ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

eeh power systems laboratory

Martin Zellner

Protecting Privacy in Smart Grids via Distributed Optimization and Differential Privacy

Master Thesis
PSL1621

EEH – Power Systems Laboratory
IDSC – Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich

Examiner: Prof. Dr. Gabriela Hug
Supervisors: Prof. Dr. Melanie Zeilinger
Dr. Tomas Tinoco De Rubira

Zurich, November 22, 2016

Abstract

Smart meters enable a variety of new and useful applications for achieving a smart grid. Unfortunately, they also pose privacy issues to consumers. In this work, techniques that exploit the benefits of smart meters and at the same time mitigate privacy issues, are explored. More specifically, we consider the application of controlling local energy storage devices of a group of consumers with the goal of minimizing energy cost and providing ancillary services in the form of aggregate load smoothing to the grid. A differentially private distributed model predictive controller (MPC) is proposed for this application based on a framework described in the literature. By using a distributed optimization algorithm, the energy cost of consumers can be minimized while keeping the load profile of each consumer private. Also, aggregate load smoothing can be achieved by exchanging net consumption information between consumers. Since these exchanges can contain information about the load profiles, the concept of Differential Privacy is used to protect them. The performance of the proposed controller is analyzed. In particular, the trade-off between privacy and performance is studied as well as the effects of having a trustworthy or untrustworthy mediator handling the information exchanges between consumers.

Contents

List of Acronyms

SOC	State of Charge
DProxG	Distributed Proximal Gradient
DProjG	Distributed Projected Gradient
MPC	Model Predictive Control
DP-DProxG	Differentially Private Distributed Proximal Gradient
EU	European Union

Nomenclature

c^b	Prices for buying energy
c^s	Prices for selling energy
D	Difference matrix
e_i	Energy contents of the battery of consumer i
f	Smoothing penalty objective function
h	Composite objective function
l_i	Load profile of consumer i
p_i	Net power consumptions of consumer i
q_i	Powers injected to battery of consumer i
r_i	Renewable generation profile of consumer i
T	Prediction horizon of the MPC
u_i	Uncontrolled part of the net consumptions of consumer i
u	Uncontrolled part of the net consumptions
x_i	Local variables associated with consumer i

Chapter 1

Introduction

Smart Meters are advanced energy meters that can measure the consumption of electrical energy. In contrast to conventional meters, they provide more precise measurements as well as additional information such as real-time data. The detailed data available from the smart meter enables a variety of new and useful applications. One such application is the control of local storage devices for minimizing the energy cost of consumers while providing certain services to the grid, e.g., demand or load smoothing. Nowadays, smart meters are gradually being deployed in more and more markets. The Third Energy Package of the European Union (EU), for example, requires all the member states to install intelligent measurement systems for the long-term benefit of consumers [1]. Between 2000 and 2005, Enel, the third largest energy provider in Europe, deployed smart meters to its 27 million customers [2]. By 2014, there were close to 45 million smart meters already installed in Finland, Italy and Sweden [1]. According to an estimate by the European Commission, there will be an expected installed base of 200 million smart meters by 2020 in the EU [1] . Despite their benefits, a big hurdle for the deployment of intelligent measurement devices are security and privacy concerns. User consumption data can reveal information such as the number of people present at a residence at a given time, and the times during which certain appliances are used [2]. Also, certain activities such as watching TV have power signatures that are detectable to some extend from smart meter data [3]. These privacy concerns have triggered an intense public debate on smart meters. For example, in 2007, the Dutch government decided to make a big smart meter rollout optional instead of mandatory for 7 million residential consumers [4].

The potential privacy issues that arise with the deployment of smart meters have not only caused public debates but also propelled significant research efforts from the academic community. For example, the possibility to detect which movie is watched based on smart meter data is shown in [5]. Examples of the type of private information that can be extracted or inferred from smart meters are described in [6]. Also, an application for detecting individual power signatures from 8 different devices with recognition rates of up to 87 % is described in [7]. Other authors have explored ways

to quantify and mitigate the privacy issues with smart meters while still exploiting their benefits. In particular, information theory has been used for this purpose quite extensively. In [8], for example, an approach for calculating the information leakage with the help of a finite state model is described. In [9], smart meter data is protected by obfuscation and the privacy-utility trade-off is analyzed. In [10], privacy is quantified by mutual information and privacy protection is achieved by distorting the net load profile using an alternative energy source. In [11], the information leakage rate is calculated for the case of batteries with zero and infinite capacity. In addition to information-theoretic techniques, techniques based on Differential Privacy [12] have also received significant attention from the research community. Differential Privacy provides a rigorous measure of privacy and ensures useful properties such as resilience to post-processing. In [13], differential privacy is used to protect privacy when using smart meter data for state estimation in distribution networks. In [14], a battery is used to ensure differential privacy of the consumer load profile. For the case of electric vehicle charging, [15] proposes a differentially private algorithm that uses distributed optimization.

In this thesis, we follow the line of research of [15] and use distributed optimization and differential privacy to optimally operate local storage devices of a group of consumers while protecting their privacy. In particular, we consider a distributed controller that uses smart meter data only locally, i.e., it does not require sharing the load profile of the consumer. For the distributed control two performance objectives are considered: the minimization of energy cost of each consumer and the provision of ancillary services to the grid operator. The specific service considered is the smoothing of the aggregate load profiles of the group of consumers. Smooth load profiles are desirable to the operator as they result in lower operation costs due to a decreased utilization of primary and secondary control. For solving the resulting optimization problem efficiently, an algorithm [16] that exploits properties of problems with a composite and partly decoupled objective is applied. Privacy protection is achieved by adding noise to the data exchanged between the different consumers. The noise has a negative effect on the performance of the algorithm. As a consequence, increasing the level of privacy degrades the performance of the controller resulting in a trade-off between protecting privacy and performance. We discuss this trade-off, comparing the case where the net consumption is kept entirely private to the base case where the mediator can be trusted with the net consumption.

The specific contributions of this thesis are:

- A distributed model predictive controller (MPC) that minimizes energy costs of a group of consumers and provides ancillary services to the grid while protecting consumers privacy.
- A characterization of the trade-off between privacy and performance for controllers with trustworthy and untrustworthy mediators.

- The utilization of a distributed optimization algorithm that is suitable for problems with composite objectives that have separable and non-separable parts.

This thesis is structured as follows: In Chapter 2, the problem of controlling local energy storage is described and formulated mathematically. Chapter 3 proposes an MPC that protects the load profile of the individual consumers using distributed optimization and differential privacy. In Chapter 4, experimental results that characterize the performance of the controller under different conditions as well as the trade-off between protecting privacy and performance are shown. Lastly, in Chapter 5 the key contributions of the thesis are summarized and potential future extensions are discussed.

Chapter 2

Problem Description

The problem considered in this work is the control of local energy storage of a group of consumers with the goal of minimizing energy cost while providing ancillary services to the grid. The specific ancillary service considered is load smoothing. Smooth load profiles are desirable to the grid operator as they reduce the use of primary and secondary control. In order to achieve a smooth aggregate load profile, the individual consumers need to coordinate by exchanging information, which can lead to privacy issues.

2.1 Problem Setting

A discrete-time system composed of N consumers is considered. Each consumer $i \in \{1, \dots, N\}$ has a local storage device, e.g., battery, and a share of consumers also has a renewable energy source, e.g., solar. The consumers are connected to the distribution grid so they can buy energy from the grid and sell excess energy to the grid. We assume the group to have market access so they can benefit from varying energy prices. In the following, the models used to represent batteries, loads, renewable generation, energy cost and ancillary services are described.

The battery charging power of consumer i at time period t is denoted by $q_i(t)$, and the battery state of charge (SOC) at the end of period t is denoted by $e_i(t)$. The battery dynamics are given by

$$q_i(t) = \frac{1}{\Delta t} \left(e_i(t) - e_i(t-1) \right), \tag{2.1}$$

where Δt represents the duration of one time period. The charging power and SOC are limited. This is captured by the following linear bounds:

$$q_i^{\mathsf{min}} \leq q_i(t) \leq q_i^{\mathsf{max}} \tag{2.2a}$$

$$e_i^{\mathsf{min}} \leq e_i(t) \leq e_i^{\mathsf{max}}. \tag{2.2b}$$

It is assumed that each consumer has to pay for the energy that he or she consumes and receives money for the energy produced. The price for all consumers is equal. The price $c^b(t)$ for buying energy is assumed to be higher than the price $c^s(t)$ for selling energy. The net consumption of consumer i is denoted by

$$p_i(t) := l_i(t) - r_i(t) + q_i(t)\,, \tag{2.3}$$

where $l_i(t)$ is the load consumption and $r_i(t)$ is the renewable generation in time period t.
Using this, the cost or revenue associated with the net consumption of each consumer i is given by the following piecewise linear function:

$$g_i(p_i(t), t) := \begin{cases} c^s(t)p_i(t) & \text{for } p_i(t) \geq 0 \\ c^b(t)p_i(t) & \text{for } p_i(t) < 0\,. \end{cases} \tag{2.4}$$

The vector containing the battery state and input of consumer i at time t is denoted by

$$x_i(t) := (q_i(t), e_i(t))\,. \tag{2.5}$$

The vector containing the battery state and input of all consumers at time t is denoted by

$$x(t) := (x_1(t), \ldots, x_N(t))\,. \tag{2.6}$$

Lastly, the measure used to quantify the changes in the aggregate load profile from time $t-1$ to time t is given by

$$f(x(t), x(t-1)) := (p(t) - p(t-1))^2\,, \tag{2.7}$$

where

$$p(t) := \sum_{i=1}^{N} p_i(t) \tag{2.8}$$

is the aggregate net consumption of all consumers. A quadratic cost function was used since high spikes are especially costly for the grid operator and should therefore be avoided.

