KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Electrical Engineering

Intelligent System Designs: Data-driven Sensor Calibration & Smart Meter Privacy

YANG YOU

Stockholm, Sweden 2022

Intelligent System Designs: Data-driven Sensor Calibration & Smart Meter Privacy

YANG YOU

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Friday the 26th August 2022, at 9:00 a.m. in Sal Kollegiesalen, Brinellvägen 8, Stockholm.

Doctoral Thesis in Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden 2022

ISBN 978-91-8040-277-4
TRITA-EECS-AVL-2022:42

Printed by: Universitetsservice US-AB, Sweden 2022

To my family

Abstract

Nowadays, the intelligent system has gained high popularity in successful implementation of real-time tasks due to its capability of providing efficient and powerful decision making in real applications. In this thesis, we aim for exploring and exploiting different concepts or methods to handle different tasks towards the intelligent system design. In particular, we focus on the following two problems: (i) Consumer-centric privacy-cost trade-off in smart metering system; (ii) Data-driven calibration for gas sensing system.

For the first target problem, an optimal privacy-preserving and cost-efficient energy management strategy is designed for each smart grid consumer that is equipped with a rechargeable energy storage. The Kullback-Leibler divergence rate is used as privacy measure and the expected cost-saving rate is used as utility measure. The corresponding energy management strategy is designed by optimizing a weighted sum of both privacy and cost measures over a finite time horizon, which is achieved by formulating our problem into a partial observed Markov decision process problem. A computationally efficient approximated Q-learning method is proposed as a extension to high-dimensional problems over an infinite time horizon.

Furthermore, the privacy-preserving and cost-efficient energy management strategy is designed for multiple smart grid consumers that are equipped with renewable energy sources. Different from the previous problem, the adversary is assumed to employ a factorial hidden Markov model based inference for load disaggregation, and the corresponding joint log-likelihood of the model is utilized as privacy measure. A dynamic pricing model is studied, where the price of unit amount of energy is determined by the consumers' aggregated power request, which suits a commodity-limited market. The consumers' energy management strategy is designed under a non-cooperative game framework by optimizing a weighted sum of both privacy measure and the user's energy cost savings. The consumers' non-cooperative game is shown to admit a unique pure strategy Nash equilibrium. As an extension, a computational-efficient distributed Nash equilibrium energy management strategy seeking method is proposed, which also avoids the privacy leakage due to the sharing of payoff functions between consumers.

For the second target problem, several data-driven self-calibration algorithms are developed for low-cost non-dispersive infrared sensors. The measurement errors of the sensors are mainly caused by the remaining model errors and can be fully described by the drift of the calibration parameter. This leads to our first formulation of a statistical inference problem on the true calibration parameter under the HMM framework, which is a stochastic model that jointly builds on different quantities introduced by the physical model. To better track the time-varying drift process of the sensor, a time-adaptive expectation maximization learning framework is proposed to efficiently update the HMM parameters. For the joint calibration of the gas sensing system, sensors first transmit their belief functions of the true gas concentration level to the cloud. Then the cloud fusion center computes a fused belief function according to certain rules. This belief function is then used as reference for calibrating the sensors. To deal with the case where belief functions highly conflict with each other, a Wasserstein distance based weighted average belief function fusion approach is first proposed as networked calibration algorithm. To achieve more long-term stable cali-

bration results, the networked calibration problem is further formulated as a partially observed Markov decision process problem, and the calibration strategies are derived in a sequential manner. Correspondingly, the deep Q-network approach is applied as a computationally efficient method to solve the proposed Markov decision process problem.

The results in this thesis have shown that our proposed design frameworks can provide concise but precise mathematical models, proper problem formulations, and efficient solutions for the target design objectives of different intelligent systems.

Sammanfattning

Intelligenta system har vunnit popularitet inom framgångsrik implementering av realtids uppgifter på grund av dess förmåga att tillgodose effektiv och kraftfull beslutsförmåga i verkliga applikationer. I denna avhandling utforskar vi, och nyttjar olika koncept eller metoder för att hantera olika uppgifter inom design av intelligenta system. Specifikt fokuserar vi på följande två problem: (i) avvägning mellan integritet och kostnad hos konsumenten i smarta mätsystem; (ii) data driven kalibrering för gassensorsystem.

Det första delproblemet hanterar designen av en optimal strategi för energi hantering, som är integritetsbevarande samt kostnadseffektiv, för varje smart-grid konsument som har uppladdningsbar energilagring. Kullback-Leibler divergensen används som mått på integritet, och den väntade kostnadsbesparingen används som nyttomått. Staregin för energihantering kommer av optimering av en viktad summa av både integritetsmått och kostnads faktorer över en ändlig tidshorisont, som åstadkoms genom att formulera problemet som en delvis observerad Markov-beslutsprocess. En beräkningsmässigt effektiv metod med approximativ Q-learning föreslås som en generalisering till högdimensionella problem över en obegränsad tidshorisont.

Vidare är den integritetsbevarande och kostnadseffektiva energihanteringsstrategin strategin designad för flera smart-grid konsumenter som har förnybara energikällor. Till skillnad från det tidigare problemet antas här motståndaren använda en slutledningsmetod för uppdelningen av energiförbrukningen, som baseras på en dold Markov modell, och den tillhörande log-sannolikheten av modellen används som mått på integritet. En dynamisk prissättningsmodell, som passar en resursbegränsad marknad, undersöks där priset för varje enhet av energi fastställs av konsumenternas aggregerade efterfrågan av effekt, som passar en resursbegränsad marknad. Konsumentens energihanteringsstrategi kommer från ett spel där en viktad summa av både integritets mått och konsumentens besparingar i energikostnader optimeras. Konsumentens spel visas ge ett unikt pure startegi Nash jämviktstillstånd. Som ett tillägg föreslås en metod som söker Nash jämvikt i energihantering på ett beräkningsmässigt effektiv sätt, som också undviker att läcka privat data som följd av att användare delar payoff funktioner.

I det andra delproblemet, utvecklas flera datadrivna och självkalibrerande algoritmer för infraröda sensorer av lågkostnadstyp. Mätfelen i sensorerna beror främst på återstående modellfel och kan beskrivas fullt ut av kalibreringsparameterns drift. Detta leder till vår första problemformulering av ett statistiskt slutledningsproblem av de sanna kalibreringsparametrarna under HMM ramverket. Detta är en stokastisk modell som simultant bygger på olika kvantiteter introducerade av den fysiska modellen. För att bättre följa den tidsvarierande förändringsprocessen av en sensor, föreslås ett tidsadaptivt expectation maxmization ramverk för att effektivt uppdatera HMM parametrarna. För den simultana kalibreringen av gassensorsystemet, sänder sensorerna deras hypotesfunktioner för den sanna gaskoncentrationsnivån till ett datormoln. Där beräknas en gemensam hypotes efter vissa regler. Denna hypotesfunktion används sedan som referens för att kalibrera sensorerna. För att hantera fallet med hög konflikt mellan hypotesfunktioner, används som nätverkskalibrerings algoritm först ett Wasserstein-avstånds baserad viktad medelvärde av hypotesfunktioner. För att uppnå mer långsiktigt stabila kalibreringsresultat, formuleras nätverkskalibreringsproblemet vidare som

ett delvis observerbart Markov-beslutsprocess problem, och kalibreringsstrategierna bestäms sekventiellt. På motsvarande sätt använd deep Q-network metoden som en beräkningsmässigt effektiv metod för att lösa det föreslagna Markov-beslutsprocess problemet.

Resultaten I denna avhandling har visat att vårt föreslagna ramverk kan tillhandahålla koncisa och noggranna matematiska modeller, propra problemformuleringar och effektiva lösningar för ändamål inom olika intelligenta system.

Acknowledgements

This thesis could not be finished without the help and support from many professors, colleagues, friends, and my family. It is my pleasure to acknowledge people who give me help, guidance, and encouragement.

First and foremost, I would like to thank my supervisor Prof. Tobias Oechtering for his consistent guidance, motivation, and support during my Ph.D. study. I am deeply grateful to all of your efforts and patience for helping me improve myself gradually. I would also like to thank Assist. Prof. Zuxing Li who brings me all the inspiration and the encouragement during the early stage of my Ph.D study. It was a precious experience to collaborate with you along all the way.

I would like to express my sincere gratitude to Assist. Prof. Olga Fink from École Polytechnique Fédérale de Lausanne for acting as the opponent, and to the grading board members: Assoc. Prof. Edith Ngai from The University of Hong Kong, Assoc. Prof. André Teixeira from Uppsala University, Assist. Prof. Mustafa A. Mustafa from The University of Manchester, and Prof. Cristian Rojas. I would like to thank Prof. Mats Bengtsson for being the defense chair as well as the director for my master program, thanks for all of your help during both my master and Ph.D. study. I would also like to thank Prof. Magnus Jansson for the constructive feedback from the advanced review. Many thanks to Alexander Karlsson and Movitz Lenninger for the Swedish translation of the abstract.

Furthermore, I want to thank my teachers: Prof. Mikael Skoglund, Prof. Joakim Jaldén, Prof. James Gross, Assoc. Prof. Ming Xiao, Assoc. Prof. Ragnar Thobaben, Assoc. Prof. Markus Flierl, Assoc. Prof. Saikat Chatterjee, and all other professors and senior researchers at ISE. I devote special thanks to Anna Mård for careful and efficient HR support. I must thank all my past and current colleagues for creating the enjoyable working environment. It is a pleasure to share the office with my kind and humor officemate Dr. Hasan Basri Celebi. I really enjoyed the pleasant and relaxed time that I have spent with Yue Xiao, Zequn Fang, Yu Ye, and Shaocheng Huang during their stay at ISE. I am also grateful to Dr. Minh Thanh Vu and C V Ramana Reddy Avula for all the interesting technical discussions and the careful proofreading for my thesis. It is my pleasure to have (or had) the nice colleagues: Assist. Prof. Sadegh Talebi, Dr. Phuong Le Cao, Dr. Yang Guang, Dr. Alireza Mahdave Javid, Dr. Sina Molavipour, Hao Chen, Yusen Wang, Hamid Ghourchian, Borja Rodriguez Galvez, Wanlu Lei, Xuechun Xu, Lissy Pellaco, Baptiste Cavarec, Michail Mylonakis, Prakash Borpatra Gohain, Leandro Lopez, Xinying Ma, Deyou Zhang, Jin Huang, and Honghao Lv. I am grateful to my friends Xiaoyu Zhao, Shuai Guo, Yang Zhou, Zeyu Lin, Yazhou Shen, Yicheng Bao, Yuxiao Cui, and Tong Han for all those wonderful memories during my past seven years in Sweden. I am also grateful to all my fellows for the energetic hours after work of playing basketball, table tennis, or Dota 2. Thank you for your accompany.

Last but not least, my deepest gratitude goes to my family members, especially my father Hongbo You and my mother Yongtao Luo for their endless love ever since my birth. I am so grateful to you and feel deeply indebted to your belief and patience. It would be impossible to come through all the difficulties in my life without your support. This thesis is dedicated to you with love!

Yang You
Stockholm, May 2022

Contents

List of Papers

The thesis is based on the following papers:

- **Paper A.** Y. You, Z. Li and T. J. Oechtering, "Energy Management Strategy for Smart Meter Privacy and Cost Saving," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1522-1537, 2021.
- **Paper B.** Y. You, Z Li, and T. J. Oechtering, "Non-cooperative Games for Privacy-preserving and Cost-efficient Smart Grid Energy Management," submitted to *IEEE Transactions on Information Forensics and Security*, January 2022.
- **Paper C.** Y. You, K. You, H. Chen, and T. J. Oechtering, "On Data-Driven Self-Calibration for IoT-Based Gas Concentration Monitoring System," in *IEEE Internet of Things journal*, pp. 1-1, 2022.
- **Paper D.** Y. You, and T. J. Oechtering, "Time-adaptive Expectation Maximization Learning Framework for HMM based Data-driven Gas Sensor Calibration," submitted to *IEEE Transactions on Industrial Informatics*, April 2022.

Other works during my Ph.D. study that are not included in this thesis:

[I] Y. You and T. J. Oechtering, "Hidden Markov Model Based Data-driven Calibration of Non-dispersive Infrared Gas Sensor," in *28TH European Signal Processing Conference (EUSIPCO 2020)*, 2020, pp. 1717-1721.

[II] Y. You, A. Xu and T. J. Oechtering, "Belief Function Fusion based Self-calibration for Non-dispersive Infrared Gas Sensor," in *2020 IEEE SENSORS*, 2020, pp. 1-4.

[III] Y. You and T. J. Oechtering, "Online Energy Management Strategy Design for Smart Meter Privacy Against FHMM-based NILM," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020.

[IV] Y. You, Z. Li and T. J. Oechtering, "Optimal Privacy-Enhancing and Cost-Efficient Energy Management Strategies for Smart Grid Consumers," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, 2018, pp. 144-148.

[VI] Z. Li, Y. You and T. J. Oechtering, "Privacy against adversarial hypothesis testing: Theory and application to smart meter privacy problem," in Privacy in Dynamical Systems, : *Springer Singapore*, 2019, pp. 43-64.

List of Acronyms

CO_2	carbon dioxide
DQN	deep Q-network
DS rule	Dempster-Shafer rule
EM	expectation maximization
EMU	energy management unit
EP	energy provider
ES	energy storage
FHMM	factorial hidden Markov model
GDPR	general data protection regulation
HMM	hidden Markov model
i.i.d.	independent and identically distributed
KL-divergence	Kullback-Leibler divergence
LARQL	linear function approximated relative Q-learning
MAP	maximum a posteriori
MDP	Markov decision process
NDIR	non-dispersive infrared
NE	Nash equilibrium
NILM	non-intrusive load monitroing
pmf	probability mass function
POMDP	partially observed Markov decision process
RES	renewable energy source
RVI	relative value iteration

Chapter 1

Introduction

Nowadays, the concept of intelligent system has emerged in information technology as a type of system derived from successful applications of artificial intelligence. In this thesis, we aim for designing intelligent algorithms that can efficiently handle the tasks such as data-driven modelling, statistical analysis, and decision making in different intelligent systems. In particular, we focus on the following two problems: (i) Consumer-centric privacy-cost trade-off in smart metering system; (ii) Data-driven calibration for gas concentration level monitoring system.

This chapter is structured as follows. The background and the state-of-the-art of this thesis are first provided in Section 1.1. Section 1.2 briefly introduces the recent work. Following that, Section 1.3 elaborates the problems studied in this thesis. The outline of this thesis is provided in Section 1.4. Finally, Section 1.5 gives copyright notification.

1.1 Background and State-of-the-Art

In this section, we present the background and the state-of-the-art of the two specific problems that have been studied in this thesis.

1.1.1 Privacy-cost Trade-off in Smart Metering System

In future smart grids, smart meters are essential components to deliver real-time information about users' energy demands to the energy provider (EP). The information can help the EP to improve the prediction of the future energy demands and therefore to increase the efficiency of the whole smart grid [1]. However, this benefit is at a cost of privacy of the users. For instance, an adversary, who could be a legitimate receiver of the data, e.g., the energy grid operator, can use standard energy load disaggregation algorithms [2–6] to learn the power consumption behaviors of users.

Regarding the smart meter privacy problem, different privacy-enhancing approaches have been proposed. One approach is to modify the smart metering data before it is sent

to the EP by using off-the-shelf methods, such as obfuscation [7], anonymization [8], and data aggregation [9]. However, these methods fail when the adversary has access to the true measurements, e.g., by compromising the EP, unauthorized installation of a sensor, or sharing of measurements with a third party for a different purpose. Another privacy-by-design approach is to use an energy storage (ES) such as a rechargeable battery [10–14], or an alternative energy supply such as renewable energy source (RES) [15–17] to modify the energy demand profile by using a privacy-enhancing energy management strategy. The latter approach will meet the requirements of the EU General Data Protection Regulation (GDPR) [18], which calls for an authorized data recipient to hold and process only the data absolutely necessary for the completion of its duties.

On the other hand, the privacy enhancement may lead to an increased energy cost, which would often violate the original cost-saving motivation of the energy storage investment for users. In such cases, it becomes important to design energy management strategies that aim for both privacy and cost efficiency. Fortunately, the implementation of privacy-by-design approaches that utilize the energy storage or an alternative energy supply to modify the users' actual energy demand profile can be also used to reduce the energy costs of the users [14].

1.1.2 Low-cost Gas Sensor Calibration

In recent years, there has been a growing interest in deploying gas sensor networks for gas concentration monitoring [19–22]. For example, gas concentration monitoring systems that are embedded with carbon dioxide (CO_2) sensors are essential for actions against global warming for monitoring the greenhouse effect. Monitoring of gas concentrations requires gas sensors with long-term stability and sufficient accuracy in their measurements. To keep the costs low, one often includes components that have unknown dependencies on external factors that can lead to a deduction in the accuracy of the sensor measurement, which is also known as drift. There are several typical working principles for the gas sensors, such as metal-oxide semiconductors, electrochemical cells or optical gas sensing. In this thesis, we focus on the optical non-dispersive infrared (NDIR) gas sensor, which is one of the most common optical gas sensors. Compared to other types of gas sensors, the NDIR sensors provides high specificity, low life-cycle cost, minimal drift, stable long-term operation [23]. However, previous works such as [24–26] have recognized that NDIR sensors are sensitive to the variations of ambient temperature, atmospheric pressure, humidity and some other environmental factors, which cannot always be compensated for. Due to this sensitivity, regular calibration is needed for long-term accuracy of the sensors.

Today, the state of the art of infrared gas sensor self-calibration is the well-established automatic baseline correction technology where the sensor is calibrated to a fixed value that is assumed to be the fresh air gas concentration [27]. However, this method does not work well in mega-cities where the sensors never get exposed to fresh air. Furthermore, the method cannot be used in the rapidly growing market of environmental sensors where the baseline, or fresh air gas concentration, is the measurement of primary interest. Thus, designing smart, more robust, and networked calibration algorithms which can be widely

applied in different environments becomes more and more important.

Next, we provide a sensing model and the corresponding drift analysis, where the measurement error happens due to the imperfect compensation for the variation of the behavior of the sensor components. Based on this model, we provide various calibration schemes to deal with the corresponding drift in this thesis. Also note that the concepts provided in this thesis are not restricted to NDIR gas sensors only, our results can also be applied to other sensors that fit this more general sensing and drift model. Generally speaking, the basic working mechanism of a sensor is to convert a specific physical quantity to an electrical signal. Denoting the intensity of the converted electrical signal as E, consider the following general sensor measurement model

$$O = f(r, E, F), \tag{1.1}$$

where O is the sensor output, F describes certain environmental factors, and r denotes a calibration parameter. The function $f(\cdot)$ describes the physical model of the sensor and maps the pair (r, E, F) to a certain sensor output O.

The behavior of the sensor components varies according to the environmental factors, which will cause a drift of the converted electrical signal E. Since the mapping f usually cannot fully capture the dependency between the behavior of the sensor components and the environmental factors, one thus need to adjust the calibration parameter according to the drift of the electrical signal E so that the remaining imperfections of the mapping f are compensated. We call the calibration parameter that perfectly compensates the imperfect mapping f as *true calibration parameter*, which is denoted as x in the following. The true calibration parameter can be obtained by operating the sensor under reference conditions, i.e., we can solve (1.1) for x given the true values for O, E and F. However, true values are usually unavailable while the sensor is operating. Our basic calibration idea therefore is to develop a calibration procedure to learn the true calibration parameter to enhance the performance of the sensors.

1.2 Literature Survey

This section presents an overview of existing research related to the scope of this thesis and clarifies the research gap, established on which we formulate the research problems in Section 1.3.

1.2.1 Privacy-cost Trade-off in Smart Metering System

Privacy measures. To design the aforementioned privacy-by-design approaches for the smart grid consumers, different privacy measures have been considered in previous works. In [28] and [29], differential privacy has been proposed as a privacy measure for the smart grid consumers. In [13], the variance of random energy supplies from the EP have been used as privacy measure. In [10, 11, 15, 16, 30], the privacy leakage is measured by different information theoretic metrics such as mutual information or conditional entropy rates.

Another important approach is to consider a privacy-preserving objective derived from a specific adversarial hypothesis test scenario, e.g., [31, 32]. Compared to the aforementioned approaches, the hypothesis testing privacy measure has a clear operational meaning, but it is also limited by the specific assumptions of the considered scenario. Recently, the Kullback-Leibler (KL) divergence is used to measure privacy leakage in [33] and [34]. The KL divergence characterizes the asymptotic Neyman-Pearson hypothesis testing performance of the adversary with independent and identically distributed (i.i.d.) observations. Along this line, we adopt the KL-divergence as the privacy measure in our **Paper A**. Correspondingly, the objective is to minimizing the KL-divergence between the distributions of energy request considering a binary hypothesis test on the consumers' behavior.

Meanwhile, the non-intrusive load monitoring (NILM) algorithms [35] have been widely studied and applied to load disaggregation. Instead of intrusively monitoring the energy consumption of the individual appliances, the NILM algorithm can achieve the load disaggregation of voltage or current waveforms measured at the electrical services entry point. The HMM and its variants have been widely used for the NILM algorithms [36–38]. In particular, as an extension of the basic HMM, the authors in [3] propose an additive factorial hidden Markov model (FHMM) based load disaggregation, where the aggregated energy consumption of the user is modeled as an additive form of each appliance's energy consumption. In more detail, the authors propose a maximum a posterior (MAP) inference method under the FHMM framework for load disaggregation, where the MAP estimation is performed to find the most likely operating state sequences of different appliances based on the complete-data (both hidden states and observations) likelihood. However, due to its high efficiency on the load disaggregation, the NILM algorithm has brought more privacy risks in the meanwhile. To protect the users' privacy against the NILM algorithms, different privacy-preserving schemes have been proposed recently [39–42]. Accordingly, in our **Paper B**, we particularly study the privacy-preserving problem against FHMM based load disaggregation algorithm that utilizes MAP inference, where the corresponding complete-data likelihood of the FHMM is adopted as the privacy measure. The design of the privacy-preserving energy management strategy is thus achieved by minimizing this complete-data log-likelihood.

Design approaches. With the aforementioned privacy measures, different approaches have been proposed for the design of privacy-preserving mechanisms. This includes heuristic approaches, such as the best-effort water-filling algorithm in [43] that aims to keep the output load at its most recent value. A battery-based noise adding mechanism is designed in [28] and [29] to achieve differential privacy. In [44] and [45], control optimization methods such as model-distribution predictive control and cloud-based control have been applied to enhance the privacy. Likewise, a stochastic control approach is considered in [10, 11, 15, 16], where the privacy-preserving energy management design problem is transferred into an optimal control strategy design problem using the Markov decision process (MDP) framework. In our **Paper A**, we first formulate the energy management design problem as an equivalent average reward partially observed Markov decision process (POMDP) problem so that the optimal solution is given by Bellman dynamic programming. However, characterizing the optimal strategy under the POMDP framework is

computationally challenging due to the continuous state-action space. Thus, more computational efficient methods are further proposed in **Paper A**.

As a formal analytical and conceptual framework with a set of mathematical tools which enables the study of complex interactions among independent rational players, game theory provides a robust theoretical framework to address different challenges in the smart grid [46]. Recently in [47], the authors propose a game-theoretical framework to model the interaction the receiver and the sensors with extra privacy concerns, which can be perfectly fitted into the privacy-preserving problems in smart grid. In [48], the authors propose a stochastic game framework to characterize the messaging behaviors of users with privacy requirements for the energy storage sharing problem. In [49], the authors propose a Stackelberg Game framework to model the competition between users' demand and energy price, under which a homomorphic encryption scheme is further deployed to preserve the users' privacy. In our **Paper B**, a non-cooperative game framework is proposed to model the competing behaviors between the users who wish to optimize their own privacy-cost trade-off objective with respect to the coupled power request. Correspondingly, the designed energy management strategy is provided by Nash equilibrium (NE) as the most suitable solution to this non-cooperative game.

Privacy-cost tradeoffs. While most of the aforementioned papers focus on how to preserve the privacy, only [11], [13], [14], and [16] have taken the consumers' cost for purchasing the energy into consideration. For example, in [13] the online dynamic programming problem is relaxed to a Lyapunov optimization problem which jointly optimizes the privacy and the energy cost. In [14], first an offline convex optimization problem for the privacy-cost trade-off is solved, then a low-complexity heuristic online control algorithm is proposed as an alternative solution to the original online dynamic programming problem. Along a different line, the authors in [11] and [16] formulate the privacy-cost trade-off problem into an offline stochastic control problem under the MDP framework. In the thesis, for both **Paper A** and **Paper B**, we adopt the quantity cost-savings as the cost measure. In order to design the energy management strategies that trade off between privacy and cost, a weighted sum of the privacy measure and the cost measure is considered as an overall objective function.

1.2.2 Low-cost Gas Sensor Calibration

Data-driven calibration. Data-driven modeling methods aim to find relationships between system state variables without explicit knowledge of the physical behavior of the system [50]. Research works such as [19, 51, 52] have applied data-driven modeling methods for calibration of different sensors or even sensor networks. Meanwhile, machine learning approaches provide increasing levels of automation and improved accuracy by discovering and exploiting dependencies in the (training) data. Recent research works have combined machine learning approaches to build data-driven models [53–55]. For example, the authors in [55] propose to use the multi-layer perceptron neural network and the extreme learning machine as two alternative data-driven modeling approaches for the wind speed time series prediction. In **Paper C** and **Paper D**, we also exploit the concept of data-driven

modeling and the machine learning approaches to enhance a gas sensor calibration mechanism. In more details, we utilize the HMM as a statistical learning tool to build data-driven self-calibration algorithms for NDIR sensors.

Multi-sensor data fusion. To reduce inaccuracy and uncertainty of single sensor measurements, the gas sensor network is often deployed [20]. In such scenario, improved calibration can be achieved by multi-sensor data fusion, where information from different sensors is combined in the cloud to provide an overall more accurate gas concentration measurement. To deal with the uncertainty caused by the imperfection of the data, different Bayesian based approaches have been proposed as data fusion techniques. As a special case of the Bayesian filters, the Kalman filter and its variants have been widely used for sensor fusion in many existing works, such as [56–58].

As a generalization to the traditional Bayesian probability theory, the belief function theory (also known as Dempster-Shafer (DS) theory) [59] provides a particularly convenient theoretical framework for uncertainty modeling and propagation in the combination of partially reliable information. It has been widely applied to multi-sensor data fusion problems, such as [60–62]. However, the traditional DS theory will result in unreasonable fused beliefs when the belief functions provided by different sensors highly conflict with each other. To mitigate this issue, pre-processing methods of the original belief functions have been proposed [63]. In [64], the authors propose to incorporate an average belief into the DS combining rule. However, this simple average approach assigns equal weight to each body of evidence and does not consider the relationship among the evidence collected from multiple sensors, which is often unreasonable in real applications. For those cases, different weighted average approaches [65–67] have been proposed to manage the conflicts between belief functions and are shown to have good performance for the specific applications. In **Paper C**, based on the stochastic model we built for the single sensor using HMM, we further propose a general belief function fusion framework for the networked calibration of NDIR gas sensors. Then we propose a modified weighted average approach to deal with the case where belief functions highly conflict with each other. Different from the aforementioned works [65–67], which utilize the Jousselme distance [68] or the modified Jousselme distance as a measure of the distance between the belief functions, we propose to use the Wasserstein distance [69] as a measure of distance between the belief functions. This is done since the Jousselme distance is not applicable for the case where the belief functions are just simple Bayesian probability measures.

Reinforcement learning for sensor fusion. The aforementioned multi-sensor fusion approach only provides an instantaneous solution for the NDIR gas calibration problem, without considering the overall long-term performance. This approach is however easy to be implemented since it does not require a statistical relation between sensors, i.e., joint statistics. It will work well in scenarios where the majority of the belief functions provides a reliable and correct evidence. Moreover, the fusion result provided by this approach can be used as an approximation of the joint statistics of the system. With this approximated joint statistics, we next formulate the calibration problem as an MDP problem [70] in **Paper C**, where the long-term stable calibration results are derived in a sequential manner.

Reinforcement learning provides model-free methods to solve such optimal control problems of dynamic systems without requiring the knowledge of the stochastic properties of the underlying model. Due to this convenience, reinforcement learning methods have been widely applied in sensor calibration problems during the recent years. In [71], the authors proposed a Q-learning based Gaussian mixture model algorithm for vehicle tracking via multi-sensor fusion. Meanwhile, a SARSA [72] based reinforcement learning sensor fusion algorithm is proposed for autonomous robot navigation in [73]. When combined with deep learning, the more powerful deep reinforcement learning methods can deal with the curse-of-dimensionality problem by approximating the value functions as well as policy functions using deep neural networks. In this thesis, we re-formulate our calibration problem as a stochastic optimal control problem and propose to use a deep Q-network (DQN) to solve it.

1.3 Thesis Scope

The research goal of this thesis is **to explore the application of existing concepts or methods on handling different tasks towards intelligent system design.** Under this goal, we summarize the general research questions of this thesis into the following.

- *Research question 1 (RQ1): How to build an intelligent system design framework such that: (i) It admits a concise mathematical model that properly captures the relationships between relevant quantities in the considered intelligent system; (ii) It includes mathematical problem that describes the target design objectives and is formulated based on the proposed mathematical model; (iii) The design problem can be solved by utilizing, adapting and/or extending existing concepts or methods from the literature.*
- *Research question 2 (RQ2): What performance can be achieved in proof-of-concept experiments and how to improve the proposed design framework to achieve better performance and efficiency?*

The above research questions are explored in two intelligent system problems, namely the smart meter privacy problem and the data-driven sensor calibration problem. More specifically, we explore the following research problems.

Research problem 1 (RP1): How to formulate a mathematical problem that can properly describe the privacy-cost trade-off for the smart grid consumers?

Research problem 2 (RP2): How to design computational efficient privacy-preserving and cost-efficient energy management strategies with guaranteed performance?

Research problem 3 (RP3): How to build a mathematical model to accurately describe the drift process of the gas sensors?

Research problem 4 (RP4): How to design intelligent self-calibration algorithms for gas sensors?

