{ds}\|^2 + \sum_{t=1}^{T} y_i \\
\text{s.t.} \quad x_{t+1} &= Ax_t + Bu_t \\
y &\geq p_b \left(u[1, -1]^T + \hat{p}\right) \\
y &\geq p_s \left(u[1, -1]^T + \hat{p}\right) \\
x &\in [x_{min}, x_{max}] \quad u \in [u_{min}, u_{max}] \\
s_{ch}, s_{ds} &\geq 0 \\
s_{ch} &\geq \hat{p} \quad s_{ds} \geq -\hat{p} \\
u &\geq [s_{ch}, s_{ds}] \\
\end{aligned}
\end{align*}
\]

Here the \( y \) represents the total costs of the prosumer. For prosumers, the cost function can be either positive or negative, depending on the overall power flow at their household’s main. The cost function can be expressed as:

\[
c(z_t) = \begin{cases} 
p_{hi}z_t, & \text{if } z_t \geq 0 \\
p_{si}z_t, & \text{otherwise}
\end{cases}
\]

Where \( z \) stands for the overall power at the mains supply. This can be thought of as the maximum over two affine functions (the first and second line of the above expression, respectively). Equation (3-4) constraint \( y \) to live in the epigraph of this maximum (green region in Fig.1). Minimizing \( y \) then guarantees that \( y^* \) will lie on the epigraph’s lower boundary (and will thus represents the prosumer’s total costs). Equation (2) represents the battery dynamics. For control purposes, we modeled the battery as a one state dynamic system. In this case \( B \) is a two element matrix whose entries are \([\eta_{ch}, 1/\eta_{ds}]\) where \( \eta_{ch} \) and \( \eta_{ds} \) represent the battery charging and discharging efficiency, respectively. This means that the charging and discharging operations must be treated independently. Furthermore, we must explicitly prevent the battery from charging and discharging at the same time. While this is not strictly necessary in the pure economic optimization problem, this can result in wrong solutions when considering
peak shaving, as the battery can charge and discharge simultaneously to maximize valley filling effects, i.e. the battery will try to dissipate excess energy.

Figure 1: Visual explanation of the scope of the $y$ variable. When linearly penalized, $y$ is pushed to its feasible space’s lower borders, collapsing on the cost function $c(p)$.

This would result in a bilinear constraint, which will increase the computational time of the solver. Fortunately, for this problem we can exploit an alternative formulation which won’t introduce nonlinear constraint. For this reason we introduce the $s_{ch}$ and $s_{ds}$ variables. Figure 2 explains the role of $s_{ch}$: the feasible space of $s_{ch}$ is constrained to be the epigraph of the maximum between 0 and the forecasted power at the mains power supply. When $s_{ch}$ is quadratically punished, it will shrink on the lower boundary of the epigraph, (orange line in the second panel of Fig.2). Its optimal value can then be used to define the feasible regions of the battery charging power. The same reasoning can be applied to define the feasible regions for the battery’s discharging power; this will result in two disjoint feasible sets for the charging and discharging powers. These constraints are represented by equations 6, 7 and 8. Finally, equation 5 represents the box constraint for the battery energy and the charging and discharging powers.

Figure 2: Visual explanation of the scope of the $s_{ch}$ variable. When quadratically penalize, it effectively reduces the feasible space for the battery charging power, such that the feasible spaces for the charging and discharging battery powers are disjoint.
Using the same set of constraints, a peak shaving controller has been formulated. A lexicographic variant of the aforementioned method has also been tested in simulations: the batteries are operated using a bilevel MPC. The first MPC tries to maximise the self-consumption, while the second one tries to perform peak shaving, while not worsening the self-consumption reached by the first, more than a given threshold.