2.2 Problem Formulation

The goal of the problem is to find the optimal policy π^* from the set of all causal and implementable policies Π for operating the local storage device of each consumer. This policy minimizes energy cost and provides demand smoothing services to the grid.

Using the definitions from the previous section, this problem can be formulated as follows:

$$\pi^* = \underset{\pi\in\Pi}{\operatorname{argmin}} \lim_{T\to\infty} \mathsf{E}\left[\frac{1}{T}\sum_{t=0}^{T}\sum_{i=1}^{N} g_i(p_i(t), t) + \gamma \sum_{t=0}^{T} f(x(t), x(t-1))\right]\,. \tag{2.9}$$

The set of implementable policies contains all policies that fulfill the following constraints for all $i \in \{1, \ldots, N\}$ and t:

$$q_i(t) = \frac{e_i(t) - e_i(t-1)}{\Delta t}\,, \tag{2.10}$$

$$q_i^{\mathsf{min}} \leq q_i(t) \leq q_i^{\mathsf{max}}\,, \tag{2.11}$$

$$e_i^{\mathsf{min}} \leq e_i(t) \leq e_i^{\mathsf{max}}\,. \tag{2.12}$$

The output of the control policy π, which in general depends only on current and past observations, are the charging powers of all batteries at each time t, *i.e.*,

$$q(t) = \pi(x(t-1), \ldots x(0), l(t), \ldots, l(0), r(t), \ldots, r(0))\,,$$

where l is the vector containing the load, and r is the vector containing the renewable generation for all consumers.

Chapter 3

Solution Approach

In the following, a model predictive controller designed to solve the energy storage control problem described in Chapter 2 is presented. First, the MPC approach is described, including the optimization subproblems that are solved at each time. Then, two distributed algorithms for solving the MPC subproblems are discussed, one that has been used in literature for solving similar problems, and another that exploits the composite structure of the objective. The information exchanges required by these algorithms can potentially contain privacy-sensitive data. In order to protect the consumer during these exchanges, Differential Privacy is introduced and its incorporation into the controller is described.

3.1 Model Predictive Control

In order to solve the energy storage control problem shown in (2.9), a model predictive control approach can be used. The approach consists of solving an optimization subproblem at each time t to obtain a control policy for a finite prediction horizon $\{t, \dots, t+T\}$. Then, only the current action is applied to the system and the process is repeated in a receding horizon fashion.

At time t, the MPC subproblem for the energy storage problem is

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & h(x) && \text{(3.1a)} \\ \text{subject to} \quad & A_i x_i = b_i\,, \quad i = 1, \dots, N\,, && \text{(3.1b)} \\ & x_i^{\min} \leq x_i \leq x_i^{\max}\,, \quad i = 1, \dots, N\,, && \text{(3.1c)} \end{aligned}$$

where x is the vector containing the battery power and SOC for all consumers and times in the prediction horizon, and x_i is given by

$$x_i := (x_i(t), \dots, x_i(t+T))$$

for each consumer i. The objective function h is given by the sum of a smoothing component $\bar{f}$, weighted by a parameter γ, and energy cost components $\bar{g}_i$ for each

consumer:

$$h(x) = \sum_{i=1}^{N} \underbrace{\sum_{\tau=t}^{t+T} g_i\left(p_i(\tau), \tau\right)}_{\bar{g}_i(x_i)} + \underbrace{\gamma \sum_{\tau=t}^{t+T} f(x(\tau), x(\tau-1))}_{\bar{f}(x)} . \tag{3.2}$$

Constraint (3.1b) captures the battery dynamics as defined in (2.1), and constraint (3.1c) captures the battery limits as defined in (2.2).

3.2 Distributed Subproblem Solution

In the following, two distributed algorithms for solving the MPC subproblem (3.1) are described. Specifically, the Distributed Projected Gradient algorithm (DProjG) and the Distributed Proximal Gradient algorithm (DProxG) are shown. The former has been used in the literature for problems related to electric vehicle charging [17] [15] that have a similar structure as that of (3.1). The latter is proposed here for solving subproblem (3.1) since it can exploit the composite structure of the objective function (3.2), namely, that part of it is separable in the variables associated with each consumer, and does not require communicating the load profile directly. This algorithm is related to the Forward-Backward Splitting algorithm proposed in [16].

DProjG consists of a gradient step with respect to the complete objective, followed by distributed projections onto the feasible set of the consumers, namely,

$$\mathcal{C}_i := \left\{ x_i \,\middle|\, A_i x_i = b_i,\, x_i^{\min} \leq x_i \leq x_i^{\max} \right\}, i \in \{1, \ldots, N\}.$$

This algorithm is formally stated below:

Algorithm 1. *Distributed Projected Gradient Algorithm [17]*

- *Gradient Step*

$$\tilde{x}^{(k)} = x^{(k)} - \alpha_k \xi^{(k)} \tag{3.3a}$$

- *Projection Step (can be done in parallel)*

$$x_i^{(k+1)} = \Pi_{\mathcal{C}_i}\left(\tilde{x}_i^{(k)}\right) \tag{3.3b}$$

for each i, where $\xi^{(k)}$ is in the subdifferential $\delta h(x^{(k)})$ of h at point $x^{(k)}$, $\Pi_{\mathcal{C}_i}$ denotes the projection operator defined by

$$\Pi_{\mathcal{C}_i}(y) := \underset{z \in \mathcal{C}_i}{\operatorname{argmin}} \left(\frac{1}{2} \|z - y\|_2^2 \right) , \tag{3.3c}$$

and $\|\cdot\|_2$ is the Euclidean norm.

The step sizes α_k have to fulfill the following conditions:

$$\sum_k \alpha_k = \infty\,, \quad \lim_{k\to\infty} \alpha_k = 0\,.$$

By including the energy cost part of the objective function into the local projection, one can obtain DProxG as proposed in [16] [18]. Formally, DProxG is given as follows:

Algorithm 2. *Distributed Proximal Gradient Algorithm [16] [18]*

- *Gradient Step*
$$\tilde{x}^{(k)} = x^{(k)} - \alpha_k \nabla \bar{f}(x^{(k)}) \tag{3.4a}$$
- *Proximal Step (can be done in parallel)*
$$\bar{x}_i^{(k+1)} = \quad \text{prox}_{\alpha_k \bar{g}_i}(\tilde{x}_i^{(k)}) \tag{3.4b}$$
for each i, where $\text{prox}_{\alpha_k \bar{g}_i}$ is the proximal operator [19] defined by
$$\text{prox}_{\alpha_k \bar{g}_i}(y) = \underset{z\in\mathcal{C}_i}{\text{argmin}} \left(\alpha_k \bar{g}_i(z) + \frac{1}{2}\|z-y\|_2^2 \right)\,. \tag{3.4c}$$

The step sizes α_k have to fulfill the following conditions:

$$\sum_k \alpha_k = \infty\,, \quad \sum_k \alpha_k^2 < \infty\,.$$

For a nonlinear cost function such as the one defined in (2.4), using DProxG can be beneficial in terms of performance, as it handles the cost function inside the proximal operator rather than by a (less effective) gradient step. If the cost function is linear, e.g., the prices for buying and selling energy are equal, it can be shown that DProjG and DProxG are equivalent. The following Lemma formalizes this fact.