1.4 Thesis Outline

The investigation of the above research problems is divided into four main chapters, one technical introduction and a brief conclusion. We summarize the content of each chapter in the following.

In Chapter 2, we first provide the review of the basic definitions and principles for HMM, MDP, and POMDP. Then, we briefly introduce the working mechanism and the corresponding drift analysis of NDIR gas sensors.

In Chapter 3, we consider the privacy-enhancing and cost-efficient energy management design problem for a single consumer that is equipped with an ES. The KL-divergence rate is used as privacy measure and the expected cost-saving rate is used as utility measure. The corresponding energy management strategy is designed by optimizing a weighted sum of both privacy and cost measures over a finite time horizon, which is achieved by formulating our problem into a POMDP problem. A computationally efficient approximated Q-learning method is proposed as a generalization to high-dimensional problems over an infinite time horizon. At last, we explicitly characterize a stationary policy that achieves the steady belief state over an infinite time horizon, which greatly simplifies the design of the privacy-preserving energy management strategy. The chapter content is based on **Paper A**.

In Chapter 4, we consider the privacy-enhancing and cost-efficient energy management design problem for multiple consumers that are equipped with RESes. The adversary is assumed to employ an FHMM based inference for load disaggregation, and the corresponding joint log-likelihood of the model is utilized as privacy measure. A dynamic pricing model is studied, where the price of unit amount of energy is determined by the users' aggregated power request, which introduces conflict of interest among different consumers. Correspondingly, the users' energy management strategy is designed under a non-cooperative game framework by optimizing a weighted sum of both privacy measure and the user's energy cost savings. This non-cooperative game is further shown to admit unique pure strategy NE. As an extension, a computational-efficient distributed Nash equilibrium energy management strategy seeking method is proposed, which also avoids the privacy leakage due to the sharing of payoff functions between consumers. The chapter content is based on **Paper B**.

In Chapter 5, we study the data-driven calibration for the low-cost gas sensors with the same sensing and drift model as described in Section 1.1.2. Specifically, the calibration procedure for the NDIR CO_2 sensors is developed, for which the temperature dependency is the most dominant drift source. For a single sensor, the HMM is used to characterize the statistical relationship between different quantities introduced by the physical model. We further study the problem of improving the HMM to better characterize the time-varying statistical relationship between different quantities of the sensors' physical model. Instead of having a deterministic HMM, a time-varying HMM is developed to handle the drift processes of low-cost gas sensors. Correspondingly, a time-adaptive expectation maximization (EM) learning approach is proposed to efficiently update the HMM parameters, which avoids the big data storage and reduces the computational load. The chapter content

is based on **Paper C** and **Paper D**.

In Chapter 6, we focus on the joint calibration based on the gas sensor networks. We consider the scenario where sensors can transmit their belief functions of the true gas concentration level to the cloud fusion center that can compute a fused belief function (used as a reference for calibrating the sensors) according to certain rules. To deal with the case where belief functions highly conflict with each other, a Wasserstein distance based weighted average belief function fusion approach is first proposed as a networked calibration algorithm. To achieve more long-term stable calibration results, the networked calibration problem is further formulated as a POMDP problem, and the calibration strategies are derived in a sequential manner. Correspondingly, the DQN approach is applied as a computationally efficient method to solve the proposed MDP problem. The chapter content is based on **Paper C**.

Finally, the conclusions and the potential future works are provided in Chapter 7.

1.5 Copyright Notice

Chapter 2

Background Knowledge

2.1 Introduction to MDP

2.1.1 Fully Observed MDP

An Markov decision process is a discrete-time state-transition system and it provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker [70]. Assume that the process is in some state s, and the decision maker may choose any action a that is available in the action set $\mathcal{A}$. The process responds at the next time step by randomly moving into a new state s' giving the decision maker a corresponding reward $R_a(s, s')$. The probability that the process moves into its new state s' is influenced by the chosen action. Specifically, it is given by the state transition function $P_a(s, s')$. Thus, the next state s' depends on the current state s and the decision maker's action a. A typical structure of an MDP is illustrated in Figure. 2.1.

According to [70], a fully observed MDP is a 5-tuple $(S, A, P_a(s, s'), R_a(s, s'), \gamma)$:

- States $s \in \mathcal{S}$: The states will play the role of outcomes in the decision theoretic approach we saw last time, as well as providing whatever information is necessary for choosing actions.
- Action $a \in \mathcal{A}$: The action decides on the realization or the statistics of the output of decision maker, and it provides an instruction for the system operation.
- State transition probability $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$.
- Reward $R_a(s, s')$ is the reward received after transitioning from state s to state s' due to action a.
- $\gamma \in [0, 1]$ is the discount factor, which represents the difference in importance between future rewards and present rewards.

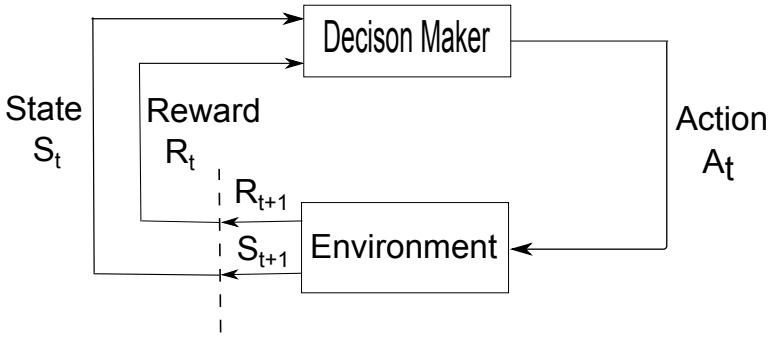


Figure 2.1: Structure of an MDP.

For a fully observed MDP, at each time step t, the decision maker observes the state s_t exactly (without noise or delay) and chooses an action a_t based on a policy f_t. A policy f is a distribution over all actions given the states and it fully defines the behavior of a decision maker, i.e.,

$$f(a|s) = \mathbb{P}[A_t = a|S_t = s]. \tag{2.1}$$

The dynamics of a fully observed MDP over a finite time horizon is described in Algorithm 1.

2.1.2 Solution to a Fully observed MDP

The goal of an MDP is to find an optimal policy $\{f_t\}_{t\geq 1}$ that maximizes or minimizes the overall objective function over either a finite horizon or an infinite horizon scenario. Here, we mainly introduce the following several different types of objective functions.

Total expected reward or cost over a finite horizon

$$\mathbb{E}\{\sum_{t=0}^{T} \gamma^t R_{a_t}(s_t, s_{t+1})\}; \tag{2.6}$$

Expected discounted sum reward or cost over an infinite horizon

$$\mathbb{E}\{\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})\}; \tag{2.7}$$

Average expected reward or cost over an infinite horizon

$$\lim_{T\to\infty} \mathbb{E}\{\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})\}, \tag{2.8}$$

Algorithm 1: Dynamics of a fully observed finite-horizon MDP

1: At time t=0, the state s_0 is generated according to the initial posterior distribution π_0.

2: For time $t > 0$, the decision maker chooses an action

$$a_t = f_t(\mathcal{I}_t) \in \mathcal{A}, \qquad t = 0, 1, ...T-1, \tag{2.2}$$

based on the available information

$$\mathcal{I}_0 = \{s_0\}, \mathcal{I}_t = \{s_0, a_0, ..., s_{t-1}, a_{t-1}, x_t\}. \tag{2.3}$$

The decision maker incurs a reward (or cost) $R_a(s, s')$ for choosing action a_t.

3: At time $t+1$, state s evolves randomly to any state s' with transition probability $P_a(s, s')$. The transition probability is defined as

$$P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a). \tag{2.4}$$

4: The decision maker updates its available information as

$$\mathcal{I}_{t+1} = \mathcal{I}_t \cup \{a_t, x_{t+1}\}. \tag{2.5}$$

If $t < T$, then set t to $t+1$ and go back to step 2.
If $t = T$, then the decision maker gains a terminal reward (or cost) $R_{a_T}(s_T)$ and the process terminates

where a_t is chosen according to $f(a_t|s_t)$. To maximize the cumulative objective function, Bellman's dynamic programming [70] can be used to solve this problem. Consider the k-th iteration, the value function is updated by

$$V_k(s) = \max_{a \in \mathcal{A}} \{R_a(s, s') + \sum_{s^{\neg ě}} \gamma P_a(s, s') V_{k+1}(s')\}, \tag{2.9}$$

where V_T is usually set to be 0. The optimal policy at time step t is given by

$$f_t^*(s) = \arg\max_{a \in \mathcal{A}} \{R_a(s, s') + \sum_{s'} \gamma P_a(s, s') V_{k+1}(s^{\neg ě})\}, \tag{2.10}$$

For any initial state s, the discounted sum reward of the optimal policy f^*, namely $V_{f^*}(s)$ is obtained as $V_0(s)$ from (2.9). The optimal deterministic Markovian policy f^* is given by $\{f_0, f_1, ..., f_{T-1}\}$.

2.1.3 Partially observed MDP

A POMDP is a 7-tuple: $(S, A, Y, P_a(s, s'), R_a(s, s'), B_a(s, y), \gamma)$

- States $s \in \mathcal{S}$: The states will play the role of outcomes in the decision theoretic approach we saw last time, as well as providing whatever information is necessary for choosing actions.
- Action $a \in \mathcal{A}$: The action decides on the realization or the statistics of the output of decision maker, and it provide an instruction for the system operation.
- Observation $y \in \mathcal{Y}$: $\mathcal{Y}$ denotes the observation space which can either be finite or a subset of $\mathbb{R}$, and y denotes the observation recorded.
- State Transition probability $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$.
- Reward $R_a(s, s')$ is the reward received after transitioning from state s to state s', due to action a.
- The observation distribution $B_a(s, y) = \mathbb{P}(y_t = y | s_t = s, a_t = a)$ is the conditional probability of observing y given state s and action a.
- $\gamma \in [0, 1]$ is the discount factor, which represents the difference in importance between future rewards and present rewards.

For a POMDP, the decision-maker does not observe the state s_t. It only observes noisy observations y_t that depends on the action and the state specified by $B_a(s, y)$. Based on the set of observations and distributions, it must maintain a probability distribution over the set of possible states, which is called belief state. The dynamics of a partially observed MDP over a finite time horizon is summarized in Algorithm 2.

2.1.4 Belief state formulation of a partially observed MDP

As it is mentioned before, for a partially observed MDP, the optimal action is chosen by the decision maker according to $a_t = f_t(\mathcal{I}_t)$, where $\mathcal{I}_t = \{\pi_0, a_0, y_1, ...a_{t-1}, y_t\}$. Since $\mathcal{I}_t$ is increasing in dimension with t, it is useful to obtain a sufficient statistic that does not grow in dimension. The posterior distribution π_t is such a sufficient statistic for $\mathcal{I}_t$. The posterior distribution of the state s_t is given by the following

$$\pi_t(s) = \mathbb{P}(s_t = s | \mathcal{I}_t), \quad s \in \mathcal{S}. \tag{2.11}$$

The vector which contains the posterior distribution of all possible realizations of $S \in \mathcal{S}$ is defined as the belief state at time t. A decision maker needs to update its belief upon taking the action a_t and observation y_t. Since the state transition is Markovian, maintaining a belief over the states solely requires knowledge of the previous belief state, the action taken, and the current observation. In this case, the belief state is updated according to the function $\pi_{t+1}(s') = T(\pi_t(s), y_{t+1}, a_t)$. In detail there is

$$T(\pi_t(s), y_{t+1}, a_t) = \frac{B_{a_t}(s', y_{t+1}) \sum_{s \in \mathcal{S}} P_{a_t}(s, s') \pi_t(s)}{\sum_{s' \in \mathcal{S}} B_{a_t}(s', y_{t+1}) \sum_{s \in \mathcal{S}} P_{a_t}(s, s') \pi_t(s)}. \tag{2.12}$$

In this case, the belief state allows a partially observed MDP to be formulated into a Markov decision process where every belief state forms a state.

Algorithm 2: Dynamics of a partially observed finite-horizon MDP .

1: At time t=0, the state s_0 is generated according to the initial posterior distribution π_0.

2: For time $t > 0$, the decision maker chooses an action

$$a_t = f_t(\mathcal{I}_t) \in \mathcal{A}, \qquad t = 0, 1, ...T-1 \tag{2.13}$$

based on the available information

$$\mathcal{I}_0 = \{\pi_0\}, \mathcal{I}_t = \{\pi_0, a_0, y_1, ...a_{t-1}, y_t\}. \tag{2.14}$$

The decision maker incurs a reward (or cost) $R_a(s, s')$ for choosing action a_t.

3: At time $t+1$, state s evolves randomly to any state s' with transition probability $P_a(s, s')$. The transition probability is defined as,

$$P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a). \tag{2.15}$$

The decision maker records a noisy observation y_{t+1} of the state s_{t+1} according to the observation distribution $B_a(s, y)$, which is defined as following

$$B_a(s, y) = \mathbb{P}(y_{t+1} = y | s_{t+1} = s, a_t = a). \tag{2.16}$$

4: The decision maker updates its available information as

$$\mathcal{I}_{t+1} = \mathcal{I}_t \cup \{a_t, y_{t+1}\}. \tag{2.17}$$

If $t < T$, then set t to $t+1$ and go back to step 2.
If $t = T$, then the decision maker gains a terminal reward (or cost) $R_{a_T}(s_T)$ and the process terminates

2.2 Introduction to HMM

In this section, we briefly introduce the basic elements as well as different learning approaches of the HMM.

2.2.1 HMM basics

Let $\{X_t\}_{t=1}^T$ denote the stochastic process that describes the hidden states, where $X_t \in \mathcal{X} = \{x_i\}_{i=1}^{i=K}$. Also denote $\{Y_t\}_{t=1}^T$ as the stochastic process of the observations of the HMM, where $Y_t \in \mathcal{Y} = \{y_j\}_{j=1}^{j=M}$. An HMM can be fully characterized by the following distributions:

- time-invariant transition probability: $X_{t+1} \sim P_{X_{t+1}|X_t,\Delta_{t+1}}$,
- time-invariant emission probability: $Y_t \sim P_{Y_t|X_t}$,
- prior distribution: $\pi_0 \sim P_{X_0}$.

For compactness, we denote the transition probability $P_{X_{t+1}|X_t}(x_{i'}|x_i)$ by $A_{ii'}$ and define $A = \{A_{ii'}\}_{i,i'}$ as the set of transition probabilities. Likewise, the emission probabilities are denoted by $B_i(j) = P_{Y_t|X_t}(y_j|x_i)$, $B = \{B_i(j)\}_{i,j}$, which is the likelihood the observation Y_t given different hidden states X_t. Lastly, we define $\pi_i = P_{X_0}(x_i)$, $\pi = \{\pi_i\}_i$, as the initial prior distribution of the hidden state.

Remark 1. *Given the above definitions, the parameters $\lambda = \{\pi, A, B\}$ fully characterize the statistical properties of an HMM.*

2.2.2 Supervised Learning of HMM

Suppose we have a dataset containing the labelled training data samples. We can then approximate the parameters $\{\pi, A, B\}$ via a simple supervised learning approach by calculating the relative frequencies of the state transitions and emissions.

Let D be the total number of data samples. The transition probability can be estimated by calculating the following relative frequency

$$A_{ii'}^{(D)} = P_{X_{t+1}|X_t}^{(D)}(x_{i'}|x_i) = \frac{count(X_{t+1} = x_{i'}, X_t = x_i)}{count(X_t = x_i)}. \tag{2.18}$$

Similarly, the emission probability can be estimated by

$$B_{ij}^{(D)} = P_{Y_t|X_t}^{(D)}(y_j|x_i) = \frac{count(Y_t = y_j, X_t = x_i)}{count(X_t = x_i)}. \tag{2.19}$$

Lastly, the prior distribution of the hidden states can be approximated by

$$\pi_i^{(D)} = P_{X_t}^{(D)}(x_i) = \frac{count(X_t = x_i)}{N}. \tag{2.20}$$

2.2.3 Unsupervised learning of HMM

2.2.3.1 Expectation Maximization

Given the observed data sequence $\bar{y}$, the main idea of the EM algorithm is to find parameters $\lambda \in \Lambda$ to maximize the log-likelihood $\ln P(\bar{y}|\lambda)$. Equivalently, we can find parameters $\lambda \in \Lambda$ to maximize the following log-likelihood written in terms of the missing data (or latent variables) X

$$\hat{\lambda} = \arg\max_{\lambda \in \Lambda} \ln \sum_{x \in \mathcal{X}} P(\bar{y}, x|\lambda), \tag{2.21}$$

where x denotes the missing data sequence and $\mathcal{X}$ denotes the missing data set. Let λ_n be the estimate of the parameters at the n-th iteration and define the following Q-function

$$Q(\lambda|\lambda_n) = \sum_{x\in\mathcal{X}} \ln P(\bar{y}, x|\lambda) P(x|\bar{y}, \lambda_n). \tag{2.22}$$

According to the Jensen's inequality [74], we can get the following lower bound on $\ln P(\bar{y}|\lambda)$.

$$\ln \sum_{x\in\mathcal{X}} P(\bar{y}, x|\lambda) \geq Q(\lambda|\lambda_n) + H(X|\bar{y}, \lambda_n), \tag{2.23}$$

where $H(X|\bar{y}, \lambda_n)$ denotes the conditional entropy of missing data (or latent variable) X given the observed data sequence $\bar{y}$ and the current parameter estimate λ_n. With the above definitions, the parameters are updated according to the following in an iteration of EM algorithm

$$\lambda_{n+1} = \arg\max_{\lambda\in\Lambda} Q(\lambda|\lambda_n). \tag{2.24}$$

Assume that the sequential parameter estimates are $\{\lambda_1, \lambda_2, ..., \lambda_n, \lambda_{n+1}, \dots\}$, the parameter estimates are updated until

$$\frac{\log P(\bar{y}|\lambda_{n+1}) - \log P(\bar{y}|\lambda_n)}{\log P(\bar{y}|\lambda_n)} < \gamma, \tag{2.25}$$

Applying the EM algorithm to HMMs leads to the well-known Baum-Welch algorithm [75]. Before presenting the details of the Baum-Welch algorithm, we first define the following quantities. Given the current estimate of parameters $\hat{\lambda} = [\hat{\pi}, \hat{A}, \hat{B}]$, let y^T be the whole observation sequence over time horizon T. The probability of seeing the partial observation sequence $(y_1, y_2, ..., y_t)$ and ending up in state i at time t is defined as

$$\alpha_i(t) = P(Y_1 = y_1, ..., Y_t = y_t, X_t = x_i|\hat{\lambda}). \tag{2.26}$$

The probability of seeing partial observation sequence $(y_{t+1}, y_{t+2}, ..., y_T)$ given in state x_i at time t is defined as

$$\beta_i(t) = P(Y_{t+1} = y_{t+1}, ..., Y_T = y_T|X_t = x_i, \hat{\lambda}). \tag{2.27}$$

Correspondingly, the probability of the hidden state being equal to x_i at time t given the whole observation sequence y^T is defined as

$$\gamma_i(t) = P(X_t = x_i|Y_1 = y_1, ..., Y_T = y_T, \hat{\lambda}), \tag{2.28}$$

which it can be further calculated by

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{j=N} \alpha_j(t)\beta_j(t)}. \tag{2.29}$$

Similarly, the probability of being in state x_i at time t and being in state x_k at time $t+1$ given the whole observation sequence y^T

$$\xi_{ik}(t) = P(X_{t+1} = x_k, X_t = x_i | Y_1 = y_1, ..., Y_T = y_T, \hat{\lambda}), \tag{2.30}$$

which can be calculated by

$$\xi_{ik}(t) = \frac{\gamma_i(t) A_{ik} B_{ij} \beta_k(t+1)}{\beta_i(t)}. \tag{2.31}$$

With the above definitions, by applying the Baum-Welch algorithm, the estimates of parameters are updated by calculating the following approximated relative frequencies.
i) Estimate of the initial prior distribution of the hidden states

$$\hat{\pi}_i(t) = \gamma_i(1). \tag{2.32}$$

ii) Estimate of the transition probability

$$\hat{A}_{ik}(t) = \frac{\sum_{t=1}^{L-1} \xi_{ik}(t)}{\sum_{t=1}^{L-1} \gamma_i(t)}. \tag{2.33}$$

iii) Estimate of the emission probability

$$\hat{B}_{ij}(t) = \frac{\sum_{t=1}^{L-1} \mathcal{I}_{Y_t}(y_j) \gamma_i(t)}{\sum_{t=1}^{L-1} \gamma_i(t)}, \tag{2.34}$$

where $\mathcal{I}$ denotes the indicator such that $\mathcal{I}_{Y_t}(y_j)$ if $Y_t = y_j, \forall t \in [1:T], j \in [1:M]$.

Then, for each iteration, the Baum-Welch algorithm updating rule is given as follows

$$\hat{\lambda}_n \xrightarrow{(2.26-2.31)} \{\xi_{ik}(t), \gamma_i(t)\} \xrightarrow{(2.32-2.34)} \hat{\lambda}_{n+1}. \tag{2.35}$$

2.3 NDIR Sensor Mechanism and Drift Analysis

The general operation principle of an NDIR sensor is illustrated in Fig. 2.2. When the sensor starts working, the broadband infrared light will be directed through the gas chamber filled by gas from the external environment towards the detector. A general wavelength range of interests for the NDIR sensing is $3 - 20\ \mu m$. However, NDIR sensing does not require light sources and detectors with a narrow spectral range of operation. As long as the source and detector cover the wavelength range of interest, i.e. the range containing the target gas absorption band of interest, an NDIR measurement can be implemented. The IR light with different wavelengths is absorbed by gas molecules in the volume probed by the light. A bandpass optical filter is used at the detector side which eliminates all light except the light with the wavelength that the target gas molecules can absorb, which provides the specificity to the target gas against other gases that might be present in the volume probed by the IR light. This principle provides the basis for specifying the sensing model (1.1).

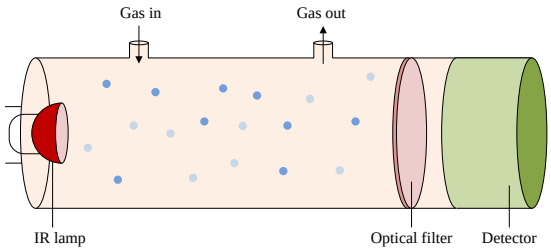


Figure 2.2: Generic working mechanism of an NDIR sensor. There are three essential components for NDIR sensing: IR light source, optical filter, and IR light detector. A general wavelength range of interests for both light source and the detector of the NDIR sensing is $3 - 20\ \mu m$. Light from the infrared lamp is absorbed by gas particles in the volume probed by the light. The intensity of remaining light is measured which provides information about the gas concentration.

In the rest of this work, we focus on the CO_2 NDIR sensors for which the temperature dependency is the most dominant effect on the behavior of the sensor components [25]. Accordingly, the CO_2 gas sensor model can be described as

$$y = g(r, i, c), \tag{2.36}$$

where c denotes the temperature, i is the current received by the detector, and r stands for the calibration parameter of the NDIR CO_2 sensor. The function $g(\cdot)$ describes the NDIR CO_2 sensor physical model based on the Beer-Lambert law [76] that maps (r, i, c) onto the CO_2 concentration level y. However, the mapping g does not perfectly capture the dependency between the behavior of the infrared light and the temperature. In this case, we want to adjust the calibration parameter r to its true value x according to the drift of the current i to compensate for the imperfections of the mapping g. To model the remaining uncertainties of the above quantities during the operating period of the sensor, we define random variables for these corresponding quantities. Specifically, we use C, I, Y, and X to denote the corresponding random variables of the environmental temperature, the current received by the detector, the CO_2 measurement, and the true calibration parameter.

Part I

Privacy-Preserving and Cost-Efficient Smart Grid Energy Management

Chapter 3

Privacy-Cost Trade-off in the Presence of an Energy Storage

In this chapter, we design privacy-enhancing and cost-efficient energy management strategies for a single consumer that is equipped with ES. The KL-divergence rate is used as privacy measure and the expected cost-saving rate is used as utility measure. The corresponding energy management strategy is designed by optimizing a weighted sum of both privacy and cost measures over a finite time horizon, which is achieved by formulating our problem into a POMDP problem. A computationally efficient approximated Q-learning method is proposed as a generalization to high-dimensional problems over an infinite time horizon. At last, we explicitly characterize a stationary policy that achieves the steady belief state over an infinite time horizon, which greatly simplifies the design of the privacy-preserving energy management strategy. The content of this chapter has been taken from **Paper A**, while some parts have been verbatim copied.

This chapter is organized as follows: the system model is first presented in Section 3.1. Section 3.2 introduces the POMDP problem formulation of the corresponding energy management design problem. Correspondingly, the solutions (for both finite time horizon and infinite time horizon) to the design problem are proposed in Section 3.2 and Section 3.4. Section 3.5 studies the privacy-preserving problem under a special case where the energy demand is assumed to be i.i.d, and a structural stationary energy management strategy is proposed as a simplified solution. Finally, numerical results are given in Section 3.6 and we conclude this chapter in Section 3.7.

3.1 System Model

Consider a smart metering system as shown in Fig. 3.1. The consumer's privacy-sensitive behavior over a certain time period T is modeled by a binary hypothesis h_0 or h_1. Under each hypothesis, the consumer will have a certain energy consumption profile. Time and energy levels are assumed to be discretized. At time step t, we denote consumers'

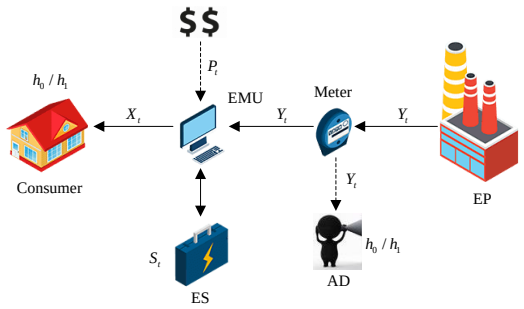


Figure 3.1: Smart metering system with ES and privacy-cost-aware energy management unit (EMU) that modifies energy consumption profile to protect against unauthorized hypothesis testing (h_0/h_1) of adversary taking dynamic energy prices into account.

energy demand by $x_t \in \mathcal{X} = \{0, 1, ..., x_{max}\}$, energy supply from the EP by $y_t \in \mathcal{Y} = \{0, 1, ..., y_{max}\}$, instantaneous price by $p_t \in \mathcal{P} = \{1, ..., p_{max}\}$. The energy storage (ES), e.g., a rechargeable battery, has a finite capacity with its instantaneous storage level denoted by $s_t \in \mathcal{S} = \{0, 1, ..., s_{max}\}$. The instantaneous energy consumption x_t should always be satisfied by supplies from either EP or ES without wasting energy. Then, the ES level evolves as

$$s_{t+1} = s_t + y_t - x_t. \tag{3.1}$$

In addition, to guarantee $0 \leq s_t \leq s_{max}$, the energy supply y_t should be chosen within the following feasible set

$$\overline{\mathcal{Y}}(x_t, s_t) = \{y_t \in \mathcal{Y} : \max\{0, x_t - s_t\} \leq y_t \leq s_{max} + x_t - s_t\}, \tag{3.2}$$

where the lower bound $x_t - s_t$ ensures that the energy supply y_t provides at least the rest energy when ES level s_t cannot solely satisfy the consumer demand; the lower bound 0 is because no energy can be sold back to the grid; and the upper bound is due to the constraints of finite maximum ES capacity and that no energy should be wasted.

We assume that the consumer energy demand X_t and the dynamic price P_t follow first order Markov processes with time-invariant transition probabilities $P_{X_{t+1}|X_t,h_i}, i \in \{0, 1\}$, and $P_{P_{t+1}|P_t}$. Over a T-time horizon, the EMU requests energy supply Y_t from the EP based on an energy management strategy $f = \{f_t\}_{t=1}^{T} \in \mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times .... \times \mathcal{F}_T$, with $f_t \in \mathcal{F}_t$. The set $\mathcal{F}_t$ of the possible strategies is given by the set of pmfs

$$\mathcal{F}_t = \{P_{Y_t|X^t,S^t,P^t,Y^{t-1}} : \sum_{\substack{y_t \in \overline{\mathcal{Y}}_t(x_t,s_t) \\ \forall x_t \in \mathcal{X}, s_t \in \mathcal{S}}} P(y_t|x^t, s^t, p^t, y^{t-1}) = 1\}. \tag{3.3}$$

For $i \in \{0, 1\}$, after initializing the joint pmf of X_1, S_1 and P_1 as $P_{X_1,S_1,P_1|h_i}$, over a finite horizon with length T, the joint conditional pmf of (X^T, S^T, Y^T, P^T) induced by f

can be written as

$$
\begin{aligned}
P^f_{X^T,S^T,Y^T,P^T|h_i}(x^T,s^T,y^T,p^T) = & \underbrace{P_{X_1,S_1,P_1|h_i}(x_1,s_1,p_1)}_{Initialization} \times \underbrace{P(y_1|x_1,s_1,p_1)}_{f_1(y_1|x_1,s_1,p_1)} \\
& \prod_{t=1}^{T-1} [\ \underbrace{P(p_{t+1}|p_t)}_{Price\ evolution} \times \underbrace{P(x_{t+1}|x_t,h_i)}_{Demand\ evolution} \\
& \times \underbrace{\mathcal{I}_{s_{t+1}}(y_t+s_t-x_t)}_{Energy\ storage\ level\ evolution} \\
& \times \underbrace{P(y_{t+1}|x^{t+1},s^{t+1},p^{t+1},y^t)}_{f_{t+1}(y_{t+1}|x^{t+1},s^{t+1},p^{t+1},y^t)}],
\end{aligned}
\tag{3.4}
$$

where $\mathcal{I}$ is the indicator function, i.e., $\mathcal{I}_{s_{t+1}}(y_t+s_t-x_t)=1$ if $s_{t+1}=y_t+s_t-x_t$, $\mathcal{I}_{s_{t+1}}(y_t+s_t-x_t)=0$, otherwise.