**Boiler control**

For the electric boilers, we cannot realistically assume them to be a fully observable systems. In fact, this assumption will require to have several sensors indicating their internal temperatures at different heights of the boilers. In a realistic setting, existing electric boilers has no more than two temperature sensors, used by their internal hysteresis controllers, and this information cannot typically be read from an external controller. Furthermore, consider the following simplistic one state model for the boiler’s thermal dynamics:

\[ cM \frac{dT}{dt} = c\Gamma(T_{l,i} - T_{o,i}) - U(T - T_{ext}) + P_{el,i} \]

Despite its simplicity, this model requires to know the incoming/outgoing water flux \( \Gamma \), which means that a fluximeter must be installed. This is not possible but in pilot projects, since installation costs of these sensors will completely cancel out the economic benefit of an avoided grid refurbishment.

As such, we assume that we can only exploit the electric power measurements for controlling electric boilers. Furthermore, we expect to be able to only turn off the boiler through a relay, and not forcing it on (due to safety reasons, since we do not any feedback ). Given these constraints, the electric boiler’s nominal power and energy needs are estimated using historical data of their power consumption. Then, the algorithm decides when to force off the boiler such that the boiler can always satisfy its energy needs inside 3-hours slots.

1. The main parameters of the boiler (\( P_{\text{nom}}, V \)) are estimated from historical power data.
2. The energy needs of the boiler are forecasted using a LightGBM model taking as input past data of the boiler’s power profile, as well as weather predictions for the next 24 hours. Furthermore, forcing the boiler off could result in an energy rebound effect. This can be corrected by passing to the forecaster also historical values of the control action as a categorical binary variable (since we want to forecast the energy needs of the uncontrolled boiler, this approximately counteracts our action on the system).
3. The algorithm decides when to force off the boiler such that the boiler can always satisfy its energy needs inside 3-hours slots. For example, if a consumption of 2 kWh is forecasted between 18h-21h, and the estimated nominal power is of 4 kW, the boiler can be forced off at most 2h30min during this period.
We based the control algorithm on the work published in [02]. In particular, we formulated it as the following optimization problem:

\[
\begin{align*}
  u^*, y^* = & \arg\min_{u,y} \sum_{t=1}^{T} y_t - \sum_{t=1}^{T} \min(y, 0) \\
  \text{s.t.} & \quad y \geq p_b (\hat{p}_b (1 - u) + \hat{p}) \\
  & \quad y \geq p_s (\hat{p}_b (1 - u) + \hat{p}) \\
  & \quad S[(1 - u) p_{nom} - \hat{p}_b] \geq \gamma \\
  & \quad \sum_{t=1}^{T-1} |\Delta u| \geq n_{ch}
\end{align*}
\]

(10) (11) (12) (13) (14)

Where \( y \) has the same role as in the battery optimization problem, representing the total costs for the prosumer, \( \gamma \) is a slack variable which relax the energy invariance constraint (13). Here \( S \) is a summation matrix which sum the energy in the pre-defined time slots (3 hours) and \( p_{nom} \) is the nominal power of the boiler, inferred by past measurements. Equation (14) further prevents the boiler for being turned on and off more than \( n_{ch} \) times in a control horizon.

### 2.2 Distributed Coordination

Since we are interested in controlling the aggregate power profile of a group of prosumers, introducing communication among them can increase the effectiveness of a control algorithm. In this project, we implemented distributed control algorithms based on distributed control and game theory. In [01], we have tested the design of a market which targets self-consumption communities. Users are assumed not to have any active decision nor belief in the formation of the market equilibrium price. We further avoided to model users’ utility as intended in standard auction and game theory, and replaced it with costs and users’ constraints sets. In fact, the latter can be interpreted as a binary and non differentiable utility function, and prevent us from making any assumption on users’ marginal satisfaction with respect to consumed energy. Briefly speaking, each user aims at minimizing its energy billing costs, while helping to optimize a system-level objective (which is function of the joint actions of all the prosumers). The distributed computation is orchestrated by a central coordinator, which is located at the point of common coupling with the distribution grid. The revenue generated by the system-level objective is then redistributed among the peers, based on statistics on their energy consumption and production. The algorithm presented in [01] has been adapted to both batteries and electric boilers.
REFERENCES