Lemma 1. *If $\bar{g}_i$ is linear for each i, DProjG and DProxG are equivalent when applied to problem (3.1).*

Proof. For the MPC subproblem (3.1) the DProjG update for each i is

$$x_i^{(k+1)} = \Pi_{\mathcal{C}_i}\left(x_i^{(k)} - \alpha_k \left(\nabla_{x_i} \bar{f}(x^{(k)}) + \nabla_{x_i} \bar{g}_i(x_i^{(k)}) \right) \right)\,. \tag{3.5}$$

On the other hand, the update for DProxG is

$$x_i^{(k+1)} := \text{prox}_{\eta_k \bar{g}_i}\big(\underbrace{x_i^{(k)} - \eta_k \nabla_{x_i} \bar{f}(x^{(k)})}_{y} \big)\,. \tag{3.6}$$

Let ξ_i be the gradient of $\bar{g}_i$ at point $x_i^{(k)}$. Note that to simplify notation, the iteration superscript is dropped in the following. To show equivalence of DProxG and DProjG

for a linear $\bar{g}_i$, the optimization problem solved by the proximal operator can be reformulated as:

$$\begin{aligned}
\text{prox}_{\eta_k \bar{g}_i}(y) &= \underset{z \in \mathcal{C}_i}{\text{argmin}} \left(\frac{1}{2} \|z - y\|_2^2 + \eta_k \xi_i^\top z \right) \\
&= \underset{z \in \mathcal{C}_i}{\text{argmin}} \left(\frac{1}{2} z^\top z - y^\top z + \frac{1}{2} y^\top y + \eta_k \xi_i^\top z \right) \\
&= \underset{z \in \mathcal{C}_i}{\text{argmin}} \left(\frac{1}{2} z^\top z - (y - \eta_k \xi_i)^\top z + \frac{1}{2} y^\top y \right) \\
&= \underset{z \in \mathcal{C}_i}{\text{argmin}} \left(\|z - (y - \eta_k \xi_i)\|_2^2 \right) \\
&= \Pi_{\mathcal{C}_i} \left(y - \eta_k \xi_i \right) \\
&= \Pi_{\mathcal{C}_i} \left(x_i^{(k)} - \eta_k \nabla_{x_i} \bar{f}(x^{(k)}) - \eta \xi_i \right) \\
&= \Pi_{\mathcal{C}_i} \left(x_i^{(k)} - \eta_k \left(\nabla_{x_i} \bar{f}(x^{(k)}) + \xi_i \right) \right) ,
\end{aligned}$$

which is equal to (3.5) if $\eta_k = \alpha_k$ for all k.

□

In DProxG, the coordination between the consumers works as follows: In each iteration of the algorithm, the individual consumers share their net consumption estimates with a trusted mediator. The mediator uses these inputs to compute the gradient $\nabla_{x_i} f(x^{(k)})$ of the coupling objective, which it then broadcasts to all consumers as it is equal for all of them. A detailed derivation of the gradient is shown in A.1 of the appendix. Using the gradient, the individual consumers can perform a proximal step to update their local variables for the next iteration. In contrast to solving the problem in a centralized manner, the information exchanges of DProxG only require sharing the net consumption estimates and not the load profile. Also DProjG requires sharing only the net consumption estimates but it does not exploit the fact that information exchanges are not needed for minimizing energy cost. Since DProxG exploits this and is also superior in terms of performance, it is used in the rest of this work as the algorithm of choice. The coordination between the different consumers is illustrated in Figure 3.1.

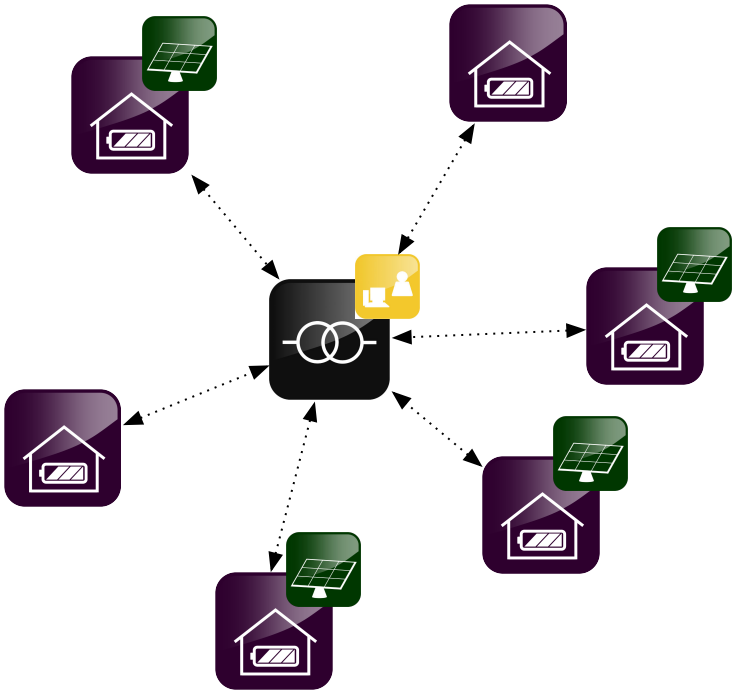

Figure 3.1: Coordination between the different consumers.

3.3 Differential Privacy

In Section 3.2, the DProxG algorithm was described for using load profile data of each consumer locally to minimize energy cost. In order to achieve smoothing, this algorithm requires information exchanges between the different consumers in the form of the net consumption estimates, which can contain privacy-sensitive data. In order to protect privacy during theses exchanges, the concept of *Differential Privacy* proposed in [20] is applied. Differential Privacy enables learning facts about a dataset while at the same time preserving the privacy of every single data entry. Furthermore, it has useful properties such as resilience to post-processing. Privacy is achieved by adding noise to queries executed on a database containing the privacy-sensitive data. The noise added typically has Laplacian or Gaussian distribution [20].

3.3.1 Definitions

In the following, Differential Privacy is formally defined, and some useful properties that are used later to obtain a differentially private MPC are described. To apply the concept of differential privacy to our problem with a trustworthy mediator, the load profiles of the consumers are treated as the contents of a privacy-sensitive database, while the information exchanges of the distributed algorithm are treated as queries on this database.

Definition 1. *Database*
A database $D \in \mathbb{R}^{NT}$ *is defined as the set of privacy-sensitive load consumptions* $l_i := (l_i(t), \dots, l_i(t+T))$ *of each consumer* $i \in \{1, \dots, N\}$ *during an MPC prediction horizon:* $D = \{l_i\}_{i=1}^{N}$.

Differential Privacy provides the guarantee of ϵ-indistinguishably of adjacent databases, therefore an adjacency relation for databases needs to be defined. For the given problem, two databases are considered adjacent if and only if at most one load profile in the database changes. Our definition also requires the change to be bounded by a certain factor δ. More formally, this can be defined as:

Definition 2. *Adjacency Relation*
Databases D *and* D' *are adjacent if and only if there exists an* $i \in \{1, \dots, N\}$ *such that* $\|l_i - l_i'\|_1 \leq \delta$ *and* $l_j = l_j'$ *for all* $j \neq i$*, where* δ *is a preselected positive scalar.*

Using Definition 1 and 2, Differential Privacy can be defined as:

Definition 3. *Differential Privacy [20]*
A randomized algorithm $\mathcal{M}$ *is* ϵ*-differentially private if for all outputs in the set of possible outputs* S*, and for all* (D, D') *that are adjacent, the probabilities are bounded by*

$$\Pr[\mathcal{M}(D') \in S] \leq \exp(\epsilon) \Pr[\mathcal{M}(D) \in S] \,.$$

A useful property of differential privacy is that for algorithms involving multiple queries, the epsilons "add up" as stated by the General Composition Theorem given in [20]:

Theorem 1. *General Composition Theorem [20]*
Let $T_1 : D \to T_1(D)$ *be* ϵ*-differentially private, and for* $k \geq 2$, $T_k : (D, s_1, \dots, s_{k-1}) \to T_k(D, s_1, \dots, s_{k-1}) \in \mathcal{C}_k$ *be* ϵ*-differentially private, for all given tuples of differentially private signals* $(s_1, \dots, s_{k-1}) \in \Pi_{j=1}^{k-1} \mathcal{C}_j$*. Then, for all adjacent* D*,* D' *and all* $S \subseteq \Pi_{j=1}^{k} \mathcal{C}_j$*,*

$$P((T_1, \dots, T_k) \in S) \leq e^{k\epsilon} P'((T_1, \dots, T_k) \in S) \,,$$

where P' *denotes the probability when* D' *is used.*

3.3.2 Differentially Private MPC

Using the definitions from the previous section, differential privacy can be applied to obtain an ϵ-differentially private version of the MPC described in Section 3.1. In order to do that, the trusted mediator adds noise to the information exchanges using the Laplace mechanism as proposed in [20]. The Laplace Distribution and the Laplace Mechanism are formally stated below:

Definition 4. *Laplace Distribution [20]*
The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function

$$\mathrm{Lap}(x|b) = \frac{1}{2b} \exp\left(\frac{|x|}{b}\right) .$$

Definition 5. *Laplace Mechanism [20]*
Given any function $\varphi : \mathbb{R}^n \to \mathbb{R}^m$, the Laplace mechanism is defined as:

$$\mathcal{M}_m(D, \varphi, \epsilon) = \varphi(D) + (Y_1, \ldots, Y_m) ,$$

where Y_i are i.i.d random variables drawn from $\mathrm{Lap}(x|\Delta/\epsilon)$ and Δ is the sensitivity of φ with respect to changes between adjacent databases.