Assume that an adversary has access to the smart metering data sequence y^T, price sequence p^T, and is fully informed about the statistics of the system, i.e., $P_{Y^T,P^T|h_0}$ and $P_{Y^T,P^T|h_1}$. The adversary infers on the consumers' privacy-sensitive consumption behavior using statistical inference methods. Due to the uncertainty about the inference behavior of the adversary, we use the KL-divergence rate as privacy leakage measure, since the KL-divergence measures the similarity between two distributions. Over a finite time horizon T, given a strategy $f \in \mathcal{F}$, the privacy leakage is measured by the following KL-divergence rate between joint pmfs of (Y^T,P^T) conditioned on hypotheses h_0 and h_1

$$
L_T(f)=\frac{1}{T}D(P^f_{Y^T,P^T|h_0}\|P^f_{Y^T,P^T|h_1}), \tag{3.5}
$$

where $P^f_{Y^T,P^T|h_i}$, for $i=0,1$, denotes the joint distribution of (Y^T,P^T) conditioned on h_i induced by f

$$
P^f_{Y^T,P^T|h_i}(y^T,p^T)=\sum_{x^T,s^T} P^f_{X^T,S^T,Y^T,P^T|h_i}(x^T,s^T,y^T,p^T). \tag{3.6}
$$

We define the cost-saving at time t as $\Delta V_t=(X_t-Y_t)P_t$. The expected cost-saving rate induced by f over a finite horizon T can then be written as

$$
V_T(f)=\frac{1}{T}\sum_{t=1}^{T}(\mathbb{E}[\Delta V_t|h_0]P(h_0)+\mathbb{E}[\Delta V_t|h_1]P(h_1)), \tag{3.7}
$$

where the expectation is taken with respect to the joint conditional distribution $P^f_{X_t,Y_t,P_t|h_i}$, for $i=\{0,1\}$, induced by f. For the privacy-cost trade-off problem, the overall objective is to choose a strategy $f\in\mathcal{F}$ that minimizes the following weighted sum objective

$$
C_T(f,\lambda)=\lambda L_T(f)-(1-\lambda)V_T(f), \tag{3.8}
$$

where $\lambda \in [0,1]$. In more detail, the trade-off between privacy and cost is realized by choosing different values of λ, e.g., $\lambda = 1$ leads to finding the optimal privacy-enhancing strategy, while $\lambda = 0$ leads to the objective function of finding the optimal cost-saving strategy. Then, the optimal strategy is

$$f^* = \arg\min_{f \in \mathcal{F}} C_T(f, \lambda). \tag{3.9}$$

3.2 POMDP Problem Formulation

Applying the chain rule of KL-divergence, $L_T(f)$ can be written in the following form

$$\begin{aligned} L_T(f) =& \frac{1}{T}\sum_{t=1}^{T} D(P^f_{Y_t,P_t|h_0,Y^{t-1},P^{t-1}} \| P^f_{Y_t,P_t|h_1,Y^{t-1},P^{t-1}}) \\ =& \frac{1}{T}\sum_{t=1}^{T}\sum_{y^t}\sum_{p^t} P^f(y^t,p^t|h_0)\log\frac{P^f(y_t,p_t|h_0,y^{t-1},p^{t-1})}{P^f(y_t,p_t|h_1,y^{t-1},p^{t-1})}. \end{aligned} \tag{3.10}$$

Meanwhile, the $V_T(f)$ can be expressed by

$$V_T(f) = \frac{1}{T}\sum_{t=1}^{T}\sum_{x_t,p_t,y_t} (P(h_0)P^f(x_t,p_t,y_t|h_0) + P(h_1)P^f(x_t,p_t,y_t|h_1))(x_t p_t - y_t p_t). \tag{3.11}$$

The above equations imply that the current choice of the energy management strategy will affect the future statistics of the smart metering system as well as the choices of the future energy management strategies. Thus, successive independent optimization of the energy management strategy at each time step will not necessarily lead to an optimal solution over the whole time horizon. Instead, our optimization problem needs to be formulated into a sequential decision making problem, e.g., Markov decision process.

We first identify a structural simplification for the energy management strategy f. Define a new strategy $f' = \{f'_t\}_{t=1}^T \in \mathcal{F}' = \mathcal{F}'_1 \times \mathcal{F}'_2 \times .... \times \mathcal{F}'_T$, with $f'_t \in \mathcal{F}'_t$. And $\mathcal{F}'_t$ is the set of pmfs that

$$\mathcal{F}'_t = \{P_{Y_t|X_t,S_t,P^t,Y^{t-1}} : \sum_{\substack{y_t \in \overline{\mathcal{Y}}_t(x_t,s_t) \\ \forall x_t \in \mathcal{X}, s_t \in \mathcal{S}}} P(y_t|x_t,s_t,p^t,y^{t-1}) = 1\}. \tag{3.12}$$

By showing the probabilities terms in objective function (3.8) remain identical when they are induced by either f' or f, we have the following proposition.

Proposition 1. *For the optimization problem proposed in (3.9), there is no loss of optimality by focusing only on strategies in $\mathcal{F}'$. And the equivalent optimization problem is*

$$f'^* = \arg\min_{f' \in \mathcal{F}'} C_T(f', \lambda). \tag{3.13}$$

Proof: The proof is provided in the Appendix A.1.

We next transform problem (3.13) into an MDP. At time step t, let the control action be $a_t \in \mathcal{A}$ which is the condition pmf $P_{Y_t|X_t,S_t,P_t}$ taken from the following set

$$\mathcal{A} = \{P_{Y|X,S,P} : \sum_{y \in \overline{\mathcal{Y}}(x,s)} P(y|x,s,p) = 1, \forall x \in \mathcal{X}, s \in \mathcal{S}\} \tag{3.14}$$

Thus, the control action a_t randomly decides on the energy supply Y_t according to the conditional pmf $P_{Y_t|X_t,S_t,P_t}$. Note that in this section we assume that the EMU is unaware of the hypothesis h_0 or h_1. Thus, the EMU will choose an action according to a hypothesis independent policy $\pi_t \in \mathit{\Pi}_t$, where $\mathit{\Pi}_t$ denotes the set of deterministic mappings from the historical observations (y^{t-1}, p^{t-1}) to a corresponding action a_t, i.e., $a_t = \pi_t(y^{t-1}, p^{t-1})$. Thus, the policy over a T-time horizon is $\pi = \{\pi_t\}_{t=1}^{T} \in \mathit{\Pi} = \mathit{\Pi}_1 \times \mathit{\Pi}_2 \times .... \times \mathit{\Pi}_T$. At each time step, any energy management strategy $f_t^{'}$ can be equivalently represented by a combination of a policy π_t and a control action a_t. For each $f_t^{'}$, we apply the policy π_t to establish a mapping from the historical sequence (y^{t-1}, p^{t-1}) to $a_t = P^{f_t^{'}}_{Y_t|X_t,S_t,P_t,Y^{t-1}=y^{t-1},P^{t-1}=p^{t-1}}$. This control action a_t is then used to obtain Y_t according to $P_{Y_t|X_t,S_t,P_t}$. Thereby, we end up with control action that itself does not depend on the historical sequence (y^{t-1}, p^{t-1}), while the history decides which control action is applied at time t. For reason of simplicity, let $Q_t = (Y^t, P^t, A^t)$, the expression $a_t = \pi_t(q_{t-1})$ is then equivalent to $a_t = \pi_t(y^{t-1}, p^{t-1})$ since (y^{t-1}, p^{t-1}) fully determines a^t.

With the above definition, we obtain an equivalent reformulation of the problem (3.13) as stated in the following proposition.

Proposition 2. *The optimization problem in Proposition 1 is equivalent to finding a policy $\pi \in \mathit{\Pi}$ that minimizes the following weighted sum objective:*

$$C_T(\pi, \lambda) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t(\pi_t, \lambda, Q_{t-1})], \tag{3.15}$$

where the per-step expected cost conditioned on each possible historical sequence q_{t-1} can be specified as

$$\begin{aligned} C_t(\pi_t, \lambda, q_{t-1}) = \lambda \sum_{p_t, y_t} P^{\pi_t}(y_t, p_t | q_{t-1}, h_0) \log \frac{P^{\pi_t}(y_t, p_t | q_{t-1}, h_0)}{P^{\pi_t}(y_t, p_t | q_{t-1}, h_1)} \\ - (1-\lambda) \sum_{x_t, p_t, y_t} (P(h_0) P^{\pi_t}(x_t, p_t, y_t | q_{t-1}, h_0) \\ + P(h_1) P^{\pi_t}(x_t, p_t, y_t | q_{t-1}, h_1))(x_t p_t - y_t p_t). \end{aligned} \tag{3.16}$$

And the optimal policy is given by $\pi^* = \arg\min_{\pi \in \mathit{\Pi}} C_T(\pi, \lambda)$.

Proof: To prove this proposition, we need to show $C_T(\pi, \lambda)$ is equal to $C_T(f^{'}, \lambda)$ under transition from strategy $f^{'}$ to policy π. Thus, we need to further show that the probability

terms in these two objective functions are equal. Since the proofs for the other probability terms are similar, we only prove $P^{f^{'}}_{Y^T,P^T|h_i} = P^{\pi}_{Y^T,P^T|h_i}$ *here.*

After expanding $P^{\pi}_{Y^T,P^T|h_i}$ *according to the policy* π*, we obtain*

$$\begin{aligned} &P^{\pi}{}_{Y^T,P^T|h_i}(y^T,p^T) \\ &\quad= \sum_{x^T,s^T} P_{X_1,S_1,P_1|h_i}(x_1,s_1,p_1)\times a_1(y_1|x_1,s_1,p_1,h_i) \\ &\qquad \prod_{t=1}^{T-1}[P(p_{t+1}|p_t)P(x_{t+1}|x_t,h_i)\mathcal{I}_{s_{t+1}}(y_t+s_t-x_t) \\ &\qquad\quad \times a_{t+1}(y_{t+1}|x_{t+1},s_{t+1},p_{t+1})]. \end{aligned} \tag{3.17}$$

By applying the equivalence between $f_t^{'}$ *and* (π_t, a_t)*, which is derived above, we get* $P^{f_t^{'}}(y_{t+1}|x_{t+1},s_{t+1},p^{t+1},y^t) = a_t(y_{t+1}|x_{t+1},s_{t+1},p_{t+1})$*. Also, we have* $P^{f_t^{'}}(y_1|x_1,s_1,p_1) = a_1(y_1|x_1,s_1,p_1)$*. Thus,* $P^{f^{'}}_{Y^T,P^T|h_i} = P^{\pi}_{Y^T,P^T|h_i}$ *holds.* □

In contrast to a standard MDP problem, as shown in (3.16), the per-step conditional expected cost will depend on not only the current state and control action but also the historical sequence q_{t-1}. In order to formulate it into a standard MDP, we introduce belief states that will be used to replace the growing historical sequences, which leads to a POMDP problem formulation. To this end, we define a belief state $\theta_{q_{t-1}} = (\theta^0_{q_{t-1}}, \theta^1_{q_{t-1}})$ as the posterior distributions of (X_t, S_t, P_t) conditioned on the realization q_{t-1} under the corresponding hypotheses as

$$\theta^i_{q_{t-1}} = P_{X_t,S_t,P_t|q_{t-1},h_i}, i \in \{0,1\}. \tag{3.18}$$

Thus, $C_t(\pi_t, \lambda, q_{t-1})$ can be expressed in terms of the corresponding belief state-action pairs, i.e., we have

$$P^{\pi_t}(x_t,p_t,y_t|q_{t-1},h_i) = \sum_{s_t} \theta^i_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t). \tag{3.19}$$

At time step t, for $i \in \{0,1\}$, given any $\theta^i_{q_{t-1}}$, observation y_t, and action a_t, the updating of belief state is given by $\theta^i_{q_t} = \varphi(\theta^i_{q_{t-1}}, a_t, y_t)$ in (3.20).

According to the definition of belief-state MDP in [70, pp.150-151], the above formulations and derivations provide the proof for the following theorem.

Theorem 1. *The original optimization problem in (3.13) can be modeled as a belief-state MDP problem such that: (i) the state at time* t *is given by (3.18) and evolves according to (3.20); (ii) the control action at time* t *is specified by* $a_t(y_t|x_t,s_t,p_t)$*; (iii) the per-step expected cost corresponding to a state-action pair is given by (3.16); (iv) the optimal policy* π *can be derived by using Bellman dynamic programming.*

$$\varphi(\theta^i_{q_{t-1}}, a_t, y_t) = \frac{\sum\limits_{x_t,s_t,p_t} \theta^i_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)P(p_{t+1}|p_t)P(x_{t+1}|x_t,h_i)\mathcal{I}_{s_{t+1}}(y_t+s_t-x_t)}{\sum\limits_{x_t,s_t,p_t} \theta^i_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)} \tag{3.20}$$

Remark 2. *In the reformulated belief-state MDP problem, at each time step, the decision maker observes the historical sequence q_{t-1} and identifies a unique belief state $\theta_{q_{t-1}}$. Based on this belief state, the decision maker will further decide on an action according to the optimal strategy π_t derived from the Bellman dynamic programming.*

3.3 Bellman Dynamic Programming based Energy Management Strategy Design over Finite Time Horizon

In this section, we will provide the optimal solution for the proposed privacy-cost trade-off problem over a finite time horizon.

Lemma 1. *For any action a_t, according to [70, pp. 152-153], the modified Bellman operator B_{a_t} for our belief-state MDP problem can be written as*

$$(B_{a_t}V)(\theta_{q_{t-1}}) = C_t(\pi_t,\lambda,q_{t-1}) + \sum_{y_t}[(\sum_{i=0,1}\sum_{x_t,s_t,p_t} P(h_i)\theta^i_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t))V(\varphi(\theta_{q_{t-1}},a_t,y_t))], \tag{3.21}$$

where V denotes the value function. The first term is the per-step expected cost for any given belief state $\theta_{q_{t-1}}$ and the second part denotes the corresponding expected cost-to-go.

Proof: For the traditional belief state MDP with one belief state variable, which is described in [70], with an abuse of notation, the Bellman operator can be written as:

$$(B_aV)(b) = r(b,a) + \sum_{y\in\mathcal{Y}} P(y|b,a)V(\varphi(b,a,y)), \tag{3.22}$$

where b is the current belief state, a is the corresponding action, $r(b,a)$ denotes the per-step reward, and $\varphi(b,a,y)$ denotes the evolution of the belief state given a specific action a and a specific observation y. Most importantly, the term $P(y|b,a)$ denotes the probability of observing y at belief state b given a specific action a, i.e., the transition probability between belief state b and belief state $\varphi(b,a,y)$ given any action a. For our problem, the belief state $\theta_{q_{t-1}} = (\theta^0_{q_{t-1}}, \theta^1_{q_{t-1}})$ is a vector that contains two beliefs conditioned either on hypothesis h_0 or h_1. In this case, with the given prior of the hypotheses $P(h_0)$ and $P(h_1)$, the probability for observing y_t at belief state $\theta_{q_{t-1}}$ given action a_t can be then

calculated by:

$$\begin{aligned} P(y_t|a_t,\theta_{q_{t-1}}) = P(h_0) \sum_{x_t,s_t,p_t} (\theta^0_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)) \\ + P(h_1) \sum_{x_t,s_t,p_t} (\theta^1_{q_{t-1}}(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)), \end{aligned} \tag{3.23}$$

Plugging the above equation into (3.22) will lead to our modified Bellman operator as defined in (3.21). □

In this case, the value function is updated according to

$$V(\theta_{q_{t-1}}) = \min_{a_t \in \mathcal{A}_t} (B_{a_t} V)(\theta_{q_{t-1}}). \tag{3.24}$$

The optimal policy $\pi^*_t(\theta_{q_{t-1}})$ and the corresponding optimal control action $a^*_t = \pi^*_t(\theta_{q_{t-1}})$ is given by the optimizer of (3.24). Let θ_1 denote the initial joint distributions of (X_1, S_1, P_1) conditioned on h_0 and h_1, then the minimum value of average expected cost C_T is given by $V(\theta_1)/T$.

3.4 Q-learning based Energy Management Strategy Design over Infinite Time Horizon

3.4.1 Optimization over Infinite Time Horizon

In this section, we provide the solutions to the proposed privacy-cost trade-off problem over infinite time horizon, i.e., $T \to \infty$. For the reason of simplicity, we use $(s, a) \in \mathcal{S} \times \mathcal{A}$ to denote the belief state and action pair, $c(s, a)$ and $P(s'|s, a)$ as the cost and transition probability to state s' from the corresponding state-action pair (s, a). Under the infinite time horizon, the optimal Bellman equation of our average expected cost MDP is given by

$$h^*(s) = \min_{a \in \mathcal{A}} [c(s,a) - \rho^* + \sum_{s'} P(s'|s,a) h^*(s')], \tag{3.25}$$

where ρ^* denotes the optimal average expected cost of the optimal policy, and $h(s)$ denotes the relative value function, i.e., the asymptotic difference between the total expected cost of starting from state s and the total expected cost that would be incurred if the per-step cost is equal to ρ^* for all states. In the following, we assume that the optimal stationary policy exists for our infinite horizon average reward MDP problem. To guarantee the existence of stationary optimal policy, we need to have some restrictions on the underlying Markov chains. For instance, the Markov chain induced by any policy should be unichain. However, the problem of checking such a unichain condition is NP-hard [77]. We thus assume that there exists an optimal stationary policy for our infinite horizon average MDP problem. For our numerical experiments, we can see that our relative iteration algorithm convergences under our discretized state-action space settings, which indicates that the optimal stationary policy exists under this specific setting.

The optimal policy for the above problem $\pi^* = (\pi, \pi, ...)$ is stationary and is defined by

$$\pi(s) = \arg\min_{a\in\mathcal{A}}[c(s,a) + \sum_{s'} P(s'|s,a)h^*(s')], \quad \forall s. \tag{3.26}$$

Given state and action sets with small cardinalities, then the relative value iteration (RVI) method [78] can guarantee a fast convergence of the above operator if an optimal stationary solution exists. However, when the cardinality of the set increases, the system dynamics become impractical to characterize, i.e., cost function and transition behavior corresponding to each state-action cannot be fully characterized. Thus, the exact solution methods such as RVI will be inapplicable.

To address this issue, we first propose to use the Q-learning algorithm, since the reinforcement learning algorithms could help to solve the MDP problem without the knowledge of the cost function and the transition probabilities. Since it is infeasible to explicitly represent the Q-function over the continuous belief state space,[1] a general function approximator is used to approximate the Q-function given each possible state. In more detail, we provide the framework of relative Q-learning with linear function approximation as a suboptimal but computationally more efficient solution to our original optimization problem.

3.4.2 Q-learning Based Stationary Energy Management Strategy Design

In the following, we first provide a brief outline of the relative Q-learning method. More details on this method can be found in [72]. The main contribution here is the linear functional approximation and the corresponding feature selection that results in a good performance of our approach.

Given the optimal relative value function $h^*(s)$ in (3.25), we define the optimal Q-function $Q^*(s,a)$ as the minimum asymptotic difference between total expected cost of starting from state s with action a and the optimal total expected cost

$$Q^*(s,a) = c(s,a) - \rho^* + \sum_{s'} P(s'|s,a)h^*(s'). \tag{3.27}$$

Since $h^*(s) = \min_{a\in\mathcal{A}} Q^*(s,a)$, we have

$$Q^*(s,a) = c(s,a) - \rho^* + \sum_{s'} P(s'|s,a) \min_{b\in\mathcal{A}} Q^*(s',b). \tag{3.28}$$

Further we define an operator H as

$$(HQ)(s,a) = c(s,a) - \rho^* + \sum_{s'} P(s'|s,a) \min_{b\in\mathcal{A}} Q(s',b), \tag{3.29}$$

the optimal Q-function then becomes a fixed point of operator H. According to the Robbins-Monro algorithm [79], the optimal Q-function can be learned by utilizing the

[1] For the reason of simplicity, we restrict our problem to the case with infinite state space but a finite action space, i.e., the action takes values from a finite subset of the continuous action set $\mathcal{A}$.

temporal difference between the new estimate and the old estimate, which is given by the following

$$\begin{aligned} Q_{n+1}(s,a) &= Q_n(s,a) + \alpha[c(s,a) - \rho^* + \min_{b\in\mathcal{A}} Q_n(s',b) - Q_n(s,a)] \\ &= (1-\alpha)Q_n(s,a) + \alpha[c(s,a) - \rho^* + \min_{b\in\mathcal{A}} Q_n(s',b)], \end{aligned} \tag{3.30}$$

where $\alpha \in (0,1]$ denotes the learning rate, which can be kept as constant during the learning process. The new estimate term $c(s,a) - \rho^* + \min_{b\in\mathcal{A}} Q_n(s',b)$ is sampled in the system by executing an action a selected by ϵ-greedy policy[2] which results in state s'. Note that the optimal gain ρ^* is unknown in advance. Thus, we introduce the following relative Q-function iteration to overcome this problem.

First, we select an arbitrary state-action pair $(\hat{s},\hat{a})$ before the algorithm starts, this state-action pair is fixed and acts as the reference state-action pair in each iteration until the algorithm ends. The Q-function corresponding to each possible state action pair (s,a) can be updated by

$$\begin{aligned} &Q_{n+1}(s,a) \\ &\quad = (1-\alpha)Q_n(s,a) + \alpha[c(s,a) - Q_n(\hat{s},\hat{a}) + \min_{b\in\mathcal{B}} Q_n(s',b)]. \end{aligned} \tag{3.31}$$

It has been shown in [78] that as $n\to\infty$, the sequence $\left(Q_n(\hat{s},\hat{a})\right)_n$ will converge to ρ^*. As a result, this algorithm will converge to the fixed point of (3.28).

Remark 3. *The computational complexity of the RVI algorithm is $\mathcal{O}(|S|^2|A|)$ per iteration and $\mathcal{O}(T|S|^2|A|)$ overall [72], where T denotes the number of iterations to converge, $|S|$ and $|A|$ denote the cardinalities of state and action spaces. Meanwhile, in the above relative Q-learning algorithm, the computational complexity is only $\mathcal{O}(|A|)$ per iteration and $\mathcal{O}(N|A|)$ overall, where N is the number of iterations to converge. The Q-learning algorithms usually need a higher number of iterations to converge than RVI, i.e., $N > T$. However, when we have large state or action spaces, i.e., large $|S|$ or $|A|$, the Q-learning algorithms will lead a significant reduction of the computational complexity .*

3.4.2.1 Linear function approximation

Since our belief state space is continuous, the number of states to learn is infinite so that an explicit characterization of each $Q(s,a)$ is infeasible. For this reason, we propose to use the function approximation method to avoid the explicit characterization of the Q-function. In more detail, we use the following linear function $\hat{Q}(s,a)$ to approximate the Q-function $Q(s,a)$, since it is simple for mathematical analysis and it can inherit the useful convergence results from different kinds of learning systems [72]. Assume we have a finite

[2] In reinforcement learning scenario, under an ϵ-greedy policy, the agent chooses the best action with probability $1-\epsilon$ and randomly chooses an action with probability ϵ.

action set $\mathcal{A}' \subset \mathcal{A}$ with a small cardinality, it is then practical to represent $\hat{Q}(s,a)$ by the following weighted sum of different features of state s

$$\hat{Q}(s,a) = \sum_{i=1}^{N} w_i(a) f_i(s), \tag{3.32}$$

where $f_i(s)$ for $i = 1, 2, ..., N$ are N feature functions corresponding to each possible belief state s; and $w_i(a)$ for $i = 1, 2, ..., N$ are weights for different features given each possible action a. More details on how to select the features will be discussed later. Let M denote the cardinality of the finite action set $\mathcal{A}$. We thus transfer our original task of learning $Q(s,a)$ for an infinite number of state action pairs into the task of learning M different weight vectors, where each vector is of length N, i.e., $w(a) = [w_1(a), w_2(a), ..., w_N(a)]$ denotes the weight vector corresponding to action a.

Using the update rule of the Q-function $Q_n(s,a)$ in (3.31), we define the temporal difference between new estimate and old estimate as follows

$$\Delta Q_n(s,a) = c(s,a) - Q_n(\hat{s},\hat{a}) + \min_{b \in \mathcal{A}} Q_n(s',b) - Q_n(s,a). \tag{3.33}$$

Equation (3.31) will converge to the optimal Q-function, when we reduce the magnitude of $\Delta Q_n(s,a)$, i.e., $\left(Q_n(\hat{s},\hat{a})\right)_n \to \rho^*$ as $\Delta Q_n(s,a) \to 0$, when $n \to \infty$. By applying the same underlying idea and substituting the Q-function with its linear function approximation (3.32), it has been shown in [80] that the optimal linear approximator, which satisfies (6.22), can be obtained by solving

$$\min_{w(a), \forall i} \mathbb{E}(\Delta \hat{Q}_n(s,a))^2, \tag{3.34}$$

where $\Delta \hat{Q}_n(s,a)$ denotes the the temporal difference between new estimate and old estimate when the Q-function is approximated by $\hat{Q}$

$$\Delta \hat{Q}_n(s,a) = c(s,a) - \hat{Q}_n(\hat{s},\hat{a}) + \min_{b \in \mathcal{A}} \hat{Q}_n(s',b) - \hat{Q}_n(s,a). \tag{3.35}$$

To find the optimal solutions to (3.34), the stochastic gradient descent method [72] is used to update the weights $w(a)$ for all actions $a \in \mathcal{A}$. The updating rule is then given by

$$w_i(a) = w_i(a) + \alpha \Delta \hat{Q}_n(s,a) f_i(s), \forall i \in \{1, 2, ...N\}, \tag{3.36}$$

where $\alpha \in (0, 1]$ is the diminishing learning rate.

Remark 4. *Given the convergence of the above algorithm, the optimal strategy can be obtained by*

$$\pi^*(s) = \arg\min_{a \in \mathcal{A}} \hat{Q}^*(s,a), \tag{3.37}$$

where $\hat{Q}^*$ is the optimal approximated Q-function.

3.4.2.2 Feature selection

In order to have a linear function $\hat{Q}(s,a)$ that can well approximate $Q(s,a)$, it is important to choose appropriate features $f_i(s)$ for the linear approximator. Given the Q-function defined by (3.27), we propose the following heuristic approach where we select features that describe well the per-step cost function $c(s,a)$ in the linear form (3.32). We first expand the per-step cost function (3.16) as follows

$$\begin{aligned} C_t(\pi_t,\lambda,q_{t-1}) &= \lambda \sum_{p_t,y_t} P^{\pi_t}(y_t,p_t|q_{t-1},h_0)\log P^{\pi_t}(y_t,p_t|q_{t-1},h_0) \\ &\quad - \lambda \sum_{p_t,y_t} P^{\pi_t}(y_t,p_t|q_{t-1},h_0)\log P^{\pi_t}(y_t,p_t|q_{t-1},h_1) \\ &\quad - (1-\lambda) \sum_{x_t,p_t,y_t} P(h_0)P^{\pi_t}(x_t,p_t,y_t|q_{t-1},h_0)(x_t p_t - y_t p_t) \\ &\quad - (1-\lambda) \sum_{x_t,p_t,y_t} P(h_1)P^{\pi_t}(x_t,p_t,y_t|q_{t-1},h_1)(x_t p_t - y_t p_t). \end{aligned} \tag{3.38}$$

Let the cardinality of the energy supply set $\mathcal{Y}$ be K, and the cardinality of price set $\mathcal{P}$ be L. Also let the operator $|\cdot|$ denotes the L_2 norm. For any belief state (θ^0,θ^1)[3] and price p_i, define θ_i^0 as the vector which contains elements $[\theta^0(x,s,p_i)]_{(x,s)\in\mathcal{X}\times\mathcal{Y}}$ and θ_i^1 as the vector with elements $[\theta^1(x,s,p_i)]_{(x,s)\in\mathcal{X}\times\mathcal{S}}$. Further, let a_i^j be the vector with elements $[a_t(y_j|x,s,p_i)]_{(x,s)\in\mathcal{X}\times\mathcal{S}}$. Let ϕ_i^j be the angle between vector a_i^j and θ_i^0, and ψ_i^j be the angle between vector a_i^j and θ_i^1. By finding a feature-action representation for each term of (3.38), the features given any action are characterized in the following proposition.

Proposition 3. *For any belief state (θ^0,θ^1) and action a, by doing a decomposition of (3.38), the corresponding heuristic feature selection is characterized as follows*

$$\begin{aligned} &f_1((\theta^0,\theta^1)) = 1,\ f_2((\theta^0,\theta^1)) = |\theta^0|,\ f_3((\theta^0,\theta^1)) = |\theta^1|, \\ &f_4((\theta^0,\theta^1)) = \sqrt{\sum_i^L (|\theta_i^0|\log|\theta_i^0|)^2},\ f_5((\theta^0,\theta^1)) = \sqrt{\sum_i^L (|\theta_i^0|\log|\theta_i^1|)^2}. \end{aligned} \tag{3.39}$$

Proof: The proof is provided in the Appendix A.2.

Remark 5. *The intuition behind the above derivation is to decompose the cost function $C_t(\pi_t,\lambda,q_{t-1})$ to identify the features that is only related to the belief state, and the result above shows the cost function can be written as a linear combination of these features. With this result, one can conclude that these features will be highly relevant to the value of the cost function.*

[3] For reason of simplicity, we use (θ^0,θ^1) as short notation for $(\theta^0_{q_{t-1}},\theta^1_{q_{t-1}})$.

3.4.2.3 Approximated relative Q-learning

Based on the above discussion, we summarize our proposed linear function approximated relative Q-learning (LARQL) in Algorithm 3.

Algorithm 3: Approximated relative Q-learning

Input: Belief states and actions
Output: The optimal strategy
Initialization: Initialize α, ϵ, $w(a), \forall a \in \mathcal{A}'$, the reference state-action pair $(\hat{s}, \hat{a})$, the initial belief state s_1.
Calculate the corresponding features of s_1.
for *n=1:T* **do**
  Select action a_n using ϵ-greedy policy;
  Execute action a_n;
  Observe a new state s_{n+1} and the per-step cost $c(s_n, a_n)$;
  Calculate the corresponding features of s_{n+1};
  Update $w(a_n)$ by using (3.36);
end
Find the optimal strategy by using (3.37).