Theorem 2. *Differential Privacy of the Laplace mechanism [20]*
The Laplace mechanism preserves ϵ-differential privacy.

We consider two different cases: One where the mediator can be trusted with the net consumption estimates, and one where the mediator cannot be trusted. In the former case, the net consumption estimates are shared with the mediator, who then creates a differentially private version of the gradient of $\bar{f}$ and broadcasts it to the consumers. In the latter case, the net consumption estimates of the individual consumers need to be differentially private signals. In contrast to the trusted mediator case, the consumers add noise to their net consumption estimates before sharing them with the mediator. Using these differentially private signals, the mediator then computes the gradient of the coupling objective. The two cases are illustrated in Figures 3.2 and 3.3, where the bold lines denote differentially private signals and the dashed lines denote unprotected privacy-sensitive signals.

The mechanism takes the gradient $\nabla_{x_i} \bar{f}(x^{(k)})$ (for the trustworthy mediator case), or the net consumption estimates of the different consumers (for the untrustworthy mediator case) and creates an ϵ_k-differentially private version of these signals. The privacy level ϵ_k depends on the scale of the noise added and the sensitivity $\Delta^{(k)}$ of these signals which will be derived in Section 3.3.3. In the untrustworthy case, the gradient is also differentially-private due to DP's resilience to post processing. Due to Theorem 1, the privacy levels ϵ_k "add up" for multiple queries of the protected signals. This captures the fact that we "loose" an ϵ_k level of privacy in every iteration as more and more information gets released. Therefore, for K iterations, a privacy level of $\sum_{k=1}^{K} \epsilon_k$ is achieved. It follows from this and Theorem 2 that in order to make the algorithm ϵ-differentially private when executed for K iterations, the scale of the noise added by the Laplace mechanism can be set to satisfy

$$b = \frac{\Delta^{(k)}}{\epsilon_k} = \frac{K\Delta^{(k)}}{\epsilon} . \tag{3.7}$$

The derivation of Δ is key and will be carried out in Section 3.3.3.

Since determining the optimal K a priori is hard, in practice K is set to a fixed number of iterations. Using the differentially private versions of the signals, we can now formulate a differentially private version of DProxG which we call DP-DProxG:

Algorithm 3. *DP-DProxG*

- *Initialize* $\left\{x_i^{(1)}\right\}_{i=1}^N$
- *Trustworthy Mediator Case*
  - *Consumers send* $p_i^{(k)} = l_i - r_i + q_i^{(k)}$.
  - *Mediator forms and broadcasts* $\hat{g}^{(k)} := \nabla_{x_i} \bar{f}(x^{(k)}) + w^{(k)}$, *where* $w^{(k)} \sim \mathrm{Lap}(x|b^{(k)})$.
- *Untrustworthy Mediator Case*
  - *Consumers send* $\hat{p}_i^{(k)} := p_i^{(k)} + w_i^{(k)}$, *where* $w_i^{(k)} \sim \mathrm{Lap}(x|b^{(k)})$.
  - *Mediator forms and broadcasts* $\hat{g}^{(k)} := \nabla_{x_i} \bar{f}(\hat{p}^{(k)})$.
- *Each consumer updates*

$$\hat{x}_i^{(k+1)} = \mathrm{prox}_{\alpha_k \bar{g}}\left(x_i^{(k)} - \alpha_k \hat{g}^{(k)}\right) \tag{3.8}$$

$$x_i^{(k+1)} = (1-\theta)x_i^{(k)} + \theta \hat{x}_i^{(k+1)} \tag{3.9}$$

  for each i, *where* $\theta \in [0,1]$.

The acceleration step (3.9) is added to improve the convergence of the algorithm in the given stochastic setting. DP-DProxG provides ϵ-DP which follows from the General Composition Theorem (Theorem 1) and Theorem 2.

3.3.3 Sensitivity Analysis

In order to determine the correct amount of noise to be added to the query outputs, the sensitivity $\Delta^{(k)}$ of the queries is needed. For the given problem with a trustworthy mediator, this is the sensitivity of the gradient with respect to changes between adjacent databases of load profiles. If the mediator is not trusted, the sensitivity of the load profiles and the local variables needs to be computed.

As shown in Appendix A.1, the gradient of $\bar{f}$ with respect to x_i can be written as

$$\nabla_{x_i} \bar{f}(x) = 2\gamma P_q^\top \tilde{D} \left(\sum_{j=1}^{N} (u_j + P_q x_j) \right) , \tag{3.10}$$

where $u_j = (l_j(t) - r_j(t), \ldots, l_j(t+T) - r_j(t+T))$.

In the following, the ℓ_1 sensitivity $\Delta^{(k)}$ of the gradient (3.10) is derived. We begin by showing an important property of the proximal operator:

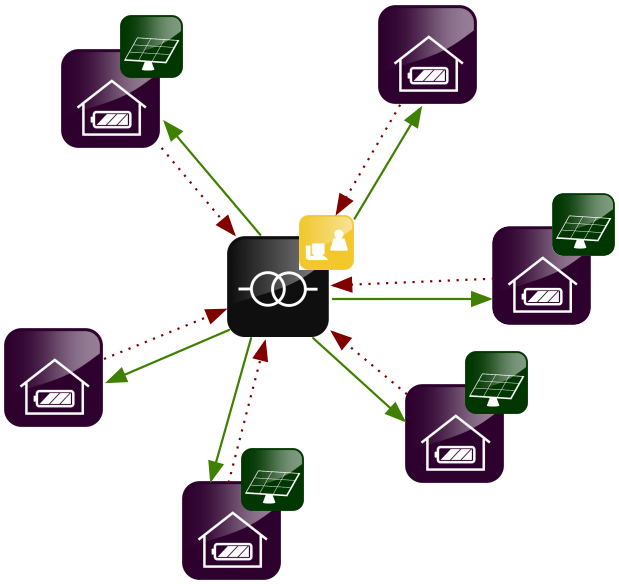

Figure 3.2: Communication for the case of a trustworthy mediator.

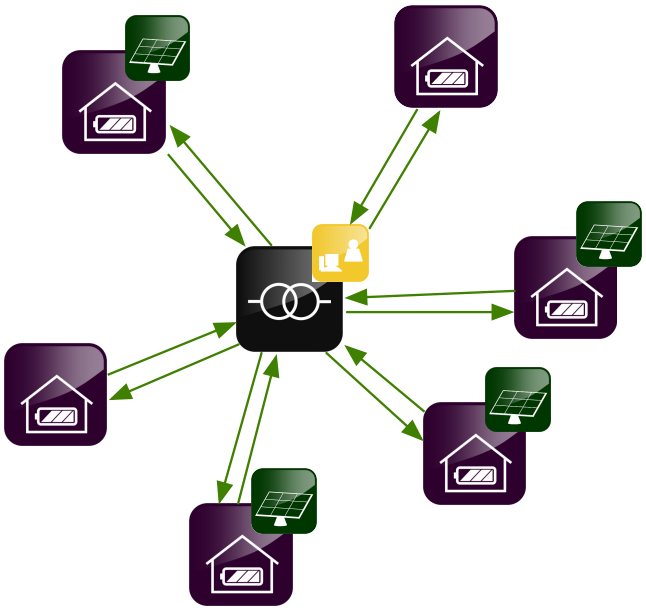

Figure 3.3: Communication for the case of an untrustworthy mediator.

Lemma 2. *The proximal operator is firmly non-expansive, i.e.,*

$$\|\mathrm{prox}_f(x) - \mathrm{prox}_f(y)\| \leq \|x - y\| \,. \tag{3.11}$$

Proof. Let x and y be two arbitrary points, $\bar{x} := \mathrm{prox}_f(x)$, $\bar{y} := \mathrm{prox}_f(y)$ and let ζ_x denote a subgradient of f at a point x. The optimality conditions for $\mathrm{prox}_f(x)$ and $\mathrm{prox}_f(y)$ give

$$(\zeta_{\bar{x}} + \bar{x} - x)^\top (\bar{y} - \bar{x}) \geq 0 \tag{3.12}$$
$$(\zeta_{\bar{y}} + \bar{y} - y)^\top (\bar{x} - \bar{y}) \geq 0 \,. \tag{3.13}$$

From the properties of the subgradients $\zeta_{\bar{x}}$ and $\zeta_{\bar{y}}$, it holds that

$$f(\bar{y}) - f(\bar{x}) \geq \zeta_{\bar{x}}^\top (\bar{y} - \bar{x}) \tag{3.14}$$
$$f(\bar{x}) - f(\bar{y}) \geq \zeta_{\bar{y}}^\top (\bar{x} - \bar{y}) \,. \tag{3.15}$$

Adding (3.14) and (3.15) gives

$$0 \geq (\zeta_{\bar{x}} - \zeta_{\bar{y}})^\top (\bar{y} - \bar{x}) \,. \tag{3.16}$$