3.5 Privacy-Preserving Under i.i.d Energy Demand

3.5.1 System Model

In this section, we consider privacy-preserving problem in the system shown in Fig. 3.2. Assume the energy demand $x_t \in \mathcal{X} = \{0, 1, ..., x_{max}\}$ is i.i.d. with distribution $P_X^0(x)$ under hypothesis h_0, and $P_X^1(x)$ under h_1 respectively. The energy supply $y_t \in \mathcal{Y} = \{0, 1, ..., y_{max}\}$ and battery level $s_t \in \mathcal{S} = \{0, 1, ..., s_{max}\}$ also satisfy the physical constraint described in (3.1) and (3.2). We further assume that the EMU knows the consumer's behavior, and design the corresponding energy management strategies under different energy consumption behaviors: $f^{(i)} = \{f_t^{(i)}\}_{t=1}^T \in \mathcal{F}^{(i)} = \mathcal{F}_1^{(i)} \times \mathcal{F}_2^{(i)} \times .... \times \mathcal{F}_T^{(i)}$, with $f_t^{(i)} \in \mathcal{F}_t^{(i)}$. $\mathcal{F}_t^{(i)}$ denotes the set of pmfs that

$$\mathcal{F}_t^{(i)} = \{P_{Y_t|X_t,S_t,Y^{t-1},h_i} : \sum_{\substack{y_t \in \overline{\mathcal{Y}}_t(x_t,s_t) \\ \forall x_t \in \mathcal{X}, s_t \in \mathcal{S}}} P(y_t|x_t, s_t, y^{t-1}, h_i) = 1, i \in \{0, 1\}\}, \tag{3.40}$$

where $f^{(i)}$ denote the specific energy management strategy under consumer's behavior hypothesis h_i. Also, let the control action be $a_t \in \mathcal{A}_t$, which is the conditional pmf $P_{Y_t|X_t,S_t}$ taken from the following set

$$\mathcal{A} = \{P_{Y|X,S} : \sum_{y \in \overline{\mathcal{Y}}(x,s)} P(y|x, s) = 1, \forall x \in \mathcal{X}, s \in \mathcal{S}\}. \tag{3.41}$$

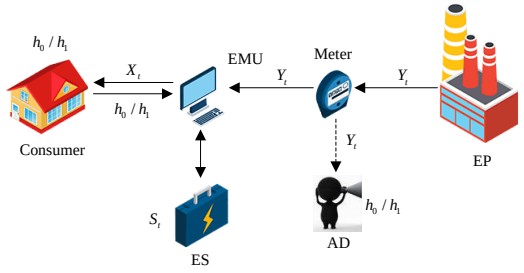


Figure 3.2: Smart metering system with rechargeable energy storage, authorized adversary, and privacy-aware energy management unit that knows the consumer's energy consumption behavior.

To make it more clear, we denote the action chosen under hypothesis h_i as $a^{(i)}$. Different from the previous definition of policy π in Section 3.2, at each time step t, the EMU will choose the action according to the policy $\pi_t^{(i)} \in \Pi_t^{(i)}$ under consumer's behavior hypothesis h_i. Define the set of binary consumer's behavior hypotheses as $\mathcal{H}$. Then $\Pi_t^{(i)}$ denotes the set of deterministic mappings from the historical observations (y^{t-1}, h_i) to a corresponding action $a_t^{(i)}$, i.e., $\Pi_t^{(i)} : \mathcal{Y}^{t-1} \times \mathcal{H} \to \mathcal{A}_t$ with $a_t^{(i)} = \pi_t^{(i)}(y^{t-1}, h_i)$. Thus, the policy under consumer's behavior hypothesis h_i over a T-time horizon is $\pi^{(i)} = \{\pi_t^{(i)}\}_{t=1}^T \in \Pi = \Pi_1^{(i)} \times \Pi_2^{(i)} \times .... \times \Pi_T^{(i)}$.

3.5.2 Design of Memory-Less Stationary Energy Management Strategy

With the above definition, we consider the problem of designing the memory-less stationary privacy-preserving strategies $\pi_t^{(i)}$ depending on different h_i, with the following objective function

$$\begin{aligned} L_T(\pi^{(0)}, \pi^{(1)}) &= \frac{1}{T} D(P_{Y^T|h_0}^{\pi^{(0)}} \| P_{Y^T|h_1}^{\pi^{(1)}}) = \frac{1}{T} \sum_{t=1}^{T} D(P_{Y_t|h_0,Y^{t-1}}^{\pi_t^{(0)}} \| P_{Y_t|h_1,Y^{t-1}}^{\pi_t^{(1)}}) \\ &= \frac{1}{T} \sum_{t=1}^{T} \sum_{y^t} P^{\pi^{(0)}}(y^t|h_0) \times \log \frac{P^{\pi_t^{(0)}}(y_t|y^{t-1}, h_0)}{P^{\pi_t^{(1)}}(y_t|y^{t-1}, h_1)}. \end{aligned} \tag{3.42}$$

Before designing the stationary energy management strategy for our proposed model, we first propose a structural simplification on the states and actions by introducing two new auxiliary random variables: $W_t \in \mathcal{W} = \{s_t - x_t : s_t \in \mathcal{S}, x_t \in \mathcal{X}\}$. Under policy $\pi^{(i)}$, define the posterior distributions of (X_t, S_t), W_t and S_t conditioned on the realization y^{t-1} as: $\theta_t^{(i)} = P_{X_t,S_t|y^{t-1},h_i}$, $\xi_t^{(i)} = P_{W_t|y^{t-1},h_i}$ and $\gamma_t^{(i)} = P_{S_t|y^{t-1},h_i}$. In particular, there is

$$\theta_t^{(i)}(x_t, s_t) = P_X^i(x_t)\gamma_t^i(s_t), \tag{3.43}$$

$$\begin{aligned}\gamma_t^{(i)}(s_t) &= P^{\pi^{(i)}}(S_t = s_t|Y^{t-1} = y^{t-1}, h_i),\\ \xi_t^{(i)}(w_t) &= P^{\pi^{(i)}}(W_t = w_t|Y^{t-1} = y^{t-1}, h_i).\end{aligned} \tag{3.44}$$

Since the following derivation works for both hypotheses h_0 and h_1, we only discuss the case for h_0 and denote $\theta_t^{(0)}, \gamma_t^{(0)}, \xi_t^{(0)}, a_t^{(0)}$ and $\pi_t^{(0)}$ by $\theta_t, \gamma_t, \xi_t, a_t$ and π_t. Le $\mathcal{D}(w_t) = |(x_t, s_t) \in \mathcal{X}_t \times \mathcal{S}_t : \ s_t - x_t = w_t|$, there is $\xi_t(w_t) = \sum_{(x_t,s_t)\in\mathcal{D}(w_t)} \theta_t(x_t, s_t)$. At time t, define a new action $b_t \in \mathcal{B}_t$ which is the condition pmf $P_{Y_t|W_t}$ taken from the set $\mathcal{B}_t = \{P_{Y|W} : \sum_{y\in\overline{\mathcal{Y}}(w)} P(y|w) = 1, \ \forall w \in \mathcal{W}\}$, where $\overline{\mathcal{Y}}(w)$ is defined by replacing $s_t - x_t$ with w in (3.2). Thus, action b_t can be expressed in terms of original belief state θ_t and action a_t,

$$\begin{aligned}b_t(y_t|w_t) \ &= \frac{P^{\pi}(Y_t = y_t, W_t = w_t|Y^{t-1} = y^{t-1}, h_0)}{P^{\pi}(W_t = w_t|Y^{t-1} = y^{t-1}, h_0)}\\ &= \frac{\sum_{(x_t,s_t)\in\mathcal{D}(w_t)} a_t(y_t|x_t, s_t)\theta_t(x_t, s_t)}{\xi_t(w_t)}.\end{aligned} \tag{3.45}$$

Similarly, we can define policy $\hat{\pi}_t$ as the deterministic mappings from the historical observations (y^{t-1}, h_0) to a corresponding action b_t, i.e., $b_t = \hat{\pi}_t(y^{t-1}, h_0)$. We further define the following distributions

$$\begin{aligned}\gamma'_t(s_t) &= P^{\hat{\pi}}(S_t = s_t|Y^{t-1} = y^{t-1}, h_0)\\ \xi'_t(w_t) &= P^{\hat{\pi}}(W_t = w_t|Y^{t-1} = y^{t-1}, h_0).\end{aligned} \tag{3.46}$$

Thus at time step t, for any realization of y_t and b_t, the evolution of ξ_t can be expressed in terms of b_t as follows

$$\begin{aligned}&\xi'_{t+1}(w_{t+1}) = \varphi'(\xi'_t, y_t, b_t) =\\ &\frac{\sum_{(x_{t+1},s_{t+1})\in\mathcal{D}(w_{t+1})} \sum_{w_t} \xi'_t(w_t) b_t(y_t|w_t) P_X^0(x_{t+1}) \mathcal{I}_{s_{t+1}}\{y_t + w_t\}}{\sum_{w_t} b_t(y_t|w_t)\xi'_t(w_t)}\end{aligned} \tag{3.47}$$

Lemma 2. *Given historical observations (y^{t-1}, h_0), the posterior distributions of W_t and S_t induced by policy π and $\hat{\pi}$ are the same, i.e.,*

$$\xi_t(w_t) = \xi'_t(w_t), \quad \gamma_t(s_t) = \gamma'_t(s_t), \ \ \forall s_t \in \mathcal{S}, \ w_t \in \mathcal{W}. \tag{3.48}$$

Proof: The proof is provided in the Appendix A.3.

With the above derivation, we have the following proposition.

Proposition 4. *Under the above assumption and transition, there is no loss of optimality in focusing on action $b_t \in \mathcal{B}_t$ and new belief state ξ_t instead of (a_t, θ_t). The per-step cost defined in (3.42) can be equivalently described by a function of state-action pair (b_t, ξ_t').*

Proof: To prove this proposition, we need to show that the conditional probability $P(y_t|y^{t-1}, a^{t-1}, h_0)$ remains identical under transition from $(a_t, \theta_t) \to (b_t, \xi_t)$. Note that the same arguments can be applied to h_1.

$$\begin{aligned} P^{\pi}(y_t|y^{t-1}, h_0) &= \sum_{w_t} P^{\pi}(Y_t = y_t, W_t = w_t|y^{t-1}, h_0) \\ &= \sum_{w_t} \sum_{(x_t,s_t)\in\mathcal{D}(w)} a_t(y_t|x_t, s_t)\theta_t(x_t, s_t) \\ &= \sum_{w_t} b_t(y_t|w_t)\xi_t(w_t) \\ &= \sum_{w_t} b_t(y_t|w_t)\xi_t'(w_t) \\ &= \sum_{w_t} P^{\hat{\pi}}(Y_t = y_t, W_t = w_t|y^{t-1}, h_0) \\ &= P^{\hat{\pi}}(y_t|y^{t-1}, h_0) \end{aligned} \tag{3.49}$$

□

Theorem 2. *Given $y \in \mathcal{Y}$, $w \in \mathcal{W}$, and any possible distributions γ_t', ξ_t', a time-invariant policy $\hat{f}$, with $\hat{f}(\xi_t) = \hat{b}_t$, leads to steady states, i.e., $\xi_t' = \xi_1'$ and $\gamma_t' = \gamma_1'$, if and only if $\hat{b}_t$ satisfies the following structure*

$$\hat{b}_t(y|w) = \begin{cases} Q_Y(y)\frac{\gamma_t'(y+w)}{\xi_t'(w)}, \ y \in \overline{\mathcal{Y}_t}(w), \\ 0, \ otherwise, \end{cases} \tag{3.50}$$

where $Q_Y(y)$ is an arbitrary probability distribution over all feasible $y \in [0, \min\{y_{max}, s_{max} + x_{max}\}]$, [4] and the same $Q_Y(y)$ is applied for the design of $\hat{b}_t$ at different time steps.

Proof: The proof is provided in Appendix A.4.

Moreover, an important conclusion drawn from Theorem 2 is summarized in the following corollary.

Corollary 1. *The distribution $Q_Y(y)$ that is used for designing the structured action $\hat{b}_t$ in Theorem 2 should satisfy the following equation:*

$$Q_Y \overset{(\Delta)}{=} P_{Y_t|Y^{t-1}=y^{t-1}}, \ \forall y^{t-1}. \tag{3.51}$$

Thus, the marginal distributions of Y_t at each time step are identical, i.e.,

$$Q_Y \overset{(\Delta)}{=} P_{Y_t}, \ \forall t. \tag{3.52}$$

[4] The energy supply y should lie in this feasible set due to the constraint of ES capacity.

3.5.3 Privacy-Preserving under Steady-State Strategy

For $i \in \{0,1\}$, let $\hat{f}^{(i)}$ be the time-invariant policy under hypothesis h_i which leads to the steady state. Also define $\hat{b}_t^{(i)}$ as the action decided by $\hat{f}^{(i)}$ at time t under hypothesis h_i, which satisfies the structure in Theorem 2. Further denote $Q_Y^0(y)$ and $Q_Y^1(y)$ as the distributions of Y used for constructing action $\hat{b}_t^{(0)}$ and $\hat{b}_t^{(1)}$, respectively. By combining the results in Theorem 2 and Corollary 1, we have the following corollary.

Corollary 2. *The objective function in (3.42) can be simplified to the single-letter expression by the following*

$$\begin{aligned} L_T(\hat{f}^{(0)},\hat{f}^{(1)}) =& \frac{1}{T} D(P_{Y^T|h_0}^{\hat{f}^{(0)}} \| P_{Y^T|h_1}^{\hat{f}^{(1)}}) \\ =& \frac{1}{T}\Big(D(P_{Y^{T-1}|h_0}^{\hat{f}^{(0)}} \| P_{Y^{T-1}|h_1}^{\hat{f}^{(1)}}) + \sum_{y^{T-1}} P^{\hat{f}^{(0)}}(y^{T-1}|h_0) \\ & \sum_{y_T} P^{\hat{f}^{(0)}}(y_T|y^{T-1},h_0) \log \frac{P^{\hat{f}^{(0)}}(y_T|y^{T-1},h_0)}{P^{\hat{f}^{(1)}}(y_T|y^{T-1},h_1)} \Big) \\ \overset{(a)}{=}& \frac{1}{T}\Big(D(P_{Y^{T-1}|h_0}^{\hat{f}^{(0)}} \| P_{Y^{T-1}|h_1}^{\hat{f}^{(1)}}) + \sum_{y} Q_Y^0(y) \log \frac{Q_Y^0(y)}{Q_Y^1(y)} \Big) \\ \overset{(b)}{=}& \frac{1}{T}\Big(T \times \sum_{y} Q_Y^0(y) \log \frac{Q_Y^0(y)}{Q_Y^1(y)} \Big) \\ =& D(Q_Y^0(y) \| Q_Y^1(y)), \end{aligned} \tag{3.53}$$

where (a) holds due to the fact $Q_Y^i(y) \overset{(\Delta)}{=} P_{Y_t|Y^{t-1}=y^{t-1},h_i}^{\hat{f}^{(i)}}$, $\forall y^{t-1}$, and (b) follows from iteratively applying the chain rule of KL-divergence.

On observing the single-letter expression (3.53) given in Corollary 3, the following corollary is proposed as a consequence.

Corollary 3. *The time-invariant policies $\hat{f}^{(i)}$ will achieve the zero-lower bound of Kullback-Leibler divergence, i.e., perfect privacy, if and only if the distributions of Y used for constructing $\hat{b}_t^{(i)}$ (decided by $\hat{f}_t^{(i)}$) are equal, i.e., $Q_Y^0(y) = Q_Y^1(y)$, $\forall y$.*

Remark 6. *Due to the physical constraints of the system we may not find the feasible Q_Y^0 and Q_Y^1 that satisfy the condition in the above corollary.*

Next, we present a case in which the perfect privacy cannot be achieved. It follows from (3.1) that the per-step expected energy amount constraint should hold as:

$$\mathbb{E}_{P_X}[X_t] + \mathbb{E}_{\gamma_t(s)}[S_t] - \mathbb{E}_{\gamma_{t-1}(s)}[S_{t-1}] = \mathbb{E}_{Q_Y}[Y_t], \tag{3.54}$$

which means, at each time step, the average expected amount of energy requested from the grid should be equal to sum of the average expected amount of energy consumed by the

consumer and the average expected amount of energy stored into the ES. Since $\gamma_t = \gamma_{t-1}$ under a stationary strategy, equation (3.54) then further reduces to $\mathbb{E}_{P_X}[X_t] = \mathbb{E}_{Q_Y}[Y_t]$. In this case, we cannot have arbitrary Q_Y^i for constructing $\hat{b}_t^{(i)}$. Instead, Q_Y^i needs to satisfy the constraint $\mathbb{E}_{P_X}[X_t] = \mathbb{E}_{Q_Y}[Y_t]$ for both hypotheses. For the perfect privacy case, in order to satisfy the condition $Q_Y^0(y) = Q_Y^1(y)$, we should at least have $\mathbb{E}_{Q_Y^0}[Y] = \mathbb{E}_{Q_Y^1}[Y]$, which will conflict with the practical case of $\mathbb{E}_{P_X^0}[X] \neq \mathbb{E}_{P_X^1}[x]$. However, only considering the constraint on expected energy amount is not enough, since several different distributions of Y might lead to the same expectation and there might be some other constraints which requires $Q_Y^0(y) \neq Q_Y^1(y)$. Thus, the condition $Q_Y^0(y) = Q_Y^1(y)$ may still not be satisfied even if we have $\mathbb{E}_{Q_Y^0}[Y] = \mathbb{E}_{Q_Y^1}[Y]$. The above analysis can be summarized in the following proposition.

Proposition 5. *If the consumer's expected demand of energy under h_0 and h_1 are different, i.e., $\mathbb{E}_{P_X^0}[X] \neq \mathbb{E}_{P_X^1}[X]$, the perfect privacy cannot be achieved, since we cannot find Q_Y^0 and Q_Y^1 satisfying the conditions in Corollary 3.*

3.6 Numerical Experiments

In this section, the performance of the proposed energy management strategies will be numerically evaluated.

3.6.1 Experiment Settings

For simplicity we do not include units in the following. We consider a finite horizon with length $T = 10$. The energy demand, supply and price alphabets are set as $\mathcal{X} = \{0, 1, 2, 3, 4, 5\}$, $\mathcal{Y} = \{0, 1, 3, 4, 5\}$, $\mathcal{P} = \{5, 10\}$, and the ES capacity can be $s_{max} = 2$ or $s_{max} = 5$. The transition probabilities of x_t under both hypotheses, the transition probability of p_t and the initial belief state are set as following:

$$\begin{aligned}
P(h_0) = P(h_1) = 0.5,\\
P_{X_{t+1}|h_0,X_t}(x_{t+1}|h_0, x_t) = \frac{1}{6}, \forall x_t, x_{t+1} \in \{0, 1, 2, 3, 4, 5\},\\
P_{X_{t+1}|h_1,X_t}(x_{t+1}|h_1, x_t) = 0.4, \forall x_{t+1} = x_t\\
P_{X_{t+1}|h_1,X_t}(x_{t+1}|h_1, x_t) = 0.12, \forall x_{t+1} \neq x_t\\
P_{P_{t+1}|P_t}(p_{t+1}|p_t) = 0.5, \forall p_{t+1} \in \{5, 10\}, p_t \in \{5, 10\},\\
\theta(x_1, s_1, p_1|h_0) = \theta(x_1, s_1, p_1|h_1) = \frac{1}{12 \times (s_{max} + 1)}\\
\forall x_1 \in \{0, 1, 2, 3, 4, 5\}, s_1 \in \{0, 1, ...s_{max}\}, p_1 \in \{5, 10\}.
\end{aligned} \tag{3.55}$$

The continuous belief state space is discretized into 36 discrete different distributions (for ES capacity $s_{max} = 2$) and 100 belief sates (for ES capacity $s_{max} = 5$), i.e., 36^2 or 100^2 belief state vectors in total. Also, we have a finite action set $\mathcal{A}'$ including 20

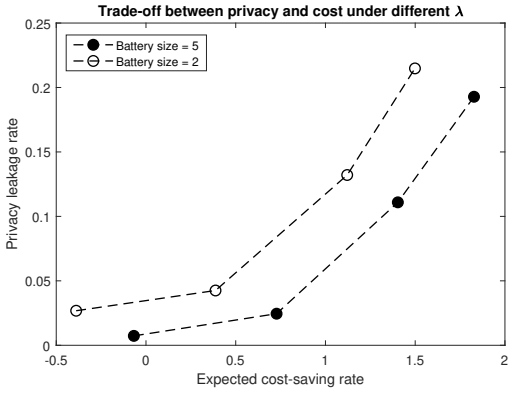


Figure 3.3: Trade-off between privacy leakage and cost-saving for different ES sizes. From left to right the data points represent: $\lambda = 1, 0.8, 0.2, 0$.

Table 3.1: Value compare between differen λ for battery size 5 in Fig. 3

λ	Privacy leakage rate	Expected cost-saving rate
0	0.1928	1.8276
0.2	0.1109	1.4034
0.8	0.0245	0.7277
1	0.0073	-0.0675

different actions that randomly decides on the energy supply Y_t according to the conidtion pmf $P_{Y_t|X_t,S_t,P_t}$.

3.6.2 Experiments for Finite Horizon Dynamic Programming

For different battery capacities, we investigate the trade-off between privacy enhancement and cost-saving by setting $\lambda = 1, 0.8, 0.2, 0$. The variation of privacy leakage rate against expected cost-saving rate with respect to λ is shown in Fig. 3.3. As λ increases, both the corresponding privacy leakage rate and expected cost-saving rate increase, which confirms the intuition that more cost-saving can be achieved at a cost of larger privacy leakage. We can also see from the figure that the performance will improve when the ES capacity gets larger.

3.6.3 Experiments for Solutions over Infinite Time Horizon

In this section, we compare the optimal and sub-optimal solutions to our belief-state MDP problem over the infinite time horizon, where the optimal solution is derived by the RVI and the sub-optimal one is derived by the LARQL. For this part, we have the ES capacity

Table 3.2: Value compare between differen λ for battery size 2 in Fig. 3

λ	Privacy leakage rate	Expected cost-saving rate
0	0.2148	1.4987
0.2	0.1321	1.1211
0.8	0.0425	0.3875
1	0.0268	-0.389

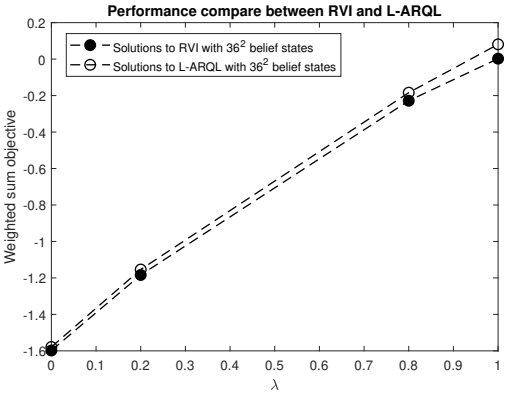


Figure 3.4: Performance compare between LARQL and RVI with 36^2 belief states.

as $s_{max} = 5$ and the finite action set $\mathcal{A}'$ includes 20 different actions. The continuous belief state space is first discritized into 36^2 belief states. We compare the optimal and sub-optimal overall objective function (weighted sum between privacy leakage and cost savings) derived by RVI and LARQL approach. As we can see from Fig. 3.4, the performance of our proposed LARQL algorithm is close to be optimal. We can also see in the figure that the gap between LARQL and the optimal solution when $\lambda = 1$ is larger than the gap for $\lambda = 0$, which means that our linear function can approximate the Q-function when only induced by expected cost-saving ($\lambda = 0$) better than when only induced by the KL-divergence term ($\lambda = 1$).

Next, as shown in Fig. 3.5, with the increased number of belief states to be 4368, i.e., 4368^2 possible belief state vectors in total, the sub-optimal results using LARQL are shown in the figure, while the RVI method is too time-consuming and it is impractical to get an exact solution.

Based on the result, we notice that the performance of LARQL improves a lot with larger amount of belief states. The gap when $\lambda = 0$ is larger is again because our linear approximator can approximate Q-function induced only by expected cost-savings better than the Q-function only induced by KL-divergence. Besides, the performance of L-ARQL with 4368^2 is better than the optimal solutions with 36^2 belief states, this is due to the larger

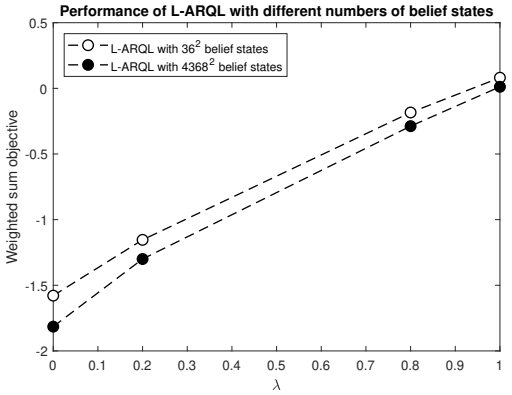


Figure 3.5: Performance compare between LARQL with 36^2 and 4368^2 belief states.

amount of belief states offers the system more freedom.

3.6.4 Experiments on real data

Finally, to better demonstrate the working mechanism of our proposed algorithm, we present our simulation results using real data from the reference data set REDD [81]. We consider a kitchen with a dishwasher which has two different operation modes: types A (hypothesis h_0) and B (hypothesis h_1). Over a time horizon with 1000 sampling instances, the load signatures of the two different operation types are illustrated in the upper two figures of Fig. 3.6. Both operation modes involve three different states $x \in [10, 200, 1100]$. The duration of staying in a state and the transition probabilities between different states depend on the operation mode. We restrict the energy supply to take values within the set $\mathcal{Y} = [0, 10, 200, 310, 400, 500, 1100]$. The battery level is quantized into $[0, 200, 800]$ and the transition probabilities for the two operation modes are known. For a time-horizon of 1000 samples, we implement and simulate our privacy-preserving energy management strategy derived from our finite horizon belief-state MDP design. Two realizations of the requested energy profiles for both operation modes, i.e., the random output of our energy management strategy, are presented in the lower two figures of Fig. 3.6. From the visual comparison of the profiles, we can see that it becomes very difficult for the AD to identify the hypothesis from the energy supply data.

3.7 Summary

In this chapter, we have shown that an energy storage can be used for both privacy enhancing and cost saving. Using the belief-state MDP framework, an energy management strategy that optimally trade off KL-divergence and expected cost-saving rates can be derived using the Bellman dynamic programming. The complexity of the optimal design

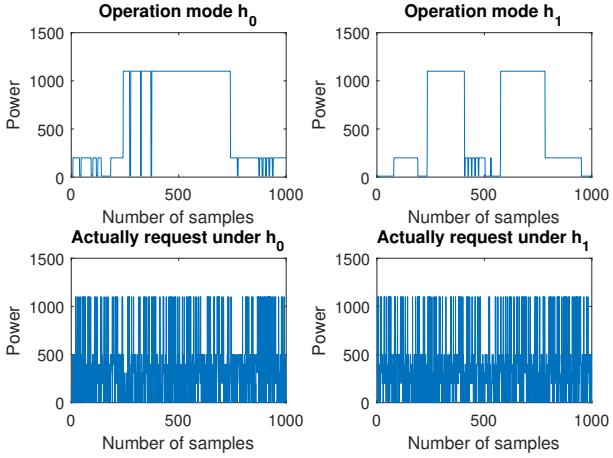


Figure 3.6: Upper figures show dishwasher signatures of two different operation modes (hypothesis). Lower figures show realizations of the requested energy profiles for both hypotheses when privacy-preserving policy is used. Operation modes will hard to differentiate due to randomness in profiles.

problems grows quickly, which calls for computationally efficient solutions. Our proposed sub-optimal linear function approximated relative Q-learning approach is computationally efficient and also works for an infinite time horizon. With the identified feature vector, the linear function approximated Q-function can be efficiently learned and therefore leads to a practical online energy management design approach. Another approach to reduce the strategy design complexity is to assume an i.i.d. energy demand, which allows further analysis, in particular the derivation of a steady state strategy. Moreover, we provide sufficient conditions to achieve perfect privacy. Our numerical experiments show that the framework leads to energy management strategies that optimally trade off privacy enhancing and cost saving. They also show that our proposed LARQL method is close to optimal performance but is significantly computationally more efficient.

Chapter 4

Privacy-Cost Trade-off in the Presence of a Renewable Energy Source

In this chapter, we design privacy-preserving and cost-efficient energy management strategies for smart grid users that are equipped with RES. The adversary is assumed to employ a FHMM based inference for load disaggregation, and the corresponding joint log-likelihood of the model is utilized as privacy measure. A dynamic pricing model is studied, where the price of unit amount of energy is determined by the users' aggregated power request, which suits a commodity-limited market. The users' energy management strategy is designed under a non-cooperative game framework by optimizing a weighted sum of both privacy measure and the user's energy cost savings. The users' non-cooperative game is shown to admit a unique pure strategy NE. As an extension, a computational-efficient distributed Nash equilibrium energy management strategy seeking method is proposed, which also avoids the privacy leakage due to the sharing of payoff functions between users. The content of this chapter has been taken from **Paper B**, while some parts have been verbatim copied.

This chapter is organized as follows: the FHMM based NILM technique is first presented in Section 4.1. Based on this, Section 4.2 proposes the privacy metric against the FHMM based adversarial load disaggregation. The system model and the non-cooperative game formulation of the privacy-cost trade-off problem are provided in Section 4.3. The corresponding distributed NE energy management strategy seeking algorithm is proposed in Section 4.4. Finally, numerical results are given in Section 4.5 and we conclude this chapter in Section 4.6.

4.1 HMM based NILM

In this section, we propose a privacy-preserving problem against an FHMM based NILM adversary. More specifically, the NILM problem is formulated as an adversarial inference by considering an FHMM framework. Further, against such an attacking behavior, we

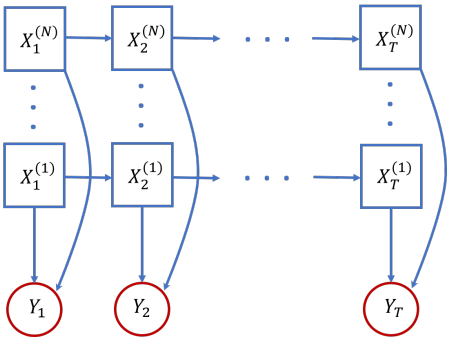


Figure 4.1: Factorial HMM for a household with N appliances.

propose a metric that can measure the user's privacy-preserving performance.

4.1.1 Basic HMM for Individual Appliance

At each time step, an appliance is assumed to operate at one of K states. However, the operating state $i \in \{1, 2, \ldots, K\}$ cannot be directly observed. Therefore, the operating state is described as the hidden state of an HMM. Each operating state can "emit" an observation of random power consumption. In more details, when an appliance operates at state i, the corresponding random power consumption is assumed to satisfy a Gaussian distribution with mean μ_i and standard deviation σ_i. The HMM for an individual appliance can be defined as follows: At time step t, the discrete hidden state X_t of the HMM represents the current operating state of the appliance, i.e., $X_t = i \in \{1, 2, ..., K\}$, and the observation $Y_t \in \mathbb{R}$ of the HMM represents the current power consumption of the appliance. The HMM is then fully characterized by the parameters $\lambda = \{\pi, A, \mathcal{B}\}$ with:

- Prior distribution of the initial state $\pi \in \mathbb{R}^{K\times 1}$, where $\pi_i = P_{X_1}(x_1 = i)$ denotes the probability that the appliance is initially operating at state i;
- Transition matrix $A \in \mathbb{R}^{K\times K}$, where the element A_{ij} denotes the stationary transition probability for the appliance switching from operating state i at time step t to operating state j at time step $t + 1$, i.e.,

$$A_{ij} = P_{X_{t+1}|X_t}(x_{t+1} = j|x_t = i), \forall t \geq 1; \tag{4.1}$$

- The set of Gaussian emission distributions $\mathcal{B} = \{\mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^K$, where $Y_t|X_t = i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ represents the power consumption distribution for the operating state i for all $t \geq 1$.

Fitting the HMM model for an individual appliance requires learning the above parameters, which has been widely studied and developed in NILM studies such as in [36–38].

4.1.2 FHMM Modeling for Multiple Appliances

In our problem, there are N appliances in each household. We assume that all appliances are operating independently. The FHMM for a household is shown in Fig. 4.1, where the hidden state $X_t^{(n)}$ represents the operating state of appliance n at time step t, and the emitted observation Y_t represents the aggregated power consumption of all appliances in the household at time step t. We define $\lambda^{(1:N)} = \{\pi^{(1:N)}, A^{(1:N)}, \mathcal{B}^{(1:N)}\}$ as the parameters that fully characterize the FHMM, where $\pi^{(1:N)} = \{\pi^{(n)}\}_{n=1}^{N}$, $A^{(1:N)} = \{A^{(n)}\}_{n=1}^{N}$, $\mathcal{B}^{(1:N)} = \{\mathcal{B}^{(n)}\}_{n=1}^{N}$ denote the prior distributions of the initial states, transition matrices, and sets of Gaussian emission distributions for all N appliances, respectively. Now the joint log-likelihood of the hidden state sequences $\boldsymbol{x}^{(1:N)} = \{\{x_t^{(n)}\}_{t=1}^{T}\}_{n=1}^{N}$ and observation sequence $\boldsymbol{y} = \{y_t\}_{t=1}^{T}$ can be rewritten as

$$\mathcal{L}\left(\boldsymbol{y}, \boldsymbol{x}^{(1:N)} \middle| \lambda^{(1:N)}\right) = \sum_{n=1}^{N} \log P\left(x_1^{(n)} \middle| \pi^{(n)}\right) + \sum_{t=2}^{T}\sum_{n=1}^{N} \log P\left(x_t^{(n)} \middle| x_{t-1}^{(n)}, A^{(n)}\right) + \sum_{t=1}^{T} \log P\left(y_t \middle| x_t^{(1:N)}, \mathcal{B}^{(1:N)}\right). \tag{4.2}$$

4.2 Privacy-Preserving against FHMM based NILM

NILM can be modeled as an adversarial inference by combining all individual HMMs [3]. Due to the privacy concern, each user wishes to hide the true operating state sequences $\boldsymbol{x}^{*(1:N)}$. Here, an EMU is employed for each household to modify the power consumption to power request. Let z_t denote the power request decided by EMU at time step t, which is assumed to be fixed during a short period until the next decision is made. Thus, the observation of the adversarial NILM is the power request sequence $\boldsymbol{z} = \{z_t\}_{t=1}^{T}$ instead of the power consumption sequence $\boldsymbol{y}$. We assume that the naive adversary knows the FHMM of each household but believes the observation is $\boldsymbol{y}$. Given the FHMM parameters $\theta^{(1:N)}$ and the true observation sequence $\boldsymbol{z}$, the adversarial NILM is MAP inference as

$$\hat{\boldsymbol{x}}^{(1:N)} = \arg\max_{\boldsymbol{x}^{(1:N)}} \mathcal{L}\left(\boldsymbol{z}, \boldsymbol{x}^{(1:N)} \middle| \lambda^{(1:N)}\right), \tag{4.3}$$

where $\hat{\boldsymbol{x}}^{(1:N)}$ are the most likely hidden operating state sequences. Taking into account the FHMM-based adversarial NILM in (4.3) and the privacy-preserving objective, the EMU can optimally design the power request sequence $\boldsymbol{z}^*$ to minimize the following joint log-likelihood as

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \mathcal{L}\left(\boldsymbol{z}, \boldsymbol{x}^{*(1:N)} \middle| \lambda^{(1:N)}\right). \tag{4.4}$$

Operationally, given the optimal power request sequence $\boldsymbol{z}^*$, the true operating state sequences $\boldsymbol{x}^{*(1:N)}$ has the *minimum* joint log-likelihood, i.e., the FHMM-based adversarial NILM will least-likely make a correct inference on the true operating state sequences.

Proposition 6. *The privacy-preserving problem* (4.4) *is equivalent to the following optimization:*

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \sum_{t=1}^{T} \left[-\frac{1}{2\sigma_t^2}(z_t - \mu_t)^2 \right], \tag{4.5}$$

where $\mu_t = \sum_{n=1}^{N} \mu_{x_t^{*(n)}}$ *and* $\sigma_t^2 = \sum_{n=1}^{N} \sigma^2_{x_t^{*(n)}}$.

Proof: Since the adversary believes his observation is $\boldsymbol{y}$*, the joint log-likelihood in* (4.4) *can be rewritten similarly as* (4.2) *as*

$$\begin{aligned}
&\mathcal{L}\left(\boldsymbol{z}, \boldsymbol{x}^{*(1:N)} \middle| \theta^{(1:N)}\right) \\
&= \sum_{n=1}^{N} \log P\left(x_1^{*(n)} \middle| \pi^{(n)}\right) + \sum_{t=2}^{T}\sum_{n=1}^{N} \log P\left(x_t^{*(n)} \middle| x_{t-1}^{*(n)}, A^{(n)}\right) \\
&\quad + \sum_{t=1}^{T} \log P\left(Y_t = z_t \middle| x_t^{*(1:N)}, \mathcal{B}^{(1:N)}\right).
\end{aligned} \tag{4.6}$$

Since the first two terms in (4.6) *do not depend on* $\boldsymbol{z}$*, the optimization problem* (4.4) *is equivalent to the following problem*

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \sum_{t=1}^{T} \log P\left(Y_t = z_t \middle| x_t^{*(1:N)}, \mathcal{B}^{(1:N)}\right), \tag{4.7}$$

where the RHS is fully parameterized by the emission distributions in $\mathcal{B}^{(1:N)}$*. From the perspective of the adversary, the power request* Z_t *nominally has the same emission distribution as the aggregated power consumption* Y_t*, which is a Gaussian distribution as* $Y_t | x_t^{*(1:N)} \sim \mathcal{N}\left(\sum_{n=1}^{N} \mu_{x_t^{*(n)}}, \sum_{n=1}^{N} \sigma^2_{x_t^{*(n)}}\right) = \mathcal{N}(\mu_t, \sigma_t^2)$*. Thus, we have*

$$\log P\Big(Y_t = z_t | x_t^{*(1:N)}, \mathcal{B}^{(1:N)}\Big) = \log \frac{1}{\sqrt{2\pi\sigma_t^2}} - \frac{1}{2\sigma_t^2}(z_t - \mu_t)^2. \tag{4.8}$$

Since only the term $-\frac{1}{2\sigma_t^2}(z_t - \mu_t)^2$ *depends on the power request* z_t*, the optimization problem (4.5) is equivalent to the optimization problem (4.4).* □

Definition 1. *Regarding the FHMM-based adversarial NILM, we define the privacy-preserving metric as*

$$\mathcal{D}(\boldsymbol{z}, \{\mu_t\}_{t=1}^T, \{\sigma_t^2\}_{t=1}^T) = \sum_{t=1}^{T} \left[-\frac{1}{2\sigma_t^2}(z_t - \mu_t)^2 \right]. \tag{4.9}$$

Note that $\boldsymbol{z}^*$ is the best privacy-preserving power request sequence against the FHMM-based adversarial NILM and achieves the minimum privacy-preserving metric $\mathcal{D}(\boldsymbol{z}^*, \{\mu_t\}_{t=1}^T, \{\sigma_t^2\}_{t=1}^T)$. Operationally, a power request sequence $\boldsymbol{z}$ with a *smaller* value of $\mathcal{D}(\boldsymbol{z}, \{\mu_t\}_{t=1}^T, \{\sigma_t^2\}_{t=1}^T)$ has a *better* privacy-preserving performance.

4.3 System Model and Non-Cooperative Game Formulation

Consider a smart metering system as shown in Fig. 4.2, where M users (households) are served by the same EP. At time step t, for user $i \in \mathcal{M} = \{1, 2, ..., M\}$, denote its power consumption by $y_{i,t}$, where $y_{i,t}$ takes values from the set of samples of a certain Gaussian distribution within the range $\mathcal{Y} = [0, y_{max}]$, and denote its power request from the EP as $z_{i,t} \in \mathcal{Z} = [0, z_{max}]$. Assume each user has an RES with time-variant finite power capacity $R_{i,t}$, we denote the power request of user i from the RES as $r_{i,t} \in \mathcal{R}_t = [0, R_{i,t}]$. Note that $y_{i,t}$, $z_{i,t}$, and $r_{i,t}$ are assumed to be fixed during the short period between time steps t and $t+1$. Moreover, the power consumption $y_{i,t}$ should always be satisfied by supplies from either EP or RES without any wasted energy, i.e., $y_{i,t} = r_{i,t} + z_{i,t}$. Since the power supply from the RES is limited as $r_{i,t} \in \mathcal{R}_t = [0, R_{i,t}]$, the power request from the EP $z_{i,t}$ should be chosen from the following feasible set

$$\overline{\mathcal{Z}}(y_{i,t}, R_{i,t}) = \{z_{i,t} \in \mathcal{Z} : \max\{0, y_{i,t} - R_{i,t}\} \leq z_{i,t} \leq y_{i,t}\}, \tag{4.10}$$

where the lower bound $y_{i,t} - R_{i,t}$ ensures that the power request $z_{i,t}$ provides at least the remaining power when the RES power supply cannot solely satisfy the user's power consumption; the other lower bound 0 means no energy can be sold back to the grid; and the upper bound is due to the constraint that no energy is wasted. Assuming the adversary is applying the above FHMM-based NILM to infer on the hidden operating states for all N appliances of user i over a relatively short time period T, i.e., $\left\{x_{i,1}^{*(n)}, x_{i,2}^{*(n)}, ..., x_{i,T}^{*(n)}\right\}$ for all $1 \leq n \leq N$. We further assume that the user can predict its own operating states over the time period T. Therefore, $\{\mu_{i,1}, \mu_{i,2}, ..., \mu_{i,T}\}$ and $\left\{\sigma_{i,1}^2, \sigma_{i,2}^2, ..., \sigma_{i,T}^2\right\}$ are known to user i. If we only consider the privacy of user i, the corresponding privacy-preserving problem is

$$\min_{\{\boldsymbol{z}_i : z_{i,t} \in \overline{\mathcal{Z}}(y_{i,t}, R_{i,t}), \forall t\}} \mathcal{D}(\boldsymbol{z}_i, \{\mu_{i,t}\}_{t=1}^T, \{\sigma_{i,t}^2\}_{t=1}^T). \tag{4.11}$$

We next introduce the pricing model. At time step t, the price is decided by the total power request of all users, and each user is billed based on the price and the amount of his own energy request. We assume that the users are truthful and have no incentive to deviate, given the possible penalties that will be incurred. Same as [82], we consider that the price is proportional to the total power request of all users as

$$\rho(\boldsymbol{z}_t, \rho_{base}, \alpha) = \rho_{base} + \alpha \sum_{i \in \mathcal{M}} z_{i,t}, \tag{4.12}$$

where ρ_{base} and α are parameters set by the EP, and $\boldsymbol{z}_t = \{z_{1,t}, z_{2,t}, \ldots, z_{M,t}\}$ are the power requests of all users at time step t. The EP adjusts ρ_{base} and α to control the price and further the amount of power consumption of the users. Before the beginning of the T-time horizon, the EP will decide on ρ_{base} and α, and the corresponding pricing function will be announced as a prior knowledge to all users. If we only consider the cost-saving objective, user i wishes to solve the following problem:

$$\max_{\{\boldsymbol{z}_i : z_{i,t} \in \overline{\mathcal{Z}}(y_{i,t}, R_{i,t}), \forall t\}} \sum_{t=1}^{T} \eta(y_{i,t} - z_{i,t}) \rho(\boldsymbol{z}_t, \rho_{base}, \alpha), \tag{4.13}$$

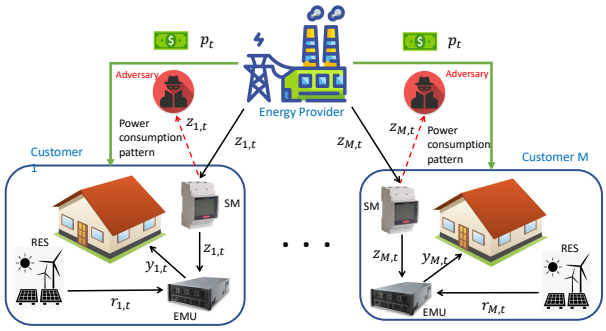


Figure 4.2: Smart grid system with an energy provider and M users. Each user is equipped with a RES and a privacy-cost-aware EMU. The EMU modifies power consumption profile to protect privacy against the adversary, who applies FHMM-based NILM, and to reduce the cost by taking into account the dynamic energy price.

where η is the factor that transfers the power to the energy.

In general, user i can have both the privacy-preserving and cost-saving objectives with different concern weights. Thus, the EMU of user i generates the power requests to maximize the weighted privacy-preserving and cost-efficient objective as

$$\sum_{t=1}^{T} \eta(y_{i,t} - z_{i,t})\rho(\boldsymbol{z}_t, \rho_{base}, \alpha) - \nu_i \mathcal{D}(\boldsymbol{z}_i, \{\mu_{i,t}\}_{t=1}^{T}, \{\sigma_{i,t}^2\}_{t=1}^{T}), \tag{4.14}$$

where $\nu_i \geq 0$ is the weight factor to tradeoff the privacy-preserving and cost-saving objectives. Solving such a problem via offline optimization methods requires full knowledge of the operating state sequences, the power consumption sequence, the time-variant power capacity of RES. Since we consider a relatively short time horizon, we assume that all those parameters can be predicated before the beginning. Furthermore, the price model depends on the power requests of all users. Therefore, all users interact with each other when they determine their own power request sequences. In the following, we will formulate the strategic interaction as a non-cooperative game.

Given the power requests of all other users $\boldsymbol{z}_{-i} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{i-1}, \boldsymbol{z}_{i+1}, \ldots, \boldsymbol{z}_M\}$, the payoff function of user i by sending the power requests $\boldsymbol{z}_i = \{z_{i,1}, z_{i,2}, \ldots, z_{i,T}\}$ is defined as

$$U_i(\boldsymbol{z}_i, \boldsymbol{z}_{-i}) = \sum_{t=1}^{T} \eta(y_{i,t} - z_{i,t})\rho(\boldsymbol{z}_t, \rho_{base}, \alpha) - \nu_i \mathcal{D}(\boldsymbol{z}_i, \{\mu_{i,t}\}_{t=1}^{T}, \{\sigma_{i,t}^2\}_{t=1}^{T}).$$

As the price is jointly determined by the power requests of all users $\boldsymbol{z}_t$, the payoff of each user is a function of power requests of other users. Thus, the users are engaged in a non-cooperative game, where each user wishes to maximize his own payoff rationally.

In this game, we are interested in characterizing the NE, in which each user plays the best-response strategy of all other users' strategies to maximize the payoff function.

Definition 2. *In the non-cooperative game, the power requests of all users* $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \ldots, \boldsymbol{z}_M^*)$ *form an NE if and only if*

$$U_i(\boldsymbol{z}_i^*, \boldsymbol{z}_{-i}^*) \geq U_i(\boldsymbol{z}_i, \boldsymbol{z}_{-i}^*), \forall i \in \mathcal{M}. \tag{4.15}$$

With the above definitions, the existence of the NE is guaranteed by the following theorem.

Theorem 3. *The non-cooperative game admits a unique pure-strategy NE if* $\alpha > \frac{\nu_i}{2\eta\sigma_{i,t}^2}$, $\forall 1 \leq i \leq M$, $1 \leq t \leq T$.

Proof: The proof is provided in Appendix A.7.

Remark 7. *The sufficient condition to guarantee the existence of a unique pure-strategy NE is equivalent to form a multi-player concave game. Therefore, the sufficient condition is more likely to be satisfied by increasing the cost-saving "weight":* $\eta\alpha$*, or by decreasing the privacy-preserving "weights":* $\frac{\nu_1}{2\sigma_{1,1}^2}$, $\frac{\nu_2}{2\sigma_{1,2}^2}$, $\ldots$, *or* $\frac{\nu_M}{2\sigma_{M,T}^2}$.

4.4 Distributed NE Energy Management Seeking Algorithm

Solving the unique NE of the non-cooperative game via the traditional centralized method is computationally complex and requires all the users to share their payoff functions. As shown in (4.15), the payoff function of each user contains the private information, such as $(\mu_{i,t}, \sigma_{i,t}^2)$ related with the hidden operating states and the power consumption $y_{i,t}$. Thus, each user's power consumption behavior will be revealed by sharing the payoff function with the other users. Due to the high complexity and privacy risk in the traditional centralized method, we develop a distributed algorithm that can converge to the unique NE based on the traditional relaxation algorithms [83]. Compared with the traditional centralized method, our proposed algorithm is implemented in a distributed manner such that each user only needs to know his own action and the best response given other users' actions without sharing the payoff function, which significantly reduces the computational complexity and avoids the privacy leakage.

With slight abuse of notation, we define the action set of an arbitrary game as $\mathcal{X}$, and let $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{X}$ two feasible action profiles.

Definition 3. *Let* U_i *be the payoff function of player* i*. Given two action profiles* $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)^{'}$ *and* $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^{'}$*, the Nikaido-Isoda function* $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is then defined as*

$$\psi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \mathcal{M}} \big[U_i(y_i, x_{-i}) - U_i(x_i, x_{-i})\big]. \tag{4.16}$$

Definition 4. *Given the action profile $\boldsymbol{x}$, the best response vector is defined as*

$$BR(\boldsymbol{x}) = (BR_1(\boldsymbol{x}), \ldots, BR_M(\boldsymbol{x}))' = \arg\max_{\boldsymbol{y}\in\mathcal{X}} \psi(\boldsymbol{x}, \boldsymbol{y}). \tag{4.17}$$

Now we consider the M-person concave game at time step t. Applying the above definitions to our problem, the corresponding Nikaido-Isoda function is

$$\begin{aligned}\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t) = \sum_{i=1}^{M} \Bigg[&(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha) z_{i,t}'^2 + (\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}) z'_{i,t} \\ &-(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha) z_{i,t}^2 - (\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}) z_{i,t} \Bigg].\end{aligned} \tag{4.18}$$

Given the above Nikaido-Isoda function, we provide the best response power request of a user at time step t in the following lemma.

Lemma 3. *At time step t, given a power request profile $\boldsymbol{z}_t$, the best response power request of user i is given by*

$$BR_i(\boldsymbol{z}_t) = \begin{cases} \frac{\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}}{2\eta\alpha - \frac{\nu_i}{\sigma_{i,t}^2}}, \text{ if } \frac{\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}}{2\eta\alpha - \frac{\nu_i}{\sigma_{i,t}^2}} \in \overline{\mathcal{Z}}(y_{i,t}, R_{i,t}) \\ y_{i,t}, \text{ if } \frac{\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}}{2\eta\alpha - \frac{\nu_i}{\sigma_{i,t}^2}} > y_{i,t} \\ \max\{0, y_{i,t} - R_{i,t}\}, \text{ otherwise} \end{cases}. \tag{4.19}$$

Proof. By taking the partial derivative of the Nikaido-Isoda function $\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t)$ with respect to $z'_{i,t}$, the best response of User i can be easily derived by solving the optimality condition $\frac{\partial\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t)}{\partial z'_{i,t}} = 0$ for $i \in \mathcal{M}$. □

Based on the derived best response, we propose the following modified relaxation algorithm that can be performed in a distributed manner. And we further show that the power request profile of all users will converge to the unique pure-strategy NE by implementing the proposed distributed relaxation algorithm. In the following theorem, we provide a sufficient condition for the convergence of Algorithm 4.

Theorem 4. *By implementing Algorithm 4, the power request profile of all users will converge to the unique pure-strategy NE of the non-cooperative game if $\alpha > \frac{\nu_i}{\eta\sigma_{i,t}^2}$ for all $1 \le i \le M$ and $1 \le t \le T$.*

Proof: The proof is provided in Appendix A.6.

4.5 Numerical Experiments

In this section, we present numerical results of our proposed privacy-preserving and cost-efficient energy management strategy. The experiments are carried out based on the low

Algorithm 4: Distributed Relaxation Algorithm for Solving the Nash Equilibrium

Initialize T-time horizon power request profile $\boldsymbol{z}_i^{[0]}$ for all $i \in \mathcal{M}$.
Set the initial iteration index $s = 0$, the threshold value ϵ, the initial step size ξ_0, and the diminishing step size $\xi_s = \frac{\xi_0}{\sqrt{s}}$ for all $s \geq 1$.
repeat
  for $t = 1, 2, \ldots, T$ **do**
    Each user broadcasts the latest updated power request to all other users, i.e., all users know $\boldsymbol{z}_t^{[s]}$;
    for $i \in \mathcal{M}$ **do**
      User i calculates his best response power request $BR_i(\boldsymbol{z}_t^{[s]})$ and updates his power request as

$$z_{i,t}^{[s+1]} \leftarrow (1 - \xi_s) z_{i,t}^{[s]} + \xi_s BR_i(\boldsymbol{z}_t^{[s]});$$

    end for
  end for
  $s \leftarrow s + 1$;
until $\left\| U_i(\boldsymbol{z}_i^{[s]}, \boldsymbol{z}_{-i}^{[s]}) - U_i(\boldsymbol{z}_i^{[s-1]}, \boldsymbol{z}_{-i}^{[s-1]}) \right\|^2 \leq \epsilon, \forall i \in \mathcal{M}$

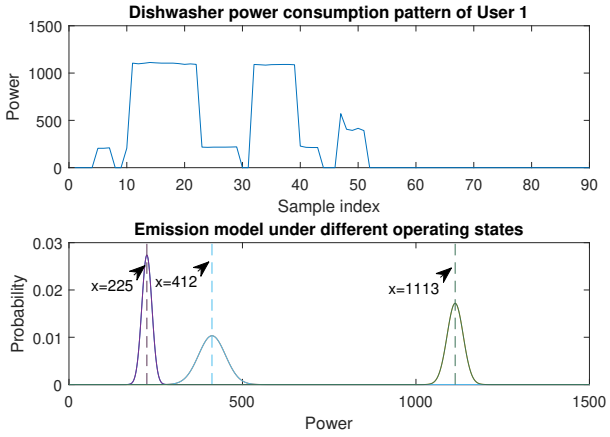


Figure 4.3: HMM emission model for dishwasher of user 1.

frequency power consumption samples of different appliances from different households in REDD dataset [81]. We re-sample the data within an interval of two minutes.

We first learn the HMM parameters for different appliances by utilizing the method proposed in [38]. As an example illustrated in Fig. 4.3, the dishwasher of user 1 has three different operating states, and the corresponding emission distributions at three different operating states are identified as three Gaussian distributions.

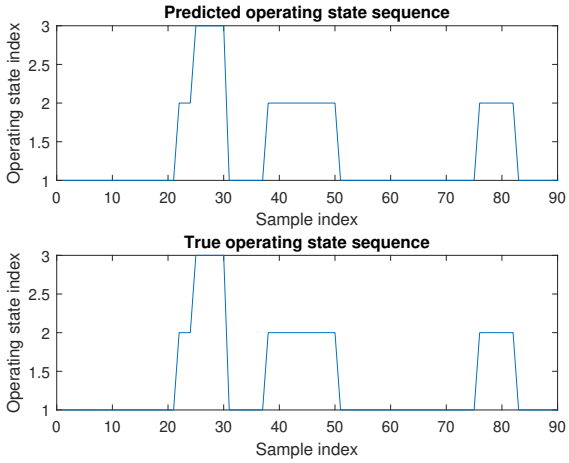


Figure 4.4: HMM load disaggregation of the true power consumption sequence.

Next, we provide a naive example to demonstrate the performance improvement of the designed privacy-preserving energy management strategy by only optimizing the privacy measure. In more details, we consider a special case where user 1 has only one appliance. As shown in Fig. 4.4, on receiving the true power consumption pattern of the refrigerator for user 1, the adversary can exactly identify the true operating states of the refrigerator by applying HMM load disaggregation [38]. To preserve the privacy, we design the corresponding privacy-preserving power requests by solving (4.5). As it is shown in Fig. 4.5, the privacy-preserving power request sequence will lead to a totally wrong load disaggregation analysis for the adversary.

In the following, we provide the simulation results for our privacy-preserving and cost-efficient power request design. In more details, we consider three users with their different appliances that involved in the non-cooperative game.

- user 1: dishwasher, light, and refrigerator.
- user 2: dishwasher, light, and refrigerator.
- user 3: dishwasher and washer-dryer.

We assume the adversary implements the load disaggregation every hour.

The power consumption patterns of different appliances of each user and the corresponding aggregated power consumption patterns are illustrated in Fig. 4.6. And the HMM emission parameters of different appliances are illustrated in Table 4.1. We further assume the adversary implements the FHMM load disaggregation based on the data samples collect in every one hour, i.e., $T = 30$ with sampling interval of two minutes. The parameters

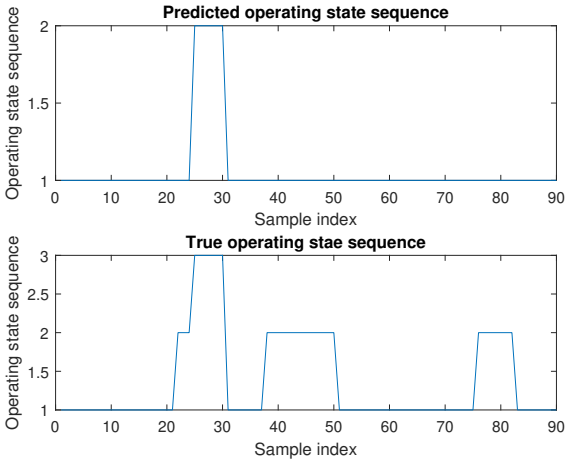


Figure 4.5: HMM load disaggregation of privacy-preserving power request sequence.

Table 4.1: HMM emission parameters for different users.

Appliances	State 1	State 2	State 3
user 1 dishwasher	$\mu = 225,\ \sigma = 15$	$\mu = 412,\ \sigma = 39$	$\mu = 1113,\ \sigma = 23$
user 1 light	$\mu = 81,\ \sigma = 2$		
user 1 refrigerator	$\mu = 1.7,\ \sigma = 2$	$\mu = 193,\ \sigma = 10$	$\mu = 425,\ \sigma = 7$
user 2 dishwasher	$\mu = 192,\ \sigma = 35$	$\mu = 737,\ \sigma = 10$	
user 2 light	$\mu = 24,\ \sigma = 3$	$\mu = 49,\ \sigma = 4$	$\mu = 193,\ \sigma = 13$
user 2 refrigerator	$\mu = 1,\ \sigma = 1$	$\mu = 114,\ \sigma = 6$	
user 3 dishwasher	$\mu = 130,\ \sigma = 12$	$\mu = 1317,\ \sigma = 18$	
user 3 washer-dryer	$\mu = 274,\ \sigma = 33$	$\mu = 713,\ \sigma = 59$	

for the pricing model are set as: $\rho_{base} = 0.5$ and $\alpha = 0.001$. All RESes are identical and are asummed to have time-invariant power capacities, i.e., $R_{1,t} = R_{2,t} = R_{3,t} = 1000$ for all $1 \leq t \leq 30$. The other parameters are set as: $\eta = 30$, $\nu_1 = 0.5$, $\nu_2 = 1$, $\nu_3 = 2$, and $\xi_0 = 1$. The convergence results of our proposed distributed relaxation algorithm are shown in Fig. 4.7. As we can see from the figure, the payoff functions of all users converge after 7 iterations. As a comparison, we consider the case where user 2 and user 3 still carry out the NE power management strategies, while user 1 takes an arbitrary power management strategy within the feasible set. As we can also see from Fig. 4.7, the payoff of user 1 reduces significantly compared to the payoff achieved at the NE, while user 2 and user 3 gain more payoffs compared to the payoffs achieved at the NE.