Furthermore, adding (3.12) and (3.13) gives

$$((x - y) + (\zeta_{\bar{y}} - \zeta_{\bar{x}}) + (\bar{y} - \bar{x}))^\top (\bar{y} - \bar{x}) \leq 0 \,.$$

From this it follows that

$$(\bar{y} - \bar{x})^\top (\bar{y} - \bar{x}) \leq (y - x)^\top (\bar{y} - \bar{x}) + (\zeta_{\bar{x}} - \zeta_{\bar{y}})^\top (\bar{y} - \bar{x}) \,. \tag{3.17}$$

According to (3.16), the last term of (3.17) is bounded above by zero, therefore (3.17) can be simplified to

$$(\bar{y} - \bar{x})^\top (\bar{y} - \bar{x}) \leq (y - x)^\top (\bar{y} - \bar{x}) \,. \tag{3.18}$$

Using the Cauchy-Schwarz inequality leads to

$$\|\bar{y} - \bar{x}\|^2 \leq \|y - x\| \|\bar{y} - \bar{x}\| \,, \tag{3.19}$$

which is equivalent to

$$\|\mathrm{prox}_f(y) - \mathrm{prox}_f(x)\| \leq \|y - x\| \,. \tag{3.20}$$

□

To derive the sensitivity, let D and D' be adjacent databases according to Definition 2. Also, for the given problem, the local constraints are independent on the database, i.e., $\mathcal{C}_j = \mathcal{C}'_j$, for all $j \in \{1, \ldots, N\}$.

Lemma 3. *Sensitivity of the iterate*
When $\hat{g}(x^{(1)}), \hat{g}(x^{(2)}), \ldots, \hat{g}(x^{(k-1)})$ are given, the ℓ_1-sensitivity of the iterate $x_i^{(k)}$ with respect to changes in the database is zero:

$$\|x_i^{(k)}(D') - x_i^{(k)}(D)\| = 0\,. \tag{3.21}$$

Proof. For $k = 1$ it holds that

$$\|x_i^{(k)}(D') - x_i^{(k)}(D)\| = 0\,, \tag{3.22}$$

because the initial point is arbitrary and therefore independent of the database. Now consider the case $k > 1$ and assume that (3.21) holds for all $j \in \{1, \ldots, k-1\}$. For notational convenience, let

$$v_i^{(k)}(D) := x_i^{(k)}(D) - \alpha_k \nabla_{x_i} \bar{f}\left(x^{(k)}(D)\right)\,.$$

From Lemma 2 it follows that

$$\begin{aligned}\|x_i^{(k)}(D') - x_i^{(k)}(D)\| &= \left\|\operatorname{prox}_{\alpha_{k-1}\bar{g}_i}\left(v_i^{(k-1)}(D')\right) - \operatorname{prox}_{\alpha_{k-1}\bar{g}_i}\left(v_i^{(k-1)}(D)\right)\right\| \\ &\leq \|v_i^{(k-1)}(D') - v_i^{(k-1)}(D)\|\,.\end{aligned}$$

The fact that $\nabla_{x_i}\bar{f}(x^{(k-1)})$ is given implies that

$$\|v_i^{(k-1)}(D') - v_i^{(k-1)}(D)\| = \|x_i^{(k-1)}(D') - x_i^{(k-1)}(D)\| \tag{3.23}$$

holds. The inductive hypothesis then gives that the right hand side of (3.23) is zero. □

Theorem 3. *Sensitivity of the gradient*
In the trustworthy mediator case, when $\hat{g}(x^{(1)}), \hat{g}(x^{(2)}), \ldots, \hat{g}(x^{(k-1)})$ are given, the ℓ_1 sensitivity of the gradient $\nabla_{x_i}\bar{f}$ at point $x^{(k)}$ is $\Delta^{(k)} = 2\gamma\|P_q^\top \tilde{D}\|_1 \delta$, where δ is defined in Definition 2.

Proof. The sensitivity of the gradient $\Delta^{(k)}$ can be bounded using the sensitivity of the load and the sensitivity of the iterate as derived in Lemma 3:

$$\begin{aligned}&\|\nabla_{x_i}\bar{f}(x^{(k)}(D)) - \nabla_{x_i}\bar{f}(x^{(k)}(D'))\|_1 \\ &= \left\|2\gamma P_q^\top \tilde{D}\sum_{j=1}^{N}\left(u_j(D) + P_q x_j^{(k)}(D)\right) - 2\gamma P_q^\top \tilde{D}\sum_{j=1}^{N}\left(u_j(D') + P_q x_j^{(k)}(D')\right)\right\| \\ &= \Bigg\|2\gamma P_q^\top \tilde{D}\sum_{j=1}^{N}\left(l_j(D) - r_j + P_q x_j^{(k)}(D)\right) \\ &- 2\gamma P_q^\top \tilde{D}\sum_{j=1}^{N}\left(l_j(D') - r_j + P_q x_j^{(k)}(D')\right)\Bigg\|\end{aligned}$$

$$
\begin{aligned}
&= \left\| 2\gamma P_q^\top \tilde{D} \sum_{j=1}^{N} \left(l_j(D) - l_j(D') \right) + 2\gamma \tilde{P}_q^\top D P_q \sum_{j=1}^{N} \left(x_j^{(k)}(D) - x_j^{(k)}(D') \right) \right\|_1 \\
&= 2\gamma \| P_q^\top \tilde{D} \|_1 \| l_i(D) - l_i(D') \|_1 \\
&\leq 2\gamma \| P_q^\top \tilde{D} \|_1 \delta \, .
\end{aligned}
$$

Note that the sensitivity with respect to the gradient is equal for all k and depends on the load smoothing factor γ and the database adjacency relation. □

If the mediator is not trusted with the net consumption p_i, it must be treated as a (privacy-sensitive) public signal that needs to be protected.

Theorem 4. *In the untrustworthy mediator case, when the query outputs $p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(k-1)}$ are given, the ℓ_1-sensitivity of $p_i^{(k)}$ is $\Delta^{(k)} = \delta$.*

Proof. The sensitivity $\Delta_i^{(k)}$ of $p_i^{(k)}$ is dependent on the sensitivity of the iterate and the load profile:

$$
\| p_i^{(k)}(D) - p_i^{(k)}(D') \|_1 \leq \left\| x_i^{(k)}(D) - x_i^{(k)}(D') \right\|_1 + \| l_i^{(k)}(D) - l_i^{(k)}(D') \|_1 \, .
$$

The sensitivity of the load profile $l_i^{(k)}$ is given by the adjacency definition, *i.e.*,

$$
\| l_i^{(k)}(D) - l_i^{(k)}(D') \| \leq \delta
$$

. Using this and Lemma 3, the overall sensitivity of the public signal $p_i^{(k)}$ is

$$
\| p_i^{(k)}(D) - p_i^{(k)}(D') \| \leq \delta
$$

. □

By using the concept of DP, these results provide a distributed MPC algorithm that ensures privacy of information exchanges required to solve a composite objective including a coupling term.

Noise level of the different cases

By comparing the two cases, one can verify that the noise added for the untrustworthy mediator case has higher variance compared to that for the trustworthy mediator case. For the untrustworthy mediator, the noise added to $s_i^{(k)}$ is denoted by $(\omega_x^{(k)}, \omega_l^{(k)})$. It can be shown that the resulting differentially private gradient is given by

$$
\nabla_{x_i} \hat{f}(x^{(k)}) = \nabla_{x_i} \bar{f}(x^{(k)}) + \underbrace{2\gamma P_q^\top \tilde{D} \sum_{i=1}^{N} \left(\omega_l^{(k)} + P_q \omega_x^{(k)} \right)}_{\omega_m^{(k)}} \, .
$$

Since $l^{(k)}$ and $x^{(k)}$ are differentially private signals protected by the Laplace mechanism they follow a Laplace distribution with the scale $\frac{K\delta}{\epsilon}$. Using properties of the variance and the fact that $\tilde{D}$ takes differences of uncorrelated random variables, the (element-wise) variance of the resulting noise $\omega_m^{(k)}$ added to the gradient in the untrustworthy mediator case is

$$\begin{aligned}
\mathsf{Var}(\omega_m^{(k)}) &= \mathsf{Var}\left(2\gamma P_q^\top \sum_{j=1}^{N}\left(\tilde{D}\omega_l^{(k)} + P_q^\top \tilde{D} P_q \omega_x^{(k)}\right)\right) \\
&= 4\gamma^2 N\, \mathsf{Var}\left(P_q^\top \tilde{D}\omega_l^{(k)} + P_q^\top \tilde{D} P_q \omega_x^{(k)}\right) \\
&= 4\gamma^2 N\left(2\,\mathsf{Var}(\omega_l^{(k)}) + 2\,\mathsf{Var}(\omega_x^{(k)})\right) \\
&= 8\gamma^2 N\left(2\frac{K^2\Delta^2}{\epsilon^2} + 2\frac{K^2\Delta^2}{\epsilon^2}\right) \\
&= 32\gamma^2 N\frac{K^2\Delta^2}{\epsilon^2}\,,
\end{aligned}$$

where $\Delta = \delta$ as shown in Theorem 4. On the other hand, using Theorem 3, the (element-wise) variance of the noise $\omega^{(k)}$ added to the gradient by trustworthy mediator case is

$$\begin{aligned}
\mathsf{Var}(\omega^{(k)}) &= 2\frac{K^2\Delta^2}{\epsilon^2} = 2\left(2\gamma\|P_q^\top \tilde{D}\|_1\delta\right)^2\frac{K^2}{\epsilon^2} \\
&= 32\gamma^2\frac{K^2\delta^2}{\epsilon^2}\,,
\end{aligned}$$

where $\|P_q^\top \tilde{D}\|_1 = 2$ due to the structure of $\tilde{D}$ and P_q. From the above it can be concluded that the effective noise on the gradient in the untrustworthy case is N times higher compared to the trustworthy mediator case.