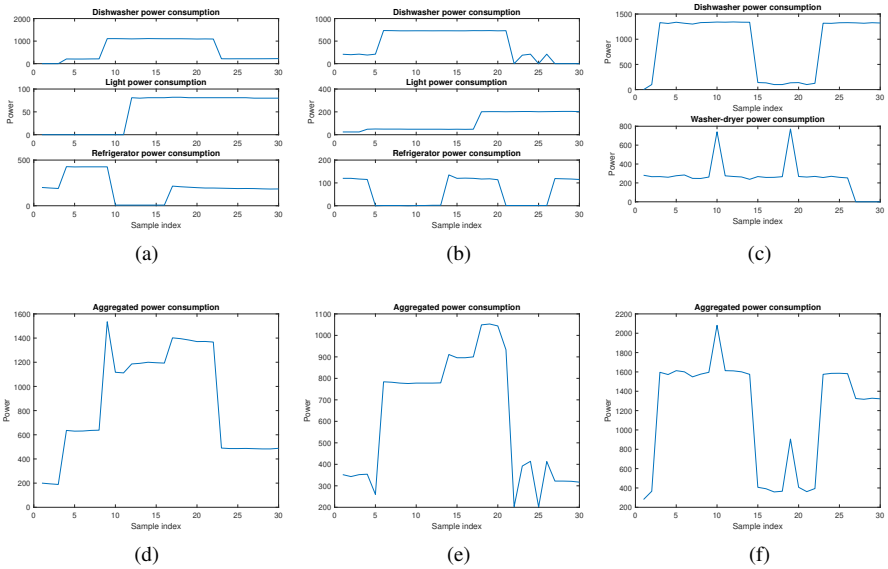


Figure 4.6: (a)-(c) Power consumption patterns of different appliances for three users. (d)-(f) Aggregated power consumption patterns for three users.

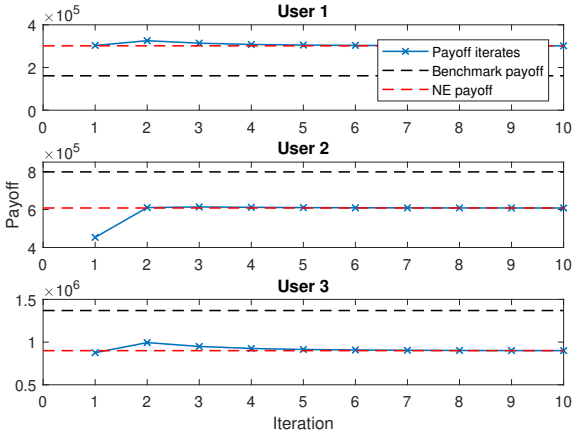


Figure 4.7: Convergence of the proposed distributed relaxation algorithm and the performance comparison between NE power management strategy with the non-NE benchmark power management strategy.

4.6 Summary

In this work, we have shown that renewable energy sources can be used for both privacy preserving and cost saving objectives. We have shown that the privacy-preserving problem under the FHMM-based adversarial NILM can be formulated as a power request design that minimizes the joint log-likelihood of the power request sequence and the operating state sequence. With the dynamic pricing model that is determined by all users' aggregated power request, the privacy-preserving and cost-efficient power request design problem becomes a non-cooperative game. Accordingly, an unique NE energy management strategy that trade off each user's privacy measure and the cost savings is designed to make sure that all users are satisfied. The complexity of solving the Nash equilibrium grows quickly, which calls for computationally efficient solutions. Our proposed distributed relaxation algorithm is computationally efficient and works for high-dimension problems. Our numerical experiments show that the distributed relaxation algorithm leads to a unique NE that trade off privacy-preserving and cost-saving so that all users are satisfied. In particular, they also show that optimizing our defined privacy objective can efficiently preserve the users' privacy against the FHMM-based adversarial NILM.

Part II

Data-Driven Self-Calibration for Gas Sensors

Chapter 5

HMM based Single Gas Sensor Calibration

In this chapter we focus on the problem of single NDIR sensor calibration, where the HMM is proposed to jointly model the stochastic relationship between the observed sensor measurements, the true calibration parameter, and the environmental temperature. The calibration of a single sensor is then performed by using statistical inference tools from the standard HMM framework. Particularly, we first develop a deterministic HMM which is shown to have a relatively high prediction accuracy within a short period after it has been trained. As the time evolves, the calibration performance of this deterministic HMM degrades, and this leads to our time-varying HMM formulation. Under the time-varying HMM framework, a time-adaptive EM learning approach is proposed to efficiently update the HMM parameters. The content of this chapter has been taken from **Paper C** and **Paper D**, while some parts have been verbatim copied.

This chapter is organized as follows: the deterministic HMM based drift process modeling approach is first presented in Section 5.1; Section 5.2 introduces the time-varying HMM based drift process modeling approach, where a time-adaptive EM based learning framework is further proposed to track the time-varying HMM. Finally, numerical results are provided in Section 5.3, and we conclude this chapter in Section 5.4.

5.1 Deterministic HMM based Stochastic Modeling of NDIR Sensor Drift Process

In this section, we propose to use an HMM to model the probabilistic relationship between the true calibration parameter and the observations of an NDIR sensor. To reduce the complexity of our proposed problem, we restrict all the relevant variables in our proposed model to lie in finite discrete spaces. We also assume that the environmental temperature at each time instance can be observed without any noise. As we mentioned before, the unobservable true calibration parameter X is adjusted according to the drift of I caused by

the temperature change. Thus, given fixed values for current i and temperature c, there is a one-to-one mapping relationship between the true CO_2 concentration level and the true calibration parameter x given by the deterministic mapping $g(\cdot)$. Since the true CO_2 level (or true calibration parameter) is not directly observable, we model the true calibration parameter as the hidden state that evolves with time. We use $\{X_t\}_{t=1}^T$ to denote the discrete-time stochastic process that describes the hidden states. While the sensor is operating using the calibration parameter r, we observe the noisy sensor measurement Y of the true CO_2 concentration level. Thus, we let $\{Y_t\}_{t=1}^T$ denote the discrete-time stochastic process of the observations of our proposed HMM. In this work, we focus on compensating the temperature dependency of the drift process. To this end, we let $\{C_t\}_{t=1}^T$ denote the discrete-time stochastic process of the temperature that is fully observed. The random variable $\Delta_t = C_t - C_{t-1}$ denotes the change of the temperature between two consecutive sampling instances. Experimental studies with the actual NDIR sensor data showed that it is sufficient to include only the temperature change to characterize the temperature dependency of the transition of the true calibration parameter, i.e., $P_{X_{t+1}|X_1^t,C_1^{t+1}} = P_{X_{t+1}|X_1^t,\Delta_1^{t+1}}$. At time step t, we further assume that the transition of the true calibration parameter only depends on the current state X_t and the temperature change from time t to time $t+1$, i.e.,

$$P_{X_{t+1}|X_1^t,\Delta_1^{t+1}} = P_{X_{t+1}|X_t,\Delta_{t+1}}. \tag{5.1}$$

We also assume the emission of the current observation Y_t only depends on the current state X_t. In this case, we end up with an HMM, which is fully characterized by the following distributions:

- Time-invariant transition probability: $X_{t+1} \sim P_{X_{t+1}|X_t,\Delta_{t+1}}$,
- Time-invariant emission probability: $Y_t \sim P_{Y_t|X_t}$,
- Prior distribution: $X_0 \sim P_{X_0}$.

We further assume that the true calibration parameter takes values from a finite integer set. Therefore, the hidden state corresponds to the quantized true calibration parameter. Assuming there are N such quantized values in total, hidden state X_t takes values from the set $\mathcal{X} = \{x_i\}_{i=1}^N \subset \mathbb{Z}$. Similarly, we assume that the CO_2 measurements can only take integer values within a certain range. In this case, given there are M possible CO_2 measurements in total, the random variable Y_t takes values from the set $\mathcal{Y} = \{y_j\}_{i=1}^M \subset \mathbb{Z}$. Likewise, the quantized temperature change takes values from the finite set $\bar{\Delta} = \{\delta_l\}_{l=1}^L \subset \mathbb{R}$.

For compactness, we denote the transition probability $P_{X_{t+1}|X_t,\Delta_{t+1}}(x_{i'}|x_i,\delta_l)$ as $A_{ii'|l}$. The set $A = \{A_{ii'|l}\}_{i,i',l}$ thus describes the transition behavior of the true calibration parameters given different temperature change values. Likewise, the emission probabilities are denoted by $B_i(j) = P_{Y_t|X_t}(y_j|x_i)$, $B = \{B_i(j)\}_{i,j}$, which is the likelihood of CO_2 measurement at time t given different X_t. Lastly, we define $\rho_i = P_{X_0}(x_i)$, $\rho = \{\rho_i\}_i$, as the initial prior distribution of the true calibration parameter.

Remark 8. *Given the above definitions, the parameters $\lambda = \{\pi, A, B\}$ fully characterize a statistical model of the drift process of a single NDIR sensor.*

The above HMM parameters can be learned via either a supervised approach or an unsupervised approach depending on whether the labeled training dataset is available or not. The calibration of a single NDIR sensor is then achieved by inferring the true underlying calibration parameter using the standard tools from the HMM framework, e.g., Viterbi decoding [75].

However, since the underlying dependency between the NDIR sensor components behavior and the temperature varies over time, the drift process of the NDIR sensor would also be time-varying. Correspondingly, instead of being time-invariant, the HMM parameters also varies over time. Thus, to maintain a good model performance, the HMM parameters should always be updated by re-training process that combines the recently incorporated data samples. However, this will further raise two main challenges regarding the computational efficiency and storage efficiency. On one hand, as time evolves, the size of the dataset also grows. It however would be inefficient to store all data samples and re-train the HMM each time using all historical data samples. On the other hand, as the dataset grows, re-training the HMM using large amount of data would be computationally complex and time-consuming. Additionally, as the EM algorithm already takes time to converge, it would be even more critical when the underlying stochastic process of the target model is varying rapidly and the re-training procedure needs to be done with high frequency. To address this, we next follow the concept of transfer learning [84] and propose a time-adaptive EM learning framework that can efficiently update the HMM according to the variation of the target model.

5.2 Time-Varying HMM based Stochastic Modeling of NDIR Sensor Drift Process

In this section, we focus on the problem of utilizing HMM to model the time-varying drift process of NDIR sensor. Besides, considering the fact that a reference measurement is usually not available, i.e., only the sensor measurements and the environmental factors can be observed while the values of true calibration parameter are missing, the EM based unsupervised learning approach is further proposed to deal with unlabelled dataset. In order to maintain a single stationary Markov chain for our HMM and reduce the computational complexity of the unsupervised learning algorithm, we make a slight modification for the propose HMM such that the hidden state is now defined as the pair of true calibration parameter and temperature change, i.e., $X_t = (Q_t, \Delta_t)$. And the rest part of the proposed HMM remains the same as it is defined in Section 5.1. In the following, before we present our designed algorithm, we first provide some general concepts on the time-varying model tracking problem.

Consider the scenario where the underlying stochastic process of the target model is non-stationary. In this case, the model parameters need to be always updated to avoid the model becoming out-of-date after a certain period, i.e., given a growing time series that

contains data samples generated by a non-stationary stochastic process, we wish to learn the parameters that can accurately characterize the statistical properties of the current target model. In the following, we first provide a condition that can be used to justify if the model is out-of-date.

Definition 5. *Let λ^{old} be the parameter which is most recently trained by a data sample sequence of length L. At time step t, assume λ^{t^*} is the optimal parameter of the current underlying model, i.e., the parameter that can the accurately characterize the statistical properties of the current target model. λ^{old} is then considered to be out-of-date at time step t if the following condition is satisfied*

$$\ln P(\bar{y}^t|\lambda^{t^*}) - \ln P(\bar{y}^t|\lambda^{old}) \geq \omega, \tag{5.2}$$

where $\bar{y}^t = [y_{t-L+1}, y_{t-L+2}, \ldots, y_t]$.

Once condition (5.2) is satisfied, the model parameter should be updated with the re-training process using the new dataset that incorporates recently collected data samples. Due to the issue of computational efficiency and time efficiency, a time-adaptive scheme should be designed to guarantee a fast convergence for the EM algorithm when re-train the model over time. According to [75], an appropriate initialization that fits the underlying model would result in a fast convergence of the EM algorithm. Thus, we next propose a time-adaptive EM algorithm, where the first iterate of each model updating phase is initialized as the learned HMM parameters from last model updating phase. In the following, we first provide details on our designed time-adaptive EM algorithm, and we will further provide the corresponding convergence rate for the proposed algorithm under certain assumptions.

5.2.1 Time-Adaptive EM based Time-Varying HMM Tracking

Let $p = [1, 2, 3, ..., \infty)$ be the index of of model training phases. At phase $p = 1$, we train the HMM model for the first time according to the standard procedure of Baum-Welch algorithm with a randomly selected initial HMM parameter estimate $\lambda_0^1 \in \Lambda$. For phase $p \geq 2$, let λ^p be the HMM parameters learned from phase p, the HMM model is also trained by using Baum-Welch algorithm but with the initial parameter estimate selected as $\lambda_0^p = \lambda^{p-1}$, i.e., we use the HMM parameters learned from the previous training phase as initialization for the current training phase. The corresponding algorithm design is summarized into Algorithm 5.

We next provide the convergence results of the above algorithm in each model training phase. For $p \geq 2$, denote y^{p-1} and y^p as the observation sequences in two consecutive training phases. Also, let X^{p-1} and X^p be the stochastic vectors that describe the hidden state sequences during phase $p-1$ and p, and denote their corresponding realizations as x^{p-1} and x^p. In phase p, we wish to find the HMM parameters $\lambda \in \Lambda$ that maximizes the

Algorithm 5: Time-adaptive EM algorithm for unsupervised HMM model updating

1: Initialization:
The initial model updating phase index $p = 1$, the threshold value γ, and a randomly selected initial parameter estimate $\hat{\lambda}_0^1 \in \Lambda$ for phase $p = 1$. Learn the HMM parameters λ^1 by following the standard procedures of Baum-Welch algorithm.
2: **for** $p \geq 2$ **do**
3: Initialize the parameter estimate as $\hat{\lambda}_0^p = \hat{\lambda}^{p-1}$. And the HMM parameters $\hat{\lambda}^p$ for phase p are learned according to the procedures of Baum-Welch algorithm
4: $p \leftarrow p + 1$
5: **end for**

likelihood defined in (2.21), i.e.,

$$\hat{\lambda} = \arg\max_{\lambda \in \Lambda} \ln \sum_{x^p \in \mathcal{X}^p} P(y^p, x^p|\lambda), \tag{5.3}$$

where $\mathcal{X}^p$ denotes the feasible set for the hidden state sequences during phase p. Correspondingly, given any parameter estimate λ', the Q-function then becomes

$$Q(\lambda|\lambda') = \sum_{x^p \in \mathcal{X}^p} \ln P(y^p, x^p|\lambda) P(x^p|y^p, \lambda'). \tag{5.4}$$

The EM algorithm then requires to optimize the above Q-function in each iteration. With the pre-defined HMM parameters, we have

$$P(y^p, x^p|\lambda) = \pi_{x_0^p} \prod_{t=1}^{T} A_{x_{t-1}^p x_t^p} B_{x_t^p}(y_t^p). \tag{5.5}$$

Plugging the above equation into (5.4), we have the following lemma.

Lemma 4. *With the factorization defined in (5.5), the Q-function defined in (5.4) can be re-written as*

$$Q(\lambda|\lambda') = \langle \ln \lambda, \vec{P} \rangle, \tag{5.6}$$

where $\forall i, k \in [1:K], j \in [1:M]$, *there is*

$$\lambda = [\{\pi_i\}_i, \{A_{ii'}\}_{i,i'}, \{B_i(j)\}_{i,j}], \tag{5.7}$$

and

$$\vec{P} = \begin{bmatrix} \{P(y^p, X_0^p = i|\lambda')\}_i \\ \{\sum_{t=1}^{T-1} P(y^p, X_{t-1}^p = i, X_t^P = i'|\lambda')\}_{i,i'} \\ \{\sum_{t=1}^{T} \mathcal{I}_{y_t^p}(j) P(y^p, X_t^p = i|\lambda')\}_{i,i'}, \end{bmatrix} \tag{5.8}$$

where $\mathcal{I}$ is the indicator function, i.e., $\mathcal{I}_{y_t^p}(j) = 1$, if $y_t^p = j$, $\mathcal{I}_{y_t^p}(j) = 0$, otherwise.

Proof: By plugging (5.5) into (5.4), the Q-function can be decomposed as

$$\begin{aligned} Q(\lambda|\lambda') = &\sum_{i=1}^{K} \ln \pi_i P(y^p, X_0^p = i|\lambda^{'}) \\ &+ \sum_{i=1}^{K}\sum_{k=1}^{K}\sum_{t=1}^{T-1} \ln A_{ik} P(y^p, X_{t-1}^p = i, X_t^p = k|\lambda^{'}) \\ &+ \sum_{i=1}^{K}\sum_{j=1}^{M}\sum_{t=1}^{T} \ln B_i(j)\mathcal{I}_{y_t^p}(j) P(y^p, X_t^p = i|\lambda^{'}). \end{aligned} \tag{5.9}$$

The above equation can be easily verified to be the inner product between the vector $\ln \lambda$ and the vector $\vec{P}$. □

Before we present the convergence results, we first make state on the following assumptions.

Assumption 1: There exist some statistical similarities or correlations between the complete dataset during phase $p-1$ and p, i.e.,

$$\ln \sum_{x^p \in \mathcal{X}^p} P(y^p, x^p|\lambda^{p^*}) - \ln \sum_{x^p \in \mathcal{X}^p} P(y^p, x^p|\lambda^{p-1}) \leq \epsilon, \tag{5.10}$$

where ϵ is a relatively small positive constant and λ^{p^*} denotes the optimal HMM parameters for phase p.

Remark 9. *Combing (5.2) and (5.10), for any $\omega < \epsilon$, also noticing $\sum_{x^p \in \mathcal{X}^p} P(y^p, x^p|\lambda^{p^*}) = \ln P(y^p|\lambda^{p^*})$, the model updating should be implemented when the following condition holds*

$$\omega \leq \ln P(y^p|\lambda^{p^*}) - \ln P(y^p|\lambda^{p-1}) \leq \epsilon, \tag{5.11}$$

which implies that the model should neither be updated when the underlying stochastic process has varied too much nor be updated when there is no obvious variation for the underlying stochastic process.

Remark 10. *To make sure (5.11) holds for each training phase p, one should frequently check if the following condition holds given any time step t*

$$\omega \leq \ln P(\bar{y}^t|\lambda^{t^*}) - \ln P(\bar{y}^t|\lambda^{old}) \leq \epsilon. \tag{5.12}$$

Notice that the optimal parameter λ^{t^*} at time t is unknown. In this case, we need to get the estimated optimal parameter $\hat{\lambda}^{t^*}$ by re-training the model using a new dataset that incorporates recently collected data points around time t, and evaluate the value of $\ln P(\bar{y}^t|\hat{\lambda}^{t^*}) - \ln P(\bar{y}^t|\lambda^{old})$ instead.

However, repeating this process to check if the model is out-of-data is not only time-consuming but also computationally inefficient, i.e., time and computational resources would be wasted if (5.12) turns out to be not true thus model updating is not necessary. As an alternative, we propose a time-adaptive scheme that can regularly update the model parameters. Under such a scheme, the model parameters are updated with fixed-length intervals, where the above test can be used to determine the appropriate time interval length[1].

Assumption 2: Given any HMM parameters $\lambda' \in \Lambda$, all elements in vector $\vec{P}$ are lower bounded by a positive number η.

With the above definitions and assumptions, we summarize the convergence results for our proposed algorithm into the following theorem.

Theorem 5. *If the HMM parameters for each single sensor are updated according to Algorithm 1, then in any model updating phase p the HMM parameter estimate λ_n^p will converge according to the following rate.*

$$\min_{n\in\{1,2,\ldots,k\}} \left\|\lambda_n^p - \lambda_{n-1}^p\right\|^2 \leq \frac{2\epsilon}{\eta k}, \tag{5.13}$$

where k denotes the number of iterations that has been ran in phase p.

Proof: The proof is provided in Appendix.

Corollary 4. *Even if Assumption 2 cannot be satisfied, by adding a strongly convex regularizer $\frac{\mu}{2}\left\|\lambda\right\|_2^2,\ \mu > 0$ to the Q-function, one could still achieve the following convergence rate*

$$\min_{n\in\{1,2,\ldots,k\}} \left\|\hat{\lambda}_n^p - \hat{\lambda}_{n-1}^p\right\|^2 \leq \frac{2\epsilon}{\mu k}, \tag{5.14}$$

where α is a positive number.

Remark 11. *Generally, the smallest element in $\vec{P}$ could be 0. For such cases, we could still achieve the convergence rate in (5.13) by maximizing the regularized Q-function $Q(\lambda|\lambda^{'}) - \frac{\mu}{2}\left\|\lambda\right\|_2^2$ in each EM iteration.*

5.3 Numerical Results

In this section, we present numerical results of our proposed calibration framework. The experiments are carried out based on the data collected from the sensors integrated at the air quality monitoring station in Düebendorf in Switzerland.[2] The dataset contains data collected from 18 low-cost Senseair LP8 NDIR CO_2 sensors, and a high-accuracy CO_2 reference sensor over a period from 2017 to 2019.

[1]This fixed length can also be decided by visualizing the real problem performance, e.g., use the recently trained HMM parameter to perform the Viterbi decoding on different parts of the time-series, and decide on an appropriate interval length based on the decoding accuracy. More details on this method will be illustrated later in the numerical results section.

[2]The sensor network is deployed by EMPA.

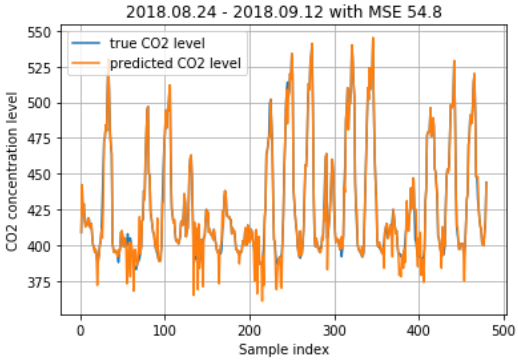


Figure 5.1: Accurate prediction for 20 days immediately after the initial training phase.

Table 5.1: Quantization of Δ_t.

Interval	Level	Interval	Level
$(-\infty, -0.2)$	1	$(0.2, +\infty)$	2
$[-0.2, 0.2]$	3		

5.3.1 Prediction Using the Fixed HMM

We first use the data collected from January 2018 to August 2018 to initially train the HMM.[3] In more details, the training data sequence is generated by sampling the original data sequence with 1 hour interval so that the length of training data sequence is 6000. We restrict that the possible CO_2 concentration levels and the true calibration parameter can only take integer values and lie in the range $[300ppm, 600ppm]$, and $[12400, 13000]$ respectively. We further round the true calibration parameter to values $12400 : 5 : 13000$, which is 61 different integer levels in total. Besides, we quantize the temperature change Δ_t into three different levels as shown in Table 5.1. The initial HMM is trained by applying the EM algorithm with a random initialization using the above training dataset. By utilizing the Viterbi decoding, the predicted CO_2 concentration levels (derived by calculation based on (4.10) using the predicted true calibration parameter) in the next 6 months are compared with the reference values.

The immediate prediction result is shown in Fig. 5.1. As we can seen from the figure, the accuracy of the immediate prediction is quite high since the model is up-to-date and the HMM parameters can accurately characterize the current drift process. However, as the time evolves, the calibration performance degrades. As we can see from the first figure of Fig. 5.2, the prediction accuracy starts to decrease at the end of that phase, which is around

[3] Due to the shut down of devices or some other unpredictable reasons, the whole dataset contains large parts of missing data samples. Thus, we select the data sequence within this range to avoid having long missing data sequence included.

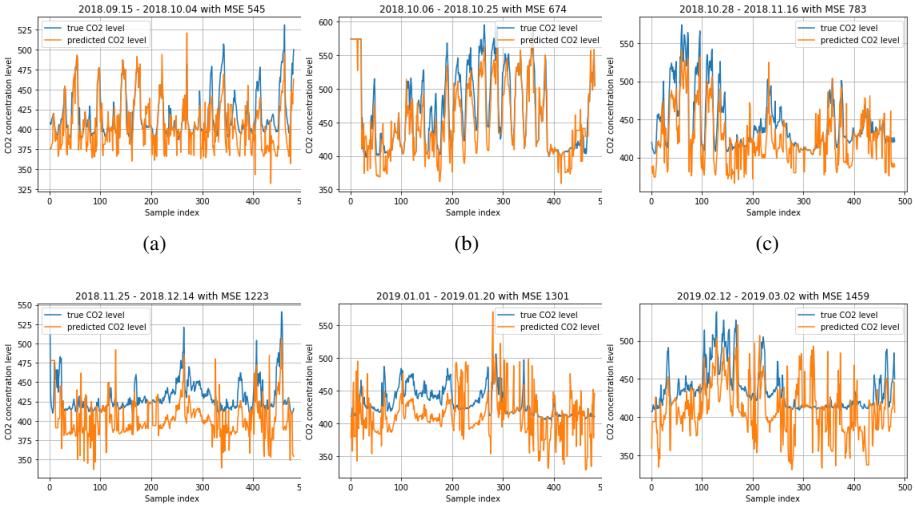


Figure 5.2: Predicted CO_2 concentration levels using the initial HMM compared with the true CO_2 concentration levels for 6 different 20-days periods within 2018.09-2019.03.

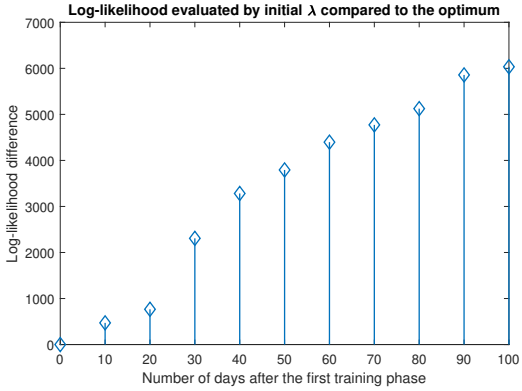


Figure 5.3: Evaluation of log-likelihood differences at different time steps.

one month after the initial HMM has been trained. Meanwhile, as we can see from the other plots in Fig. 5.2, the calibration performance keeps degrading as time evolves, which means the initial HMM becomes more and more inaccurate and is not able to characterize the later drift processes. As it can be also seen from the plots as well as the MSE values in Fig. 5.2, even though the prediction accuracy of the initial model starts to decrease, it can still be maintained at a certain level for around one month and not going to decrease

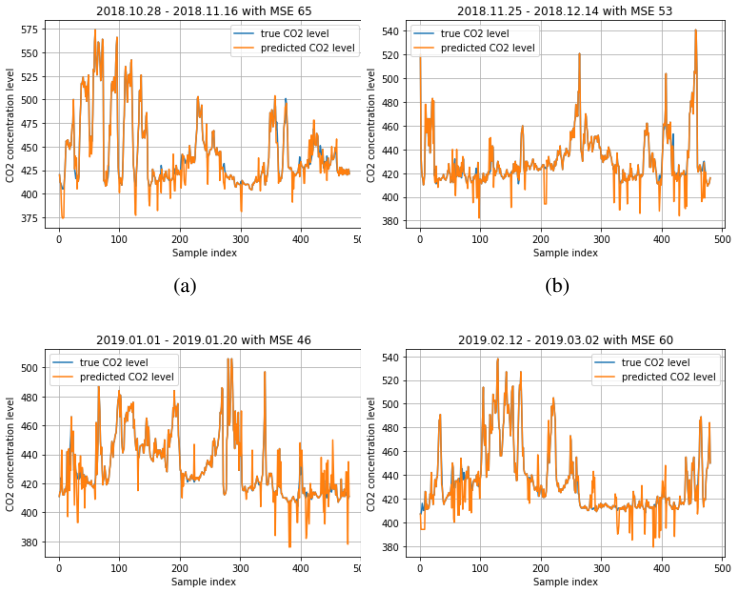


Figure 5.4: Predicted CO_2 concentration levels using the updated HMM compared with the true CO_2 concentration levels for 4 different 20-days periods within 2018.10-2019.03.

too much. In this case, the length of the intervals between model updates should be chose as any duration between 1-2 months depending on the desired prediction accuracy, i.e., the shorter the length the higher the prediction accuracy. To verify this, we further provide a figure that shows the differences of log-likelihood as defined in (5.2). Fig. 5.3 illustrates the variation of the differences $\ln P(\bar{y}^t|\hat{\lambda}^{t^*}) - \ln P(\bar{y}^t|\lambda^{old})$ at different time steps after the initial training phase, where the estimate of the current optimal parameter $\hat{\lambda}^{t^*}$ is learned by utilizing the training dataset that builds on 3000 consecutive data samples before the current time step. As we can see from the figure, the difference of the log-likelihood has an obvious increase between 20-40 days and keeps increasing afterwards. This coincides with the conclusion we draw by visualizing the prediction accuracy in Fig. 5.2 and shows that the initial HMM becomes more and more inaccurate. Combining Fig. 5.2 and Fig. 5.3, we can also determine possible values of ω and ϵ for our time-adaptive algorithm, e.g., update the model when $2000 \leq \ln P(\bar{y}^t|\lambda^{t^*}) - \ln P(\bar{y}^t|\lambda) \leq 4000$.

5.3.2 Prediction Using the Time-adaptive HMM

Next, we provide results to demonstrate the performance of the proposed time-adaptive HMM. After the initial HMM has been trained, we re-train the model after every 30 days using 3000 consecutive data samples before that re-training time step. And the initializa-

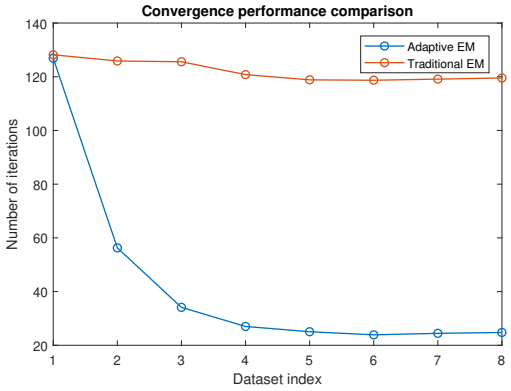


Figure 5.5: Comparison of the average convergence speed of our proposed scheme and the benchmark scheme.

tion of the EM algorithm in each re-training phase is set as the HMM parameter that has been learned during last training phase. The calibration performance of our proposed algorithm is illustrated in Fig. 5.4. Comparing the results with Fig. 5.2, it can be seen that the prediction accuracy of our proposed time-adaptive HMM is significantly better than the prediction accuracy of using a fixed HMM.