Chapter 4

Numerical Experiments

The controller described in Chapter 3 was implemented and evaluated on a test case. In the following, details about the implementation and the test case are provided. Furthermore, experimental results that asses the effect of load smoothing, the performance of the different distributed algorithms, the trade-off between privacy and performance, and the performance of the MPC, are shown.

4.1 Implementation

The controller was implemented in Python using a variety of different tools and frameworks. The modeling of the group of consumers was done with the software package `PFNET` [21], which was extended by the package `MP-PFNET` [22] in this thesis to support multiple time periods. For evaluating the proximal and projection operators, the `CVXPY` modeling framework [23] was used together with the commercial solver `ECOS` [24]. For the development and simulations, a `SciPy` [25] toolchain using Jupyter notebooks was used together with `ipyparallel` [26] for accessing the ETHZ high-performance computing cluster.

4.2 Test Case

As a test case for the experiments, a group of 30 consumers was considered. All consumers were equipped with a battery according to the technical specifications shown in Table 4.1.

The consumers were all assumed to be residential. To generate individual realistic load profiles, the open source load profile generator `ALPG` proposed in [27] was used with the consumer types shown in Table 4.2. Figure 4.1 shows the generated load profiles of the individual consumers and the aggregate load profile for the simulation horizon of one day.

In the case considered, a total of 12 consumers were equipped with renewable generation in the form of solar. To model an accurate solar profile, the `PVWatts` solar

Parameter	Value
$q_i^{\min}$	-50 kW
$q_i^{\max}$	50 kW
$e_i^{\min}$	0 kWh
$e_i^{\max}$	2 kWh

Table 4.1: Battery specifications.

Type	Quantity
single workers	10
dual workers (part-time)	5
dual retired	5
families	10

Table 4.2: Consumer types.

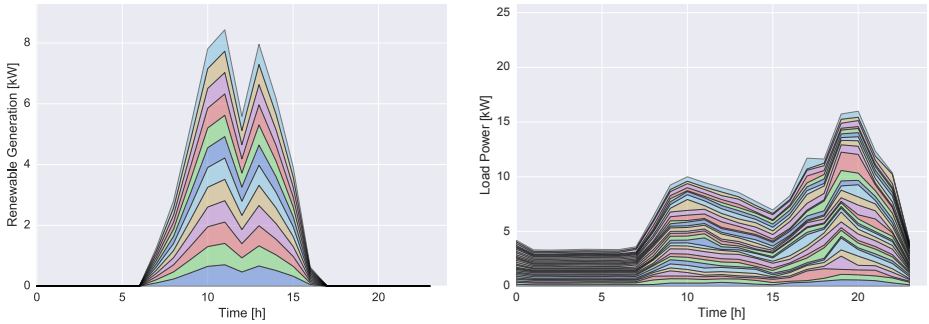

Figure 4.1: Aggregate load and solar profiles.

profile generator [28] was used with the input parameters shown in Table 4.3. The resulting solar generation profile is shown in Figure 4.1.

Parameter	Value
Tilt	30 deg
Azimuth	190 deg
DC System Size	1 kW
Module Type	Standard
Array Type	Open Rack
System Losses	0%

Table 4.3: Renewable generator specifications.

Also, all consumers were assumed to have market access and could therefore buy or sell energy to the grid at a varying price. For the prices, values from European Energy Exchange for the year 2010 were used. The energy pice realization during the simulation period is shown in Figure 4.2.

4.3 Load Profile Smoothing

As described in Chapter 2, one of the objectives of the controller is to provide smoothing to the aggregate load profile of the group of consumers. To show the effect of

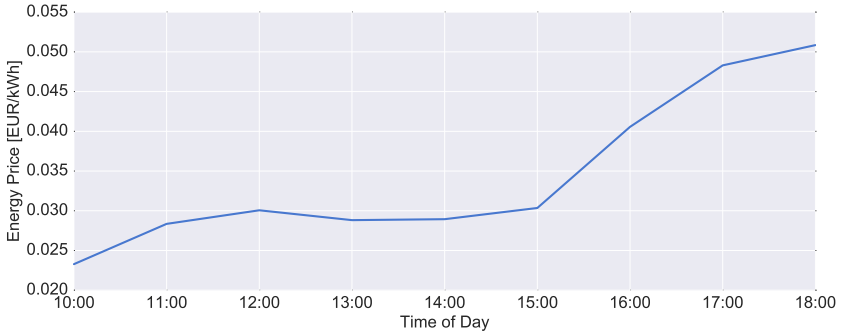

Figure 4.2: Energy buying price c^b used for the numeric experiments.

smoothing in the given test case, the optimization subproblem described in (3.1) was solved in a centralized manner for a fixed time horizon using the solver `ECOS`. The experimental results obtained showing the net consumption with and without the smoothing objective are shown in Figure 4.3 for the input parameters stated in Table 4.4. From the plots, it can be seen that when smoothing is not considered, the battery charging pattern is purely dependent on the energy price. The battery is charged when the price is low and discharged when the price is high. When smoothing is added to the objective with an ancillary service parameter γ of 100, the battery is mainly used to make the aggregate consumption smooth.

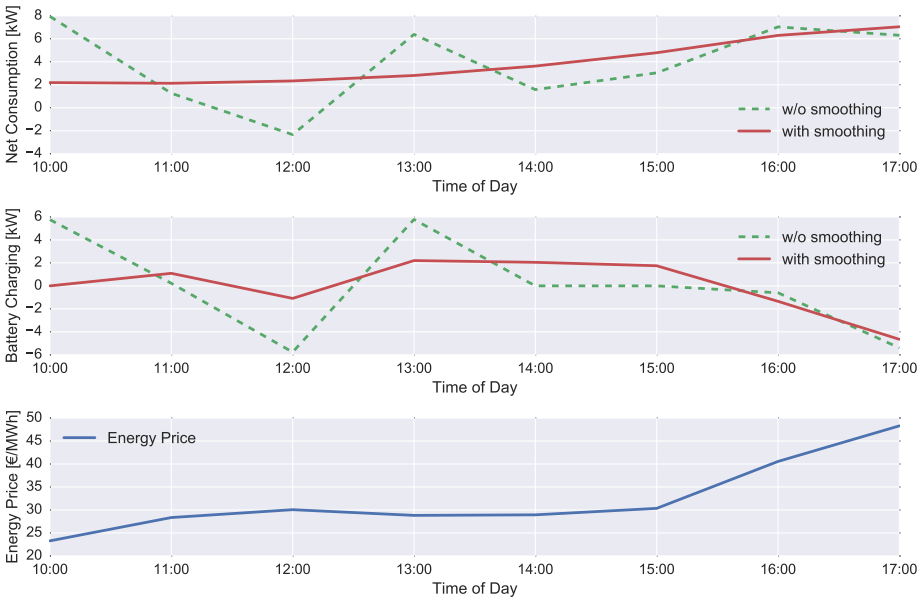

Figure 4.3: Net power profile with smoothing objective.

Parameter	Value
T	7
Δt	1 h
γ	100

Table 4.4: Smoothing experiment parameters.

Parameter	Value
T	7
Δt	1 h
γ	0.1
α_k	$1/k$
θ	1
K	100
c^s/c^b	$\{0, 0.8\}$

Table 4.5: Performance experiment parameters.

4.4 Distributed Optimization Algorithms

To compare the performance of the two algorithms described in Section 3.2, they were applied to problem (3.1) using the input parameters shown in Table 4.5. Both algorithms (without the acceleration step) were run for 100 iterations and two different c^s/c^b ratios. The performance of the algorithms in the different cases is shown in Figure 4.4. From the plots, it is visible that the performance improvement that can be achieved by using DProxG is heavily dependent on the properties of the separable objective. If the separable objective is nonlinear, e.g., when there is no revenue for feeding back energy into the grid, DProxG significantly outperforms DProjG. On the other hand, when the separable objective is close to linear, the benefits of DProxG are only marginal. This is consistent with the fact that both algorithms (without the acceleration step) are equivalent when the separable objective is linear, as shown in Lemma 1.