5.3.3 Convergence Analysis

Lastly, we provide simulations results illustrating the convergence of our designed algorithm. The experiment is carried out 1000 rounds, where in each round the initial HMM is trained with a different random initialization. We further re-train the HMM 7 times after every 30 days using our proposed scheme and the standard EM algorithm (with a random initialization). The convergence threshold γ in (2.25) is set to be 0.0001, and the algorithm runs until (2.25) is satisfied during each training phase. The average number of iterations that is needed for convergence in each training phase for both schemes is illustrated in Fig. 5.5. As it is shown in the figure, our proposed scheme significantly improves the convergence speed as time evolves.

5.4 Summary

In this chapter, we developed a time-adaptive framework to learn parameters of a hidden Markov model used for data-driven calibration of low costs NDIR gas sensors which drift. The data-driven calibration routine is described by a statistical inference problems on the hidden state that describes the true calibration parameter. It is shown how a time-adaptive expectation maximization learning framework that can be used to efficiently update the hidden Markov model to track the time-varying drift process of the sensor. Compared to

the previous works that use a fixed HMM to predict the true calibration parameter, our proposed framework significantly improves the prediction accuracy over the whole lifetime of the sensors. Moreover, our designed framework can be seen as a transfer learning approach that (i) always achieves a fast convergence rate with a relatively small training data set and (ii) allows to discard previous measurements and thereby systematically prevents a big data storage problem due to growing stored data for training. This shows the great value of our designed framework regarding data efficiency, computational efficiency, and time efficiency.

Chapter 6

Joint Calibration for Gas Sensor Network

The previous HMM based self-calibration method only works for a single sensor. It will perform poorly when the stochastic calibration model does not characterize the drift behavior accurately or when the model becomes out-of-date. The uncertainty of the single sensor measurement and calibration can be reduced by deploying multiple sensors to monitor the gas concentration. Thus, in this chapter, we consider the networked calibration procedure in a gas sensing system, where a joint calibration is performed on multiple sensors. In the following, we first introduce the structure our gas sensing system. Based on this model. we further propose two different joint calibration algorithms for the NDIR sensor network. The content of this chapter has been taken from **Paper C**, while some parts have been verbatim copied.

This chapter is organized as follows: The gas sensing system model is presented in Section 6.1. Section 6.2 introduces the method to derive the belief functions by using the HMM of each sensor. The belief Function Fusion based self-calibration algorithm and the deep reinforcement learning based self-calibration algorithm are provided in Section 6.3 and 6.4. The performance of our proposed calibration algorithms is validated via experiments in Section 6.5. We conclude the chapter in Section 6.6.

6.1 System Overview

Consider an NDIR CO_2 sensor system with multiple NDIR sensors that measure the time-varying CO_2 concentration level in the same environment over a certain time horizon, i.e., the sensors are measuring the same CO_2 level at each time instant. By using the hidden Markov model framework in Section 5.1, each sensor can compute its own posterior distribution of the current true CO_2 level given historical measurements, i.e., belief functions of the current true CO_2 level, at the local computing node. Since the sensors are unaware of the true CO_2 level, to infer on the current true CO_2 level, the sensors will

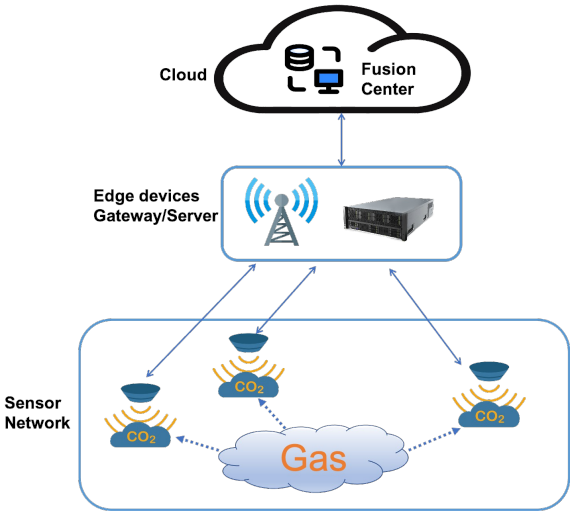


Figure 6.1: Working mechanism of CO_2 sensing system with multiple locally distributed sensors, edge devices, and one fusion center in the cloud.

send their belief functions on the true CO_2 level to a fusion center in the cloud[1]. After running certain designed joint calibration algorithms, the cloud fusion center will send the calibration results back to all the local sensors through the gateway. The system model described above is depicted in Fig. 6.1. In order to store the data sent from the sensors and compute the calibration results for all sensors, the cloud fusion center should be composed by certain computing nodes and storage nodes. The amount of needed computational resources depends on the choice of the algorithms as well as the dynamic of the environment. There always be trade-off between the amount of required computational resources and the calibration accuracy. If we can assume that the environment is slowly varying compared to the sensing interval, then there is sufficient time for the cloud fusion center to do the computation before the true CO_2 level changes. As long as the fusion result is computed and sent back to the sensors before the true CO_2 level changes, it would be valid for the current period and can be utilized as reference to calibrate all sensors. In case the environment varies fast, the cloud needs to do fast computation and give the sensors a quick response. However, the fast computation can still be achieved by limited computational resources with lower calibration accuracy, e.g., have more rough quantization for the true

[1]Note that all the sensors should transmit belief functions that correspond to measurements done during the same period while the environmental conditions remain sufficiently unchanged. Otherwise, the calibration result will be bad since the belief functions on true CO_2 levels of different environments will be fused. This requires that the sensors synchronously take the measurement during the same period while the environmental conditions remain sufficiently unchanged. In case the environment is slowly varying, the condition for the synchronous sensing will be relaxed since the sensors can take the measurement at any time in a relatively long period while the environment remains unchanged.

CO_2 levels to make the belief function vectors shorter. Otherwise, since there will be a delay caused by the limited computational resources, sensors need to store intermediate CO_2 and temperature measurements during the delay period, do the calibration at a later stage when they receive the calibration result then replay the belief function evolution.

6.2 On Deriving the Belief Function

Assume we have $\bar{N}$ sensors in total. Let $z_t \in \mathcal{Z}$ denote the true CO_2 level of the environment at time t. As before, we quantize the true CO_2 level into L integer values. Thus, $\mathcal{Z} = \{z_l\}_{l=1}^{L} \subset \mathbb{Z}$ denotes the set of all possible quantized true CO_2 levels. Similarly, we define $i_t \in \mathcal{I}$ as the current value at time t, where the set $\mathcal{I}$ contains $\tilde{L}$ integer values within a certain range, i.e., $\mathcal{I} = \{i_{\tilde{l}}^{\tilde{L}}\} \subset \mathbb{Z}$.

Given the above notations, we provide the following definitions of the involved belief functions:

- At time t, the posterior distribution of the true calibration parameter (*belief function of true calibration parameter*) of sensor n is defined as $P(x_t^n|y_{1:t-1}^n, c_{1:t})$, where x_t^n denotes the true calibration parameter of sensor n at time t, $y_{1:t-1}^n$ denotes the historical CO_2 measurement from the first time step until time $t-1$ of sensor n, and $c_{1:t}$ is the historical environmental temperature sequence from the first time step until time t.
- At time t, the posterior distribution of the true CO_2 level (*belief function of true CO_2 level*) of sensor n specified by the current value i_t^n is defined as $P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t})$.
- At time t, the joint posterior distribution of the true CO_2 level (*belief function of true CO_2 level*) of all sensors at time t is defined as $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$, where $y_{1:t-1}^{1:\bar{N}}$ denotes the historical CO_2 measurement sequence of all $\bar{N}$ sensors.

Utilizing the HMM parameters for single sensor and the Bayesian rule, the belief function of the true calibration parameter can be updated according to the following lemma.

Lemma 5. *The belief function $P(x_t^n|y_{1:t-1}^n, c_{1:t})$ can be updated according to the following rule*

$$P(x_t^n|y_{1:t}^n, c_{1:t}) = \frac{P(x_t^n|y_{1:t-1}^n, c_{1:t})P(y_t^n|x_t^n)}{\sum_{x_t^n \in \mathcal{X}} P(x_t^n|y_{1:t-1}^n, c_{1:t})P(y_t^n|x_t^n)}, \forall x_t^n \in \mathcal{X}. \tag{6.1}$$

$$P(x_{t+1}^n|y_{1:t}^n, c_{1:t+1}) = \sum_{x_t^n \in \mathcal{X}} P(x_t^n|y_{1:t}^n, c_{1:t})P(x_{t+1}^n|x_t^n, y_{1:t}^n, c_{1:t+1}), \forall x_{t+1}^n \in \mathcal{X}. \tag{6.2}$$

Proof: According to the definition of the emission probability in our HMM, we then have the Markov chain $y_t^n - x_t^n - (y_{1:t-1}^n, c_{1:t})$, which leads to $P(y_t^n|x_t^n) = P(y_t^n|x_t^n, y_{1:t-1}^n, c_{1:t})$.

Plugging $P(y_t^n|x_t^n) = P(y_t^n|x_t^n, y_{1:t-1}^n, c_{1:t})$ into (6.1), the equation is easily to be verified. Besides, (6.2) holds since we have $P(x_{t+1}^n|x_t^n, y_{1:t}^i, c_{1:t+1}) = P(x_{t+1}^n|x_t^n, \delta_{1:t+1}) = P(x_{t+1}^n|x_t^n, \delta_{t+1})$ according to (5.1), and $P(x_t^n|y_{1:t}^n, c_{1:t}) = P(x_t^n|y_{1:t}^n, c_{1:t+1})$ according to the Markov chain $x_t^n - (y_{1:t}^n, c_{1:t}) - c_{t+1}$. Thus, given the HMM parameters $\{\pi^n, A^n, B^n\}$ for sensors n, the belief function $P(x_t^n|y_{1:t-1}^n, c_{1:t})$ at any time step t can be calculated according to (6.1) and (6.2) by initializing the belief function as the prior distribution π^n for sensor n. □

Remark 12. *At time t, given fixed i_t^n and c_t, $\forall z_t \in \mathcal{Z}$, if there is any $\forall x_t^n \in \mathcal{X}$ that satisfies $z_t = g(x_t^n, i_t^n, c_t)$, the belief function of true CO_2 level $P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t})$ is then calculated by*

$$P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t}) = P(x_t^n = h(z_t, i_t^n, c_t)|y_{1:t-1}^n, c_{1:t}), \tag{6.3}$$

where the function $h(\cdot)$ is the inverse of function $g(\cdot)$ such that $g(h(z_t, i_t^n, c_t), i_t^n, c_t) = z_t, \forall i_t^n, c_t$, which is a one-to-one mapping between z_t and x_t^n when i_t^n and c_t are fixed. Otherwise, $P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t}) = 0$.

With the belief function of true CO_2 level calculated according to Remark 12, we next propose a method to fuse these belief functions. For notation simplicity, we use $P_n(z_t)$ to denote the belief function of the true CO_2 level for sensor n at time t and $P(z_t)$ to denote the joint belief function of the true CO_2 level for all sensors at time t in the following part of this section.

6.3 Belief Function Fusion based Self-Calibration for NDIR Sensor Network

For this calibration scheme, the belief function fusion approach will be performed at the cloud fusion center to get a fused belief function. The fused belief function can finally be adopted as updated belief by all sensors (calibration). Assume there are $\bar{N}$ sensors in total and the belief functions provided by all sensors are reliable. Our objective then is to design a fusion rule to combine the beliefs provided by all $\bar{N}$ sensors. The DS rule [59] provides a general framework for such a reasoning problem with uncertain information or partial information. Let $n, m \in \{1, 2, ..., \bar{N}\}$ be the indices of two sensors. We first consider the most simple two sensor fusion case, where we wish to combine the belief functions provided by sensor n and sensor m. Applying the DS combination rule with singleton sets results in the following fusion rule

$$(P_n \oplus P_m)(z_l) = \frac{1}{1-F} P_n(z_l) P_m(z_l), \forall z_l \in \mathcal{Z}, \tag{6.4}$$

where $F = \sum_{\substack{z_{l_1}, z_{l_2} \in \mathcal{Z} \\ z_{l_1} \neq z_{l_2}}} P_n(z_{l_1}) P_m(z_{l_2})$ denotes the normalizing factor. The operator $\oplus$ denotes the combination operator via DS rule.

For the N sensors fusion case, the N-fold extension of the DS rule with singleton sets results in

$$P(z_l) = (P_1 \oplus P_2 \oplus ... \oplus P_{\bar{N}})(z_l) = \frac{1}{F'} P_1(z_l)P_2(z_l)...P_{\bar{N}}(z_l), \forall z_l \in \mathcal{Z}, \tag{6.5}$$

where the normalizing factor is $F^{'} = \sum_{z_l \in \mathcal{Z}} P_1(z_l)P_2(z_l)...P_{\bar{N}}(z_l)$. The fused belief $P(z)$ can be then further used by all the sensors.

The DS rule will result in unreasonable fused beliefs when the belief functions provided by different sensors highly conflict with each other, e.g., the DS rule will fail when some sensors hold highly different belief functions compared to the other sensors in particular when the intersection of the support sets of the beliefs is empty. To address this issue, we utilize the weighted average approach with a weighting of each belief function based on its similarity with other belief functions. To measure the similarity, we propose to use the Wassertein distance as a distance measure between the belief functions. The Wasserstein distance is suitable for the case where the underlying support sets of two probability measures are very different.

Let X and Y be two random variables with probability distribution P_Y and P_Z. The Wasserstein distance W_2 between P_X and P_Y is defined as

$$W_2(P_X, P_Y) = \sqrt{\min_{P_{XY}: \sum_x P_{XY} = P_Y, \sum_y P_{XY} = P_X} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} |x - y|^2 P_{XY}(x, y)}. \tag{6.6}$$

The distance between the belief functions of sensor n and sensor m is then given by $W_2(P_n(z), P_m(z)), \forall n, m \in \{1, 2, ..., \bar{N}\}$. In the following, we follow the setup as proposed in [65]. In more detail, after getting the distance between each pair of sensors, we normalize all distances into the interval $[0, 1]$ as follows

$$\hat{W}_2(P_n(z), P_m(z)) = \frac{2 \times W_2(P_n(z), P_m(z))}{\sum_n \sum_m W_2(P_n(z), P_m(z))}. \tag{6.7}$$

With the normalized distance measure provided above, we define the similarity measure between belief functions $P_i(z)$ and $P_j(z)$ as

$$S(P_n(z), P_m(z)) = 1 - \hat{W}_2(P_n(z), P_m(z)). \tag{6.8}$$

We next define the support degree of a given belief function $P_i(z)$ as

$$Supp(P_n(z)) = \sum_{m=1, m \neq n}^{\bar{N}} S(P_n(z), P_m(z)), \tag{6.9}$$

which characterizes the relative importance of the belief function P_i. The weighting factor of belief function P_i is then obtained by the following normalization

$$\alpha_n = \frac{Supp(P_n(z))}{\sum_{n=1}^{\bar{N}} Supp(P_n(z))}. \tag{6.10}$$

The weighted average of all $\bar{N}$ belief functions can be expressed as $\hat{P}(z) = \sum_{i=1}^{\bar{N}} \alpha_n P_n(z)$. The Wasserstein metric based fused belief function is finally obtained by using the DS rule to combine $\hat{P}(z)$ for $N-1$ times

$$P(z) = (\hat{P} \oplus \hat{P} \oplus ... \oplus \hat{P})(z), \tag{6.11}$$

where we apply '$\oplus$' for $N-1$ times.

Remark 13. *The probability mass function $P(z_t)$ derived by the above procedure, which is the fusion result of the belief functions $P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t}), \forall n \in \{1, 2, ..., \bar{N}\}$, is used as an approximation of the joint belief function $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ since it is difficult to obtain a joint statistics of the whole system. We denote this approximated belief function as $P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ which is further used in the next section, where $i_t^{1:\bar{N}}$ denotes the sequence of the current values for all $\bar{N}$ sensors at time t.*

6.4 Sequential Self-calibration for NDIR Sensor Network via Deep Reinforcement learning

The previous modified belief function fusion method only provides an instantaneous solution for the sensor calibration problem instead of considering the overall long-term performance. The advantage is that it is simple to be implemented since it does not require the statistical relation between sensors, i.e., joint statistics. It can work well in scenarios where the majority of the belief functions provide reliable and correct evidence. Moreover, the fusion result provided by this approach can be used as an approximation of the joint belief function $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$, which is then used as the joint statistics of the system. And this joint statistics allows us to propose the following reference belief function selection problem, which is further reformulated into a POMDP problem that with an objective considering the impact from both past and future.

6.4.1 POMDP Problem Formulation for Reference Belief Selection

Under the CO_2 sensing system model as illustrated in Section 6.1, assume we have $\bar{N}$ sensors in total. At time step t, each sensor sends its belief function on the current true CO_2 level to the fusion center (decision maker) at the cloud. The fusion center will further decide on the best belief function (named reference belief function) $P_{b^*}(z_t)$, where b^* is the index of the sensor with the best belief. The reference belief $P_{b^*}(z_t)$ is then sent back to all sensors. Each sensor then will update its own belief function as the reference belief $P_{b^*}(z_t)$.

Considering the whole time horizon, the reference belief selection needs to be performed in a sequential manner such that the overall calibration scheme can achieve a long-term stable performance. In the following, we provide details on how to reformulate the sequential reference belief selection problem into a POMDP. According to the process described above, at time step t, the calibration action of the fusion center is defined as

$$H(Z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, Y_t^{b_t}) = \sum_{y_t^{b_t}} P(y_t^{b_t}|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) \sum_{z_t} P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, y_t^{b_t}) \log P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, y_t^{b_t}) \tag{6.12}$$

$b_t \in \mathcal{B} = \{1, 2, ..., \bar{N}\}$. Thus the control action decides on the index of the sensor, whose belief function on the true CO_2 level will be selected as reference belief. As the next step, we need to decide upon an objective function, i.e., the cost function (reward function) of the POMDP, which measures the calibration performance. Previous research such as [85] has demonstrated the effectiveness of conditional entropy as such an objective function. Given the calibration action b_t, we propose to use the conditional entropy as the objective function as shown in (6.12). An intuitive explanation of the above objective is that we wish to select a reference sensor, whose individual statistical model of this sensor brings least uncertainty when guessing the current true CO_2 level given the observations of CO_2 measurements and temperature.

As shown in (6.12), the cost does not only depend on the current 'true' state and the control action but also depends on the historical sequences $(y_{1:t-1}^{1:\bar{N}}, c_{1:t})$. In order to formulate it into a standard MDP, we introduce belief states which will be used to replace the growing historical sequences by identifying an update rule. To this end, we identify the belief state $\theta_t \triangleq P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ as the state of our POMDP problem. However, as we mentioned before, the joint statistics $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ is difficult to be obtained. Thus, we define $\tilde{\theta}_t \triangleq P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ as the approximation of the joint statistics $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$, where $P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ is the fusion result as defined in Remark 13. At each time step, on observing the sequences $(y_{1:t-1}^{1:\bar{N}}, i_t^{1:\bar{N}}, c_{1:t})$, the fusion center will obtain an approximation of the joint statistics, i.e., approximated belief state. Based on this approximation, the fusion center will further choose a calibration action according to the calibration policy. Accordingly, the calibration can be defined as $\pi_t \in \mathit{\Pi}_t$, where $\mathit{\Pi}_t$ denotes the set of deterministic mappings from approximated belief state $\tilde{\theta}_t$ to a corresponding action b_t, i.e., $b_t = \pi_t(\tilde{\theta}_t)$. Correspondingly, the calibration policy over a T-time horizon is $\pi = \{\pi_t\}_{t=1}^T \in \mathit{\Pi} = \mathit{\Pi}_1 \times \mathit{\Pi}_2 \times .... \times \mathit{\Pi}_T$.

With the elements for the POMDP defined above, considering long term calibration performance, our overall objective is to find the calibration policy $\pi \in \mathit{\Pi}$ that minimizes the expected discounted total cost

$$L_T(\pi) = \mathbb{E}_\pi\big\{\sum_{t=1}^{T} \gamma^t H(Z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, Y_t^{b_t})\big\}, \tag{6.13}$$

where $\gamma \in [0, 1)$ is the discounted factor to determine the weight of the future cost. $\gamma = 0$ means that we only care about the immediate reward. When $\gamma < 1$, the costs in the future will cause less impact than the costs of the earlier time steps.

And the optimal calibration policy is defined as

$$\pi^* = \arg\min_{\pi \in \Pi} L_T(\pi). \tag{6.14}$$

Given a state-action pair (θ_t, b_t) and the approximated belief state $\tilde{\theta}_t$, we provide the following lemma.

Lemma 6. *The per-step cost defined in (6.12) can be written as a function of the state-action pair (θ_t, b_t), and can be further approximated by substituting θ_t with $\tilde{\theta}_t$.*

Proof: The proof is provided in Appendix A.8.

In this case, the per-step cost at time t is defined as the following function of state-action pair (θ_t, b_t)

$$l_t(\theta_t, b_t) = H(Z_t | y_{1:t-1}^{1:\bar{N}}, i_{1:t}^{1:\bar{N}}, c_{1:t}, Y_t^{b_t}), \tag{6.15}$$

and we denote the approximated per-step cost as $l_t(\tilde{\theta}_t, b_t)$.

With the above model, we next identify the system dynamics, e.g., evolution of the approximated belief state $\tilde{\theta}_t$ given a certain control action b_t^*. The evolution is summarized into the following two steps.

- At time step t, given assume the fusion center decides on the index of the reference sensor as b_t^*, the belief function $Pi_t^{b_t^*}(z_t | y_{1:t-1}^{b_t^*}, c_{1:t})$ is then sent back to all sensors as the reference belief function. After receiving the reference belief function, each sensor n will first update its own belief function $P(x_t^n | y_{1:t-1}^n, c_{1:t}), \forall n \in \{1, 2, ..., \bar{N}\}$ (belief function of current true calibration parameter). Given fixed i_t^n and c_t, $\forall x_t^n \in \mathcal{X}$, if there is any $\forall z_t \in \mathcal{Z}$ that satisfies $z_t = g(x_t^n, i_t^n, c_t)$, $P(x_t^n | y_{1:t-1}^n, c_{1:t})$ can be then calculated according to the following equation

$$P_{new}(x_t^n | y_{1:t-1}^n, c_{1:t}) = P_{i_t^{b_t^*}}(z_t = g(x_t^n, i_t^n, c_t) | y_{1:t-1}^{b_t^*} c_{1:t}), \tag{6.16}$$

  where the function $g(\cdot)$ is the mapping function defined in (4.10).
  Otherwise, $P_{new}(x_t^n | y_{1:t-1}^n, c_{1:t}) = 0$. Thus, under the control action b_t^*, each sensor n has now an updated belief function $P_{new}(x_t^n | y_{1:t-1}^n, c_{1:t})$.

- At time step $t+1$, for all sensors, the updated belief function $P_{new}(x_t^n | y_{1:t-1}^n, c_{1:t})$ will evolve to $P(x_{t+1}^n | y_{1:t}^n, c_{1:t+1})$ according to equation (6.1) and (6.2) in Section 6.2. Each sensor then calculates $P_{i_{t+1}^n}(z_{t+1} | y_{1:t}^n, c_{1:t+1})$ using $P(x_{t+1}^n | y_{1:t}^n, c_{1:t+1})$ according to (6.3). And the new belief state $\theta_{t+1} = P(z_{t+1} | y_{1:t}^{1:\bar{N}}, c_{1:t+1})$ is approximated by $P_{i_t^{1:\bar{N}}}(z_{t+1} | y_{1:t}^{1:\bar{N}}, c_{1:t+1})$ which is obtained by fusing $P_{i_{t+1}^n}(z_{t+1} | y_{1:t}^n, c_{1:t+1})$ of all sensors via the belief function fusion approach provided in Section 6.3.

According to the definition of POMDP in [70, pp.150-151], the above formulations and derivations prove the following theorem.

Theorem 6. *The sequential self-calibration for the NDIR sensor network can be formulated as a POMDP problem such that: (i) the state at time t is given by a belief state $\theta_t \triangleq P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$; (ii) the control action at time t is specified by $b_t \in \{1, 2, ..., \bar{N}\}$; (iii) the per-step expected cost corresponding to a state-action pair is given by (6.15); (iv) given a control action b_t, the belief state will evolve according to the two-step process described above; and as a consequence we have that (v) the corresponding optimal calibration policy π^* can be derived by using Bellman dynamic programming.*

Remark 14. *Due to difficulties in obtaining the joint statistics θ_t, we alternatively use $\tilde{\theta}_t \triangleq P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ as the approximated belief state of our POMDP problem. Accordingly, all the calculations involved in the Bellman dynamic programming is done by using the state-action pair $(\tilde{\theta}_t, b_t)$ instead of using (θ_t, b_t).*

6.4.2 Bellman Dynamic Programming based Networked Calibration over Finite Time Horizon

By applying the Bellman operator [70], the value function $V_t(\theta_t)$ is updated according to

$$V_t(\tilde{\theta}_t) = \min_{b_t \in \{1,2,...,\bar{N}\}} l_t(\tilde{\theta}_t, b_t) + \gamma \mathbb{E}_{P(\tilde{\theta}_{t+1}|\tilde{\theta}_t, b_t)}[V_{t+1}(\tilde{\theta}_{t+1})], \tag{6.17}$$

where $\gamma \in (0, 1)$ is the discount factor and the term $\mathbb{E}_{P(\tilde{\theta}_{t+1}|\tilde{\theta}_t, l_t)}[V_{t+1}(\tilde{\theta}_{t+1})]$ denotes the expected cost-to-go. The optimal calibration policy $\pi_t^*(\tilde{\theta}_t)$ and the corresponding optimal calibration action $b_t^* = \pi_t^*(\tilde{\theta}_t)$ is given by the optimizer of (6.17). Further, the minimum value of the expected discounted total cost is given by $V(\tilde{\theta}_1)$.

Unfortunately, the dynamic programming only provides a non-stationary solution for the above finite time horizon POMDP problem. Also note that the state space of our problem is continuous, i.e., we have infinite number of states. When dealing with high-dimensional problems, the computational complexity of dynamic programming grows rapidly. To avoid the curse of the dimensions and the growth of the complex system dynamics, we propose to use deep Q-network as a computationally efficient method to solve the above POMDP problem. As an extension to the infinite time horizon, we provide stationary solutions for our reference belief selection problem with the following objective

$$L_T^{inf}(\pi) = \lim_{T \to \infty} \mathbb{E}_\pi \Big\{ \sum_{t=1}^{T} \gamma^t H(Z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, Y_t^{b_t}) \Big\}. \tag{6.18}$$

6.4.3 Deep Reinforcement Learning based Networked Calibration over Infinite Time Horizon

In this section, we consider the infinite time horizon case. Since the system statistics and dynamics are assumed to be time-invariant, we use $(s, a) \in \mathcal{S} \times \mathcal{A}$ to denote the belief state and action pair $(\tilde{\theta}_t, b_t)$, $l(s, a)$ and $P(s'|s, a)$ as the cost $l_t(\tilde{\theta}_t, b_t)$ and transition probability to state s' from the state-action pair (s, a) which corresponds to $P(\tilde{\theta}_{t+1}|\tilde{\theta}_t, b_t)$. Under

the infinite time horizon, the optimal Bellman equation of our reference belief selection problem becomes

$$V(s) = \min_{a\in\{1,2,\ldots,\bar{N}\}} l(s,a) + \gamma \mathbb{E}_{P(s'|s,a)}[V(s')]. \tag{6.19}$$

The optimal policy $\pi^* = (\pi, \pi, ...)$ is stationary and is defined by

$$\pi(s) = \arg \min_{a\in\{1,2,\ldots,\bar{N}\}} [l(s,a) + \gamma \mathbb{E}_{P(s'|s,a)} V^*(s')], \quad \forall s. \tag{6.20}$$

In the following, we provide a brief outline to solve the above problem using DQN. More details on this method can be found in [86].