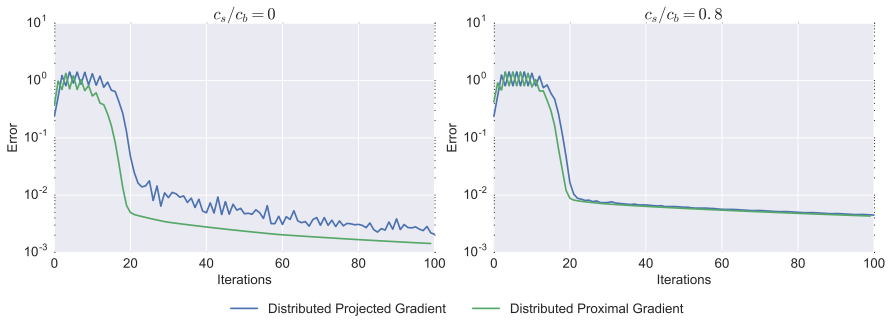

Figure 4.4: Algorithm performance with different energy cost functions.

4.5 Differential Privacy

To evaluate the effect of different privacy levels, several experiments were done. Since differential privacy adds randomness, the experiments were done multiple times, averaging the results over multiple samples. The parameters shown in Table 4.6 were used for the experiments. Six different values for ϵ were considered, ranging from $\log(3)$ (high privacy) to $\log(10^9)$ (very low privacy). Figure 4.5 shows the expected suboptimality achieved for different privacy levels as a function of the total iterations K for the cases of trustworthy and untrustworthy mediators. From the plots, several facts are evident: It can be seen that not trusting the mediator leads to poorer performance. This is consistent with the theoretical results from Section 3.3, which show that the variance of the gradient for the trustworthy mediator case is smaller compared to that for the untrustworthy case. Also, it is visible that there exists a "sweet-spot" for the number of total iterations K between 1 and 15, depending on the privacy level. When running the algorithm for more iterations, the variance of the noise needs to be higher to compensate for the increased information leakage due to more information exchanges. Hence, there exists a trade-off between the achieved privacy and the algorithm performance. For high privacy levels, only a few iterations can be done before the noise required to guarantee the desired privacy level makes the algorithm diverge. For very low privacy levels, the noise is smaller, resulting in better performance.

Parameter	Value
γ	100
θ	0.5
α_k	$1/k$
ϵ	$\{\log(3), \log(5), \log(10), \log(1e2), \log(1e4), \log(1e9)\}$
K	$\{0, \ldots, 30\}$
c^s/c^b	0.8
samples	10

Table 4.6: Parameters for the performance/privacy trade-off experiment.

4.6 Model Predictive Control

To evaluate the MPC, it was simulated during the time period from 5:00 to 23:00 with a prediction horizon of 8 hours with the parameters shown in Table 4.7. Four cases were considered: two cases with smoothing for a medium privacy level of $\epsilon = \log(10)$ in the trustworthy and untrustworthy mediator case, one case with smoothing and no privacy $\epsilon = \infty$, and one case without smoothing (for which privacy is not an issue). The simulation results obtained showing the battery SOC and charging power, the net consumption, and accumulated energy cost are shown in Figure 4.6. For convenience

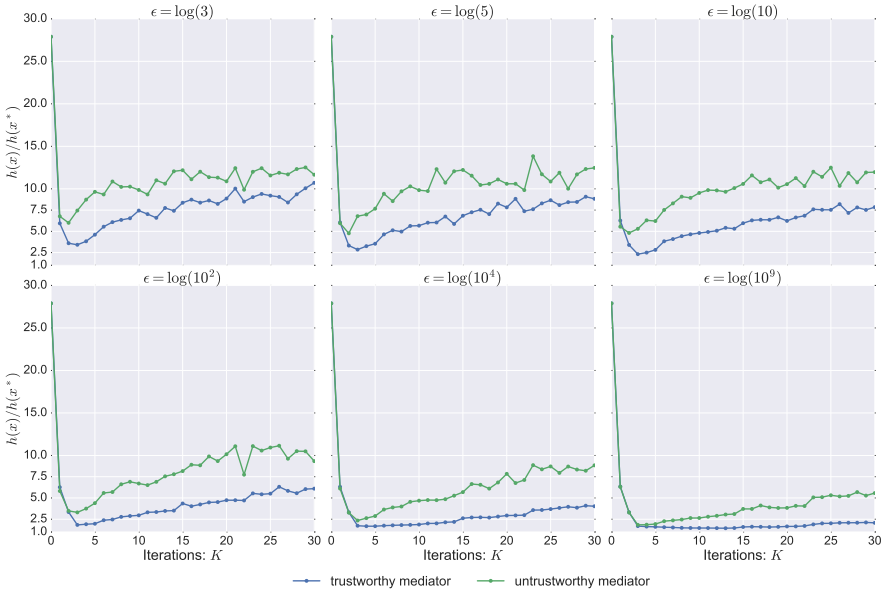

Figure 4.5: Performance/privacy trade-off.

to the reader, also the aggregate load consumption and the energy price during this period are shown. From the plot of the net consumption, it can be seen that the smoothing achieved by the MPC is clearly visible for all cases that consider smoothing. For the two cases with medium privacy, the smoothing performance is only slightly worse compared to the case without privacy. Looking at the plot of accumulated energy cost, it is visible that over time, the energy cost is higher when smoothing and privacy are considered. The periods where the cost for the cases with smoothing is lower can be explained by the prediction ability of the controller, which makes it store energy to exploit higher (predicted) energy prices in the future.

Parameter	Value
γ	100
θ	0.5
α_k	$1/k$
ϵ	$\{\log(10), \infty\}$
K (trustworthy)	4
K (untrustworthy)	2
c^s/c^b	0.8
T	8

Table 4.7: Parameters for the MPC simulations.

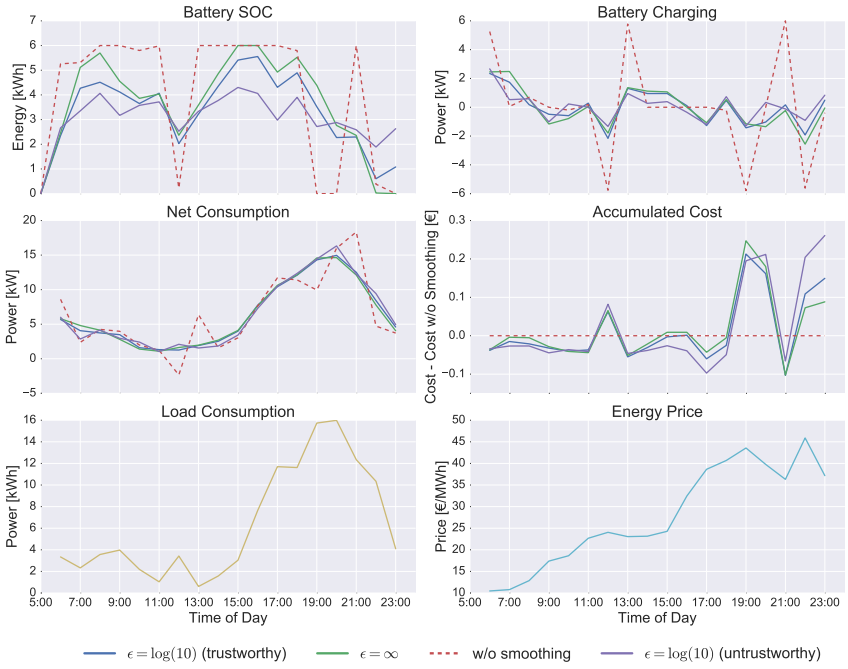

Figure 4.6: MPC performance.

Chapter 5

Conclusions

In this work, a differentially private distributed MPC for energy storage was proposed to achieve three goals: to minimize the energy cost of a group of consumers, to make their aggregate load profile smooth, and to protect their individual privacy. Making the aggregate load profile smooth is a task that requires communication between the different consumers and can therefore reveal privacy-sensitive information. Hence, one of the challenges addressed in this thesis was to achieve coordination of the different consumers while protecting their privacy.

Distributed optimization was proposed for solving the MPC subproblems. In particular, the Distributed Proximal Gradient algorithm was explored. This algorithm allows exploiting the composite structure of the objective and uses the load profile of each consumer locally for minimizing the energy cost. For achieving load smoothing, the algorithm needs to share net consumption information among consumers. Although the distributed approach does not require sharing the load profile directly, the information exchanges can reveal privacy sensitive data. To protect consumer privacy during these exchanges, the concept of Differential Privacy was introduced and incorporated into the MPC. Differential Privacy guarantees ϵ-indistinguishability of two load profiles that differ by a certain percentage. This means that the probability of seeing a certain sequence of information exchanges does not change much when the load profile of a consumer changes. Using the differentially private MPC, the trade-off between privacy and performance was characterized for the cases of trustworthy and untrustworthy mediators. It was found that if the mediator handling the information exchanges can be trusted, the variance of the noise required to achieve a certain privacy level is lower, resulting in better performance of the algorithm compared to the case of an untrustworthy mediator.

The proposed methodology provides protection against attacks that intercept the communication signals of the distributed control, but is unable to counter attacks that measure the physical consumption of the consumer. In oder to guarantee differential privacy in the face of such attacks, it would be necessary to add noise to the physical load profile using for example the battery. This poses some interesting research questions for future work, since the perturbations achievable with physical devices are

limited. Another possible extension to this work would be to also protect the specifications of the battery of each consumer.

Appendix A

Appendix

A.1 Gradient Derivation

In the following, the gradient $\nabla_{x_i} \bar{f}$ of the smoothing objective of the MPC subproblem (3.2) is derived. From (2.7), this is derived starting with

$$\begin{aligned} f(x(\tau), x(\tau-1)) &= \big(\sum_{i=1}^{N} p_i(\tau) - p_i(\tau-1)\big)^2 \\ &= \Big(\sum_{i=1}^{N} \left(u_i(\tau) - u_i(\tau-1) + q_i(\tau) - q_i(\tau-1)\right)\Big)^2, \end{aligned}$$

where

$$u_i(\tau) := l_i(\tau) - r_i(\tau)$$

denotes the uncontrolled part of the net consumption. Let u_i and q_i be the vectors containing the uncontrolled and battery power of consumer i for all $\tau \in \{t, \dots, t+T\}$:

$$\begin{aligned} u_i &= (u_i(t), \dots, u_i(t+T)) \\ q_i &= (q_i(t), \dots, q_i(t+T)) \,. \end{aligned}$$

Using this, the smoothing objective $\bar{f}$ can be expressed as

$$\bar{f}(x) = \gamma \big\| \sum_{i=1}^{N} (Du_i + Dq_i) \big\|_2^2 \,, \tag{A.1}$$

where the difference matrix $D^{T \times (T+1)}$ is defined as

$$D := \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} . \tag{A.2}$$

Using the projection matrix $P_q^{(T+1)\times 2N}$ that extracts the battery powers from x_i, (A.1) can be rewritten as

$$\bar{f}(x) = \Big\| \sum_{i=1}^{N} (Du_i + DP_q x_i) \Big\|_2^2 . \tag{A.3}$$

Defining $u := \sum_{i=1}^{N} u_i$ and $\tilde{D} := D^\top D$, (A.3) can be rewritten as

$$\begin{aligned}
\bar{f}(x) = \quad & \|Du + DP_q \sum_{k=1}^{N} x_k\|_2^2 \\
= \quad & \big(Du + DP_q \sum_{k=1}^{N} x_k\big)^\top \big(Du + DP_q \sum_{m=1}^{N} x_m\big) \\
= \quad & u^\top D^\top Du + 2u^\top D^\top DP_q \sum_{k=1}^{N} x_k + \sum_{k=1}^{N}\sum_{m=1}^{N} x_k^\top P_q^\top D^\top DP_q x_m \\
= \quad & u^\top \tilde{D}u + 2u^\top \tilde{D}P_q \sum_{k=1}^{N} x_k + \sum_{k=1}^{N}\sum_{m=1}^{N} x_k^\top P_q^\top \tilde{D}P_q x_m \\
= \quad & u^\top \tilde{D}u + 2u^\top \tilde{D} \sum_{k=1}^{N} P_q x_k + \sum_{k\neq i}\sum_{m\neq i} x_k^\top P_q^\top \tilde{D}P_q x_m \\
& + 2\sum_{k\neq i} x_k^\top P_q^\top \tilde{D}P_q x_i + x_i^\top P_q^\top \tilde{D}P_q x_i \,.
\end{aligned}$$

Using the above, the gradient of $\bar{f}$ with respect to x_i can be derived to be:

$$\begin{aligned}
\nabla_{x_i}\bar{f} &= 2P_q^\top \tilde{D}u + 2P_q^\top \tilde{D}P_q \sum_{k\neq i} x_k + 2P_q^\top \tilde{D}P_q x_i \\
&= 2P_q^\top \tilde{D}u + 2P_q^\top \tilde{D} \sum_{k=1}^{N} (P_q x_k) \\
&= 2P_q^\top \tilde{D}\big(\sum_{k=1}^{N} (u_k + P_q x_k)\big) \,.
\end{aligned}$$

The expression obtained for the gradient of $\bar{f}$ can now be used to derive the subdifferential of the total objective h at point x with respect to x_i:

$$\partial_{x_i} h(x) = \big\{\nabla_{x_i}\bar{f}\big\} + \partial_{x_i}\bar{g}_i(x_i) \,, \tag{A.4}$$

where $\partial_{x_i}\bar{g}_i(x_i)$ is the subdifferential of $\bar{g}_i$ at point x_i.

Bibliography

[1] Benchmarking smart metering deployment in the EU-27 with a focus on electricity. *Report for the European Comission*, 1, 2014. 1

[2] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, and Vijay Devabhaktuni. Smart meters for power grid: Challenges, issues, advantages and status. *Renewable and Sustainable Energy Reviews*, 15(6):2736–2742, aug 2011. 1

[3] Patrick McDaniel and Stephen McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7(3):75–77, 2009. 1

[4] Wilmer Heck. Smart energy meter will not be compulsory. *vorige.nrc.nl*, pages 1–2, 2009. 1

[5] Ulrich Greveler, B Justus, and D Loehr. Multimedia content identification through smart meter power usage profiles. *Computers, Privacy and Data Protection*, 2012. 1

[6] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. Private memoirs of a smart meter. *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, pages 61–66, 2010. 1

[7] Markus Weiss, Adrian Helfenstein, Friedemann Mattern, and Thorsten Staake. Leveraging smart meter data to recognize home appliances. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 190–197. IEEE, mar 2012. 1

[8] David Varodayan and Ashish Khisti. Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1932–1935, 2011. 2

[9] S. Raj Rajagopalan, Lalitha Sankar, Soheil Mohajer, and H. Vincent Poor. Smart meter privacy: A utility-privacy framework. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 190–195. IEEE, oct 2011. 2

[10] Jesús Gómez-Vilardebó and Deniz Gündüz. Smart meter privacy for multiple users in the presence of an alternative energy source. *IEEE Transactions on Information Forensics and Security*, 2015. 2

[11] Giulio Giaconi, Deniz Gunduz, and H. Vincent Poor. Smart meter privacy with an energy harvesting device and instantaneous power constraints. In *IEEE International Conference on Communications*, 2015. 2

[12] Cynthia Dwork, F. McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, pages 265–284, 2006. 2

[13] Henrik Sandberg, György Dán, and Ragnar Thobaben. Differentially Private State Estimation in Distribution Networks with Smart Meters. (Cdc):4492–4498, 2015. 2

[14] Jing Zhao, Taeho Jung, Yu Wang, and Xiangyang Li. Achieving differential privacy of data disclosure in the smart grid. *Proceedings - IEEE INFOCOM*, (Ilm):504–512, 2014. 2

[15] Shuo Han, Ufuk Topcu, and George J. Pappas. Differentially private distributed protocol for electric vehicle charging. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2014*, pages 242–249. Institute of Electrical and Electronics Engineers Inc., 2015. 2, 8

[16] John Duchi and Y Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009. 2, 8, 9

[17] D. Bertsekas. *Nonlinear Programming*. 1999. 8

[18] Patrick L. Combettes and Valérie R. Wajs. Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005. 9

[19] Neal Parikh and Stephen Boyd. Proximal Algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014. 9

[20] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(2013):211–407, 2014. 11, 12, 13

[21] Tomás Tinoco De Rubira and Martin Zellner. PFNET: A library for modeling and analyzing electric power networks. `http://ttinoco.github.io/PFNET/`, July 2015. 20

[22] Martin Zellner. MP-PFNET: A multi-period extension to pfnet. `https://github.com/martinzellner/mp-pfnet`, 2016. 20

[23] Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016. 20

[24] A Domahidi, E Chu, and S Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013. 20

[25] Eric Jones, Travis Oliphant, Pearu Peterson, and Others. SciPy: Open source scientific tools for Python. 20

[26] The IPython Development Team. ipyparallel, 2016. 20

[27] Gerwin Hoogsteen, Albert Molderink, Johann L Hurink, and Gerard J M Smit. Generation of Flexible Domestic Load Profiles to Evaluate Demand Side Management Approaches. *2016 IEEE International Energy Conference (ENERGYCON)*, 2016. 20

[28] NREL. PVWatts Calculator, 2016. 21