Given the optimal value function $V^*(s)$ in (6.20), we define the optimal Q-function $Q(s,a)$ as the optimal total expected cost starting from state s with action a as follows

$$Q^*(s,a) = l(s,a) + \gamma \mathbb{E}_{P(s'|s,a)} V^*(s'). \tag{6.21}$$

Since $V^*(s) = \min_{a\in\{1,2,\ldots,\bar{N}\}} Q^*(s,a)$, we have

$$Q^*(s,a) = C(s,a) + \gamma \mathbb{E}_{P(s'|s,a)} \min_{b\in\{1,2,\ldots,\bar{N}\}} Q^*(s',b). \tag{6.22}$$

The traditional tabular Q-learning [72] can learn the optimal Q-function by utilizing the temporal difference between the new estimate and the old estimate. This is given by the following

$$Q_{n+1}(s,a) = (1-\beta) Q_n(s,a) + \beta [l(s,a) + \gamma \min_{b\in\{1,2,\ldots,\bar{N}\}} Q_n(s',b)], \tag{6.23}$$

where $\beta \in (0,1]$ denotes the learning rate. The learning rate can be kept constant during the learning process. However, traditional tabular Q-learning cannot deal with the continuous state space problem. To this end, we propose to implement DQN to learn the function representation of the optimal Q-function. This function representation can be then applied to each state over the whole continuous state space. In more detail, the Q-function $Q(s,a)$ is estimated by a function representation $Q(s,a,w)$, where w denotes the parameters of the deep neural network applied in DQN. To avoid the divergence of the algorithm, the DQN utilizes replay buffer D which is filled by samples of the transition tuples, i.e., (s,a,r,s'). At each time step, the DQN agent will execute an action a selected by ϵ-greedy policy. At training step i, the DQN agent randomly decides on a mini-batch D_i of the replay buffer. The objective of DQN is to minimize the following mean square error

$$L(w_i) = \mathbb{E}_{s,a,r,s'\sim D_i}[(l(s,a) + \gamma \min_b Q(s',b,w_i^-) - Q(s,a,w_i)^2], \tag{6.24}$$

where w^- represents the parameters of the target Q-network. It is updated every certain number of training steps, while $\mathbb{E}_{s,a,r,s'\sim D_i}$ indicates the mean square error is calculated

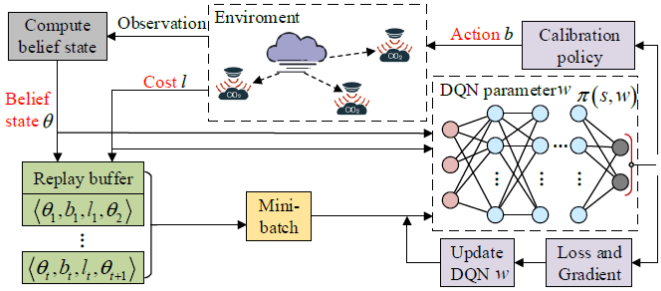


Figure 6.2: System model of DQN based networked calibration for CO_2 sensing system. A storage device is equipped at the cloud that acts as the replay buffer for DQN algorithm, while several computing nodes are also equipped at the cloud that enables computation of belief state, training of DQN, and the calibration policy design performed by the DQN agent.

by using the samples of transition tuples from the mini-batch D_i. The loss is further sent backward through the deep neural network to calculate the gradient with respect to the parameter w_i to update the weight w_i according to

$$w_{i+1} = w_i + \alpha \nabla_{w_i} L(w_i), \tag{6.25}$$

where $\alpha \in (0, 1]$ is the learning rate. After learning the optimal Q-function, the optimal policy is given by

$$\pi^*(s) = \arg \min_{a \in \{1,2,\ldots,\tilde{N}\}} \hat{Q}^*(s, a, w), \tag{6.26}$$

where $\hat{Q}^*(s, a, w)$ is the optimal approximated Q-function. Combining the DQN framework, the process of our sensor self-calibration is presented in Algorithm 6, and the design of our CO_2 sensing system is illustrated in Fig. 6.2.

Algorithm 6: DQN based reference belief function selection and self-calibration for NDIR gas sensor network

Input: HMM model parameters of each sensor $\{\rho, A, B\}$.

Initialization: DQN with initial Q-function $Q(s, a, w)$, learning rate α, experience replay buffer $\mathcal{S}$ with size S , mini-batch size N_D, and target model updating frequency F.

for *each episode* $e = 1, 2, ..., N^{epi}$ **do**

  Initialize the belief function of the true calibration parameter for each sensor as $P_n(x_1)$ according to the prior distribution parameter ρ.

  for *each training step* $t = 1, 2, ..., T$ **do**

    Each sensor observes its CO_2 measurement y_t^n, current signal i_t^n, temperature c_t, and calculates its belief function on true CO_2 $P_{i_t^n}(z_t|y_{1:t-1}^n, c_{1:t})$ using (6.3);

    Each sensor sends its belief function to the cloud;

    The cloud fusion center computes the approximated belief state $\tilde{\theta}_t$ according to the fusion rule proposed in Section V-C, and selects a reference sensor b_t based on the ϵ-greedy policy;

    The cloud fusion center executes action b_t by sending the reference belief function $Pi_t^{b_t}(z_t|y_{1:t-1}^{b_t}, c_{1:t})$ back to all sensors, and observes the cost $l_t(\tilde{\theta}_t, b_t)$;

    Each sensor updates belief function on true calibration parameter to $P_{new}(x_t^n|y_{1:t-1}^n, c_{1:t})$ according to the reference belief function, which evolves to $P(x_{t+1}^n|y_{1:t}^n, c_{1:t+1})$ according to equation (6.1) and (6.2);

    Each sensor calculates its updated belief function on true CO_2 $P_{i_{t+1}^n}(z_{t+1}|y_{1:t}^n, c_{1:t+1})$ using (6.3) and send it back to the cloud ;

    The cloud fusion center computes the approximated belief state $\tilde{\theta}_{t+1}$ according to the fusion rule proposed in Section V-C;

    Store the transition $(\tilde{\theta}_t, b_t, l_t, \tilde{\theta}_{t+1})$ into experience replay buffer $\mathcal{S}$, if $\mathcal{S}$ is full, remove the oldest entry in $\mathcal{S}$;

    If the number of the elements in replay buffer is larger than N_D, randomly select N_D samples for the mini-batch calculate loss using (6.24);

    Update the parameter w_t using (6.25);

    Update the target Q-network parameter w_t^- every F time steps using (6.25);

  end

  Save both w and target model parameter w^-.

end

Output: Reference sensor selection policy according to (6.26).

6.5 Simulation Results

In this section, we will evaluate the performance of the proposed networked calibration algorithms. We consider a network with the five sensors mentioned above (with computation ability). It is also equipped with a computing node that serves as the fusion center (for belief function fusion calibration) or the DQN agent (for deep Q-network based calibration), and the sensors can communicate with the computing node.

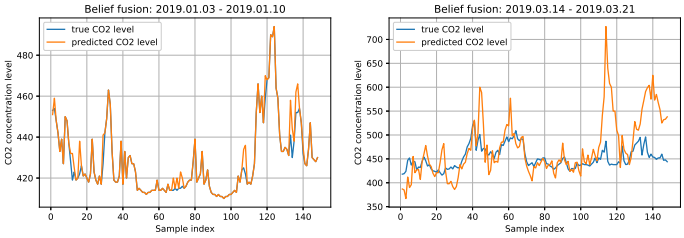
As a comparison, we first provide the performance of the deterministic HMM based single sensor calibration algorithm for five different sensors. The HMMs are trained by using the same dataset and same method as described in Section 5.3.1. As it is shown in Fig. 6.3, the HMM for each sensor performs well in the beginning, where the prediction is made only three days after the model has been trained. However, the temperature dependency of the drift behavior cannot be perfectly captured by the HMM due to the remaining model errors. Thus, the aggregation of the imperfectness will lead to the degradation of the HMM performance over time, because more and more drift behaviors that have not been experienced during training phase will appear. As we can see from the figures, the performance of the HMM for Sensors 1 and 2 start to degrade three months after the models have been trained, and the performance of the HMM for Sensor 4 remains very good after six months.

6.5.1 Numerical Results for Belief Function Fusion based Calibration

We first provide the numerical results for our Wasserstein distance based weighted average belief function fusion approach. We fuse the belief functions of all five sensors in both period 2019.01.03 - 2019.01.10 and 2019.03.21 - 2019.03.27. The predicted CO_2 level is given by the MAP estimation performed based on the fusion result. As it can be observed from Fig. 6.3, the belief function of the true CO_2 level provided by Sensors 3, 4, and 5 are consistent and reliable during the period 2019.01.03 - 2019.01.10. In this case, our proposed weighted average based belief function fusion approach will assign a high weight to the belief functions from Sensors 3, 4, and 5, thus will lead to a accurate and correct fused belief function. On the other hand, during the period 2019.03.21 - 2019.03.27, only the belief function of Sensor 4 provides the correct evidence. In this case, higher weight will be assigned to the other belief functions which provide incorrect evidence, and the performance of our proposed belief function fusion approach degrades rapidly. The above results are illustrated in Fig. 6.4.

6.5.1.1 Numerical Results for Deep Q-network based Calibration for Sensor Network

To test the performance our proposed deep reinforcement learning algorithm, we have to build a simulation environment that can interact with the DQN agent. The code for building this simulation environment is provided at [87]. On receiving an action from the DQN agent, i.e., decision on the reference sensor, the environment can simulate the evolution of the belief state according to the process as described in Section 6.4.1 and



(a) (b)

Figure 6.4: Predicted true CO_2 level using belief function fusion approach. (a) Prediction result from 2019.01.03 to 2019.01.10. (b) Prediction result from 2019.03.14 to 2019.03.21.

Table 6.1: Setting of parameters for DQN environment

Parameters	Value
Number of sensors	5
Number of DQN agent	1
Temperature (°C)	-5:0.1:24
True CO_2 level (ppm)	250:1:750

generates a reward which is sent back to the DQN agent. Since our algorithm is built upon a predefined probabilistic space, we restrict the belief functions of the true calibration parameter and true CO_2 levels from different sensors to be defined on the same support set. In more details, the true calibration parameter for each sensor can only take certain integer values within the same range. The settings of the true CO_2 level and the environmental temperature are presented in Table I, while the CO_2 measurements and the sensor current of the detector for different sensors are again quantized to certain integer values within certain ranges.

The selection of the network parameters decide the learning convergence speed and efficiency. In this thesis, we specifically illustrate the impact of different values of learning rate on the performance of our DQN algorithm. Fig. 6.5 shows the average calibration cost versus training episodes under different learning rates, i.e., $\alpha = \{0.1, 0.01, 0.001\}$. It can be observed that it requires longer convergence time when the learning rate decreases. However, fast convergence would probably make the agent fall into a local optimum and miss the better strategy. Based on the above results, we set our learning rate equals 0.001. For the DQN agent, we use a 5-layer perceptron where we have three connected hidden layers containing 500, 300 and 200 neurons respectively. The learning rate is set to $\alpha = 0.001$, the discount factor is set to $\gamma = 0.99$, and the exploration rate ϵ is initially set as 0.9 and decays to 0.001 at rate 0.995. The DQN model is trained for 2000 epochs under a experience replay with memory size 50000. The mini-batch size is set as 64 and the target Q network is updated every 500 time steps.

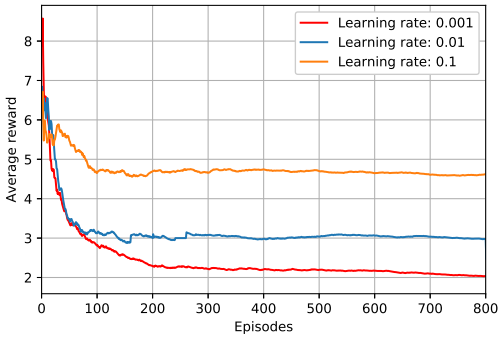
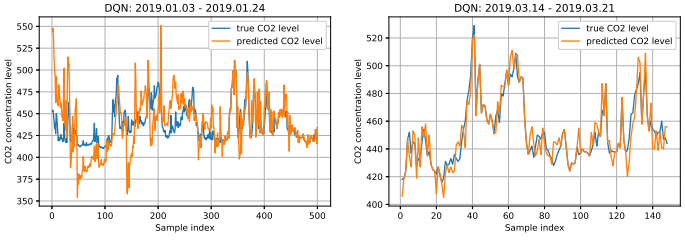


Figure 6.5: Average cost performance versus episodes under different learning rates.



(a) (b)

Figure 6.6: Predicted true CO_2 level using DQN based calibration. (a) Prediction result from 2019.01.03 to 2019.01.10. (b) Prediction result from 2019.03.14 to 2019.03.21.

With the trained DQN agent, we test our DQN based calibration algorithm during period 2019.01.03 - 2019.03.21. The performance is illustrated in Fig. 6.6. As it can be seen from the Fig. 6.6(a), our proposed DQN based calibration algorithm performs worse than our modified belief function fusion approach in the beginning of the testing period. This is the case since the DQN based algorithm provides a long-term calibration strategy instead of calibrating the sensors according the best instantaneous choice. When most of the sensors have a well-performing single sensor HMM model, the modified belief function fusion approach can thus outperform the DQN based calibration algorithm. As it can be also observed, the performance illustrated in Fig. 6.6(a) improves over time and finally end up with a high-accuracy performance as illustrated in Fig. 6.6(b). In comparison, the performance of modified belief function fusion degrades with the time as shown in Fig. 6.4(b). This means that even with some drawbacks under some specific cases, the DQN based calibration algorithm has a stable and good performance over the long term.

6.6 Summary

In this chapter, we developed two networked calibration algorithms for the NDIR CO_2 sensing system. The proposed weighted average belief function fusion approach can be used to deal with the case when belief functions provided by different sensors highly conflict with each other. The corresponding fusion result can be used as an approximation of the joint statistics when it is difficult to form a joint statistics among different sensors. As a consequence, the proposed POMDP based multi-sensor sequential self-calibration approach utilizes the fusion result as an approximation of the joint statistics, and achieves more stable long-term calibration calibration performance. Based on the numerical experiments, the weighted average belief function fusion approach is shown to have higher calibration accuracy than the POMDP based approach when the majority of the sensors beliefs provide reliable and correct evidence. On the other hand, the POMDP based approach can achieve mild but stable calibration accuracy over a relatively long period compared to the weighted average belief function fusion approach.

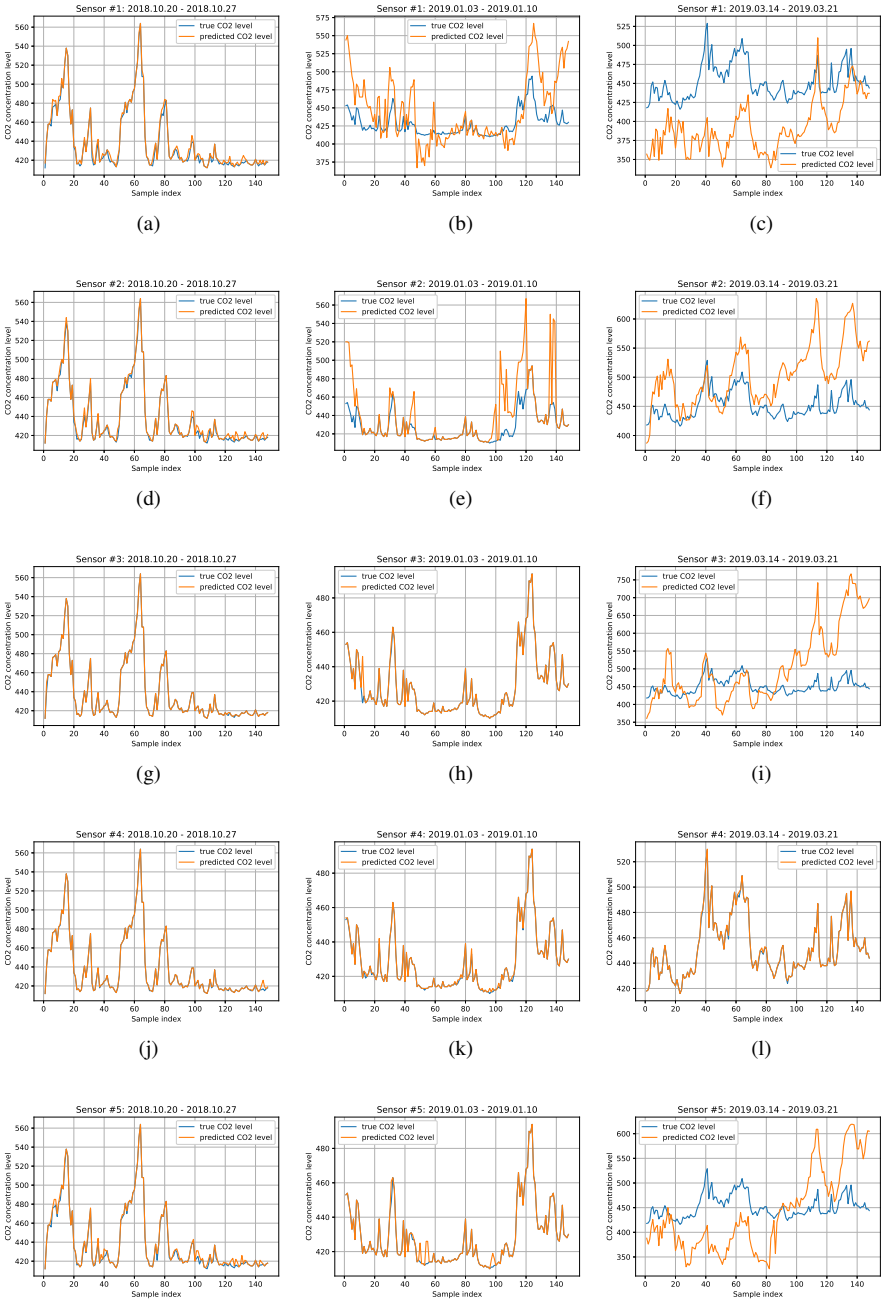


Figure 6.3: Predicted true CO_2 level in different periods from 2018.10.20 to 2019.03.21 using HMM based framework. (a)-(c) Prediction results for Sensor 1. (d)-(f) Prediction results for Sensor 2. (g)-(i) Prediction results for Sensor 3. (j)-(l) Prediction results for Sensor 4. (m)-(o) Prediction results for Sensor 5.

Chapter 7

Conclusions and Future Work

In this thesis, different aspects of intelligent system design are studied. Particularly, different design frameworks have been proposed to address the privacy-cost trade-off problem of smart grid consumers as well as the data-driven gas sensor calibration problem. We summarize our key findings and the potential future works regarding these two topics separately in the following.

7.1 Privacy-cost Trade-off for Smart Grid Consumers

To solve this problem, different privacy-preserving and cost-efficient energy manage strategy design frameworks have been proposed for either single smart grid consumer or multiple smart grid consumers. Particularly, in the presence of an ES, a POMDP based stochastic control approach is proposed to solve the privacy-cost trade-off problem for single smart grid consumer. Accordingly, an energy management that optimally trade off KL-divergence (privacy measure) and the expected cost-savings (cost measure) can be derived using the Bellman dynamic programming. Meanwhile, in the presence of RES, the privacy-cost trade-off problem for multiple smart grid consumers has been studied. For this part of work, we have shown the privacy-preserving problem against the FHMM-based adversarial NILM can be formulated as a power request design that minimizes the joint log-likelihood of the power request sequence and the operating state sequence. With the dynamic pricing model that is determined by all users' aggregated power request, the privacy-preserving and cost-efficient power request design problem becomes a non-cooperative game. Accordingly, a unique NE energy management strategy that trade off each user's privacy measure and the cost savings is designed to make sure that all users are satisfied. The above findings answer research problem **RP1**.

Furthermore, to address the research question **RP2**, different computationally more efficient approaches have been proposed to solve the above problems. Based on the POMDP formulation for single smart grid consumer's privacy-cost trade-off, we propose a sub-optimal but computationally efficient LARQL approach that also works for an infinite time

horizon. With the identified feature vector, the linear function approximated Q-function can be efficiently learned and therefore leads to a practical online energy management design approach. Another approach to reduce the strategy design complexity is to assume an i.i.d. energy demand, which allows further analysis, in particular the derivation of a steady state strategy. Moreover, we provide sufficient conditions to achieve perfect privacy. Regarding the non-cooperative game formulation for multiple smart grid consumers' privacy-cost trade-off, a distributed relaxation algorithm is further proposed to reduce the complexity of solving the NE. Moreover, the distributed relaxation algorithm is shown to lead to a unique NE.

To reduce the complexity of the POMDP based energy management design, we adopted the natural choice of Q-learning as algorithm that follows the value function based approach under the proposed MDP framework. Since there also exists other efficient reinforcement learning techniques to deal with the continuous state-action space MDP problem, e.g., policy gradient algorithms, an interesting extension would be to implement such algorithms and assess which approach can provide a better solution to our proposed privacy-cost trade-off problem. On the other hand, when designing the NE energy management strategy for multiple consumers under the non-cooperative game framework, we assume the future power consumption over a certain of the consumers is always predictable. Thus, a potential extension would be designing an online energy management strategy when the future power consumption is unpredictable, and the consumers would admit a generalized NE in this case.

7.2 Data-driven Gas Sensor Calibration

For this part of work, we developed several self-calibration algorithms for low-cost gas sensors. The measurement errors of the sensors are mainly caused by the remaining model errors and can be fully described by the drift of the calibration parameter. This leads to our first formulation of a statistical inference problem on the true calibration parameter under the HMM framework, which is a stochastic model that jointly builds on observed sensor measurements sequence, observed environmental factor sequences, and the sequence of true calibration parameter. To better track the time-varying drift process of the sensor, a time-adaptive EM learning framework is proposed to efficiently update the HMM parameters. Compared to the approach which utilizes the deterministic HMM to predict the true calibration parameter, our proposed framework significantly improves the prediction accuracy over the whole lifetime of the sensors. Moreover, our designed framework can be seen as a transfer learning approach that (i) always achieves a fast convergence rate with a relatively small training data set and (ii) allows to discard previous measurements and thereby systematically prevents a big data storage problem due to growing stored data for training. This shows the great value of our designed framework regarding data efficiency, computational efficiency, and time efficiency. The above findings jointly addressed the research problems **RP3** and **RP4**.

Also, as an answer to the research problem **RP4**, the joint calibration over the whole gas sensing network is further studied. A belief function fusion framework is proposed to

solve the multi-sensor data-fusion problem. Furthermore, a weighted average belief function fusion approach is proposed as an improvement that can deal with the case when belief functions provided by different sensors highly conflict with each other. The corresponding fusion result can be used as an approximation of the joint statistics when it is difficult to form a joint statistics among different sensors. As a consequence, by utilizing the fusion result as an approximation of the joint statistics, a POMDP based multi-sensor sequential self-calibration approach is next proposed, which achieves more stable long-term calibration calibration performance.

The performance of the learning algorithm heavily depends on the quality and size of the training data. Due to packet losses, the training data suffers from missing data points. Training data contains outliers which need to be treated efficiently during learning. Some measurement outliers have a physical explanation which should not be learned for complexity reason. Thus, more robust learning procedures which are data-efficient and complexity-efficient need to be further explored, i.e., studying data augmentation techniques to enhance the quality of training data for more efficient learning of the model, or study the impact of ensemble learning techniques such as bootstrap to exploit training data more efficiently.

Appendix A

A.1 Proof of Proposition 1

To prove this proposition, we need to show $C_T(\pi, \lambda)$ is equal to $C_T(f^{'}, \lambda)$ under transition from strategy $f^{'}$ to policy π. Thus, we need to further show that the probability terms in these two objective functions are equal. Since the proofs for the other probability terms are similar, we only prove $P^{f^{'}}_{Y^T,P^T|h_i} = P^{\pi}_{Y^T,P^T|h_i}$ here. After expanding $P^{\pi}_{Y^T,P^T|h_i}$ according to the policy π, we obtain

$$\begin{aligned} P^{\pi}_{Y^T,P^T|h_i}(y^T, p^T) = \sum_{x^T,s^T} & P_{X_1,S_1,P_1|h_i}(x_1, s_1, p_1) \times a_1(y_1|x_1, s_1, p_1, h_i) \\ & \prod_{t=1}^{T-1} [P(p_{t+1}|p_t)P(x_{t+1}|x_t, h_i)\mathcal{I}_{s_{t+1}}(y_t + s_t - x_t) \\ & \times a_{t+1}(y_{t+1}|x_{t+1}, s_{t+1}, p_{t+1})]. \end{aligned} \tag{A.1}$$

By applying the equivalence between $f^{'}_t$ and (π_t, a_t), which is derived above, we get $P^{f^{'}_t}(y_{t+1}|x_{t+1}, s_{t+1}, p^{t+1}, y^t) = a_t(y_{t+1}|x_{t+1}, s_{t+1}, p_{t+1})$. Also, we have $P^{f^{'}_t}(y_1|x_1, s_1, p_1) = a_1(y_1|x_1, s_1, p_1)$. Thus, $P^{f^{'}}_{Y^T,P^T|h_i} = P^{\pi}_{Y^T,P^T|h_i}$ holds.

A.2 Proof of Proposition 3

For reason of simplicity, we provide the derivation for the features-action representation of the first term in (3.38), and the others can be derived in a similar way.

Ignoring λ, the first term in (3.38) can be further decomposed into the following:

$$\begin{aligned}
&\sum_{p_t,y_t} P^{\pi_t}(y_t,p_t|q_{t-1},h_0)\log P^{\pi_t}(y_t,p_t|q_{t-1},h_0)\\
&\quad=\sum_{p_t,y_t}\Big(\sum_{x_t,s_t}\theta^0(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)\\
&\quad\times\log\sum_{x_t,s_t}\theta^0(x_t,s_t,p_t)a_t(y_t|x_t,s_t,p_t)\Big)\\
&\quad\overset{(a)}{=}\sum_i^L\sum_j^K|a_i^j||\theta_i^0|\cos\phi_i^j(\log|a_i^j|+\log|\theta_i^0|+\log\cos\phi_i^j)\\
&\quad=\sum_i^L|\theta_i^0|\sum_j^K|a_i^j|\cos\phi_i^j(\log|a_i^j|+\log\cos\phi_i^j)\\
&\quad+\sum_i^L|\theta_i^0|\log|\theta_i^0|\sum_j^K|a_i^j|\cos\phi_i^j\\
&\quad\overset{(b)}{=}\vec{A}_1\cdot\vec{B}_1+\vec{A}_2\cdot\vec{B}_2\\
&\quad=|\vec{A}_1||\vec{B}_1|\cos\left\langle\vec{A}_1,\vec{B}_1\right\rangle+|\vec{A}_2||\vec{B}_2|\cos\left\langle\vec{A}_2,\vec{B}_2\right\rangle,
\end{aligned}\tag{A.2}$$

where $\cdot$ denotes the inner product between two vectors in the Euclidean space. The equality (a) in (A.2) follows from considering sum over all possible (y_t,p_t) and the inner product between vector θ_i^0 and a_i^j. Likewise, the equality (b) is an inner product where the i-th element of $\vec{A}_1,\vec{A}_2,\vec{B}_1,\vec{B}_2$ are defined by the following:

$$\begin{aligned}
A_{1i}&=\sum_j^K|a_i^j|\cos\phi_i^j(\log|a_i^j|+\log\cos\phi_i^j),\quad B_{1i}=|\theta_i^0|,\\
A_{2i}&=\sum_j^K|a_i^j|\cos\phi_i^j,\quad B_{2i}=|\theta_i^0|\log|\theta_i^0|.
\end{aligned}\tag{A.3}$$

Similarly, the second term in (3.38) can be decomposed by:

$$\begin{aligned}
&\sum_{p_t,y_t}P^{\pi_t}(y_t,p_t|q_{t-1},h_0)\log P^{\pi_t}(y_t,p_t|q_{t-1},h_1)\\
&\quad=\vec{A}_3\cdot\vec{B}_1+\vec{A}_2\cdot\vec{B}_3\\
&\quad=|\vec{A}_3||\vec{B}_1|\cos\left\langle\vec{A}_3,\vec{B}_1\right\rangle+|\vec{A}_2||\vec{B}_3|\cos\left\langle\vec{A}_2,\vec{B}_3\right\rangle,
\end{aligned}\tag{A.4}$$

where the i-th element of $\vec{A}_3,\vec{B}_3$ are defined by the following:

$$A_{3i}=\sum_j^K|a_i^j|\cos\phi_i^j(\log|a_i^j|+\log\cos\psi_i^j),B_{3i}=|\theta_i^0|\log|\theta_i^1|.\tag{A.5}$$

Thus, the decomposition of the first two terms identifies the features $f_2((\theta^0,\theta^1)) = |\vec{B}_1| = |\theta^0|$, $f_4((\theta^0,\theta^1)) = |\vec{B}_2| = \sqrt{\sum_i^L (|\theta_i^0|\log|\theta_i^0|)^2}$, and $f_5((\theta^0,\theta^1)) = |\vec{B}_3| = \sqrt{\sum_i^L (|\theta_i^0|\log|\theta_i^1|)^2}$, and we can conclude these features will be highly relevant to the value of KL-divergence term.

Meanwhile, by applying the same method to decompose last two terms in (3.38), i.e., the cost-saving term, $f_2((\theta^0,\theta^1)) = |\theta^0|$ and $f_3((\theta^0,\theta^1)) = |\theta^1|$ are identified as the features that will be highly relevant to the value of the cost-saving term. Besides, feature "1" is selected to add an offset to the approximated function and compensate the errors.In this case, summarizing the above analysis leads to the feature selection scheme in Proposition 3.

A.3 Proof of Lemma 2

Since $\xi_t(w_t)$ and $\gamma_t(s_t)$ are linearly related to each other by $\xi_t(w_t) = \sum_{(x_t,s_t)\in\mathcal{D}(w_t)} P_X(x_t)\gamma_t(s_t)$, it is then sufficient to show $\xi_t(w_t) = \xi'_t(w_t),\ \forall w_t \in \mathcal{W}$ at each time step.

$$\xi_{t+1}(w_{t+1}) = \frac{\sum_{(x_{t+1},s_{t+1})\in\mathcal{D}(w_{t+1})} \sum_{x_t,s_t} \theta_t(x_t,s_t)a_t(y_t|x_t,s_t)P_X^0(x_{t+1})\mathcal{I}_{s_{t+1}}\{y_t+s_t-x_t\}}{\sum_{x_t,s_t} a_t(y_t|x_t,s_t)\theta_t(x_t,s_t)} \tag{A.6}$$

$$\xi'_{t+1}(w_{t+1}) = \frac{\sum_{(x_{t+1},s_{t+1})\in\mathcal{D}(w_{t+1})} \sum_{w_t} \sum_{(x_t,s_t)\in\mathcal{D}(w_t)} \theta_t(x_t,s_t)a_t(y_t|x_t,s_t)P_X^0(x_{t+1})\mathcal{I}_{s_{t+1}}\{y_t+s_t-x_t\}}{\sum_{w_t}\sum_{(x_t,s_t)\in\mathcal{D}(w_t)} a_t(y_t|x_t,s_t)\theta_t(x_t,s_t)}. \tag{A.7}$$

In the following, we use the induction method to prove that $\xi_t(w_t) = \xi'_t(w_t),\ \forall w_t \in \mathcal{W}$ at each time step. For $t = 1$, the initial distributions $\xi_1(w)$ and $\xi'_1(w)$ are identical since they do not depend on the actions a_t or b_t. Then, for any $t > 1$, given $\xi_t(w_t) = \xi'_t(w_t),\ \forall w_t \in \mathcal{W}$, we need to show $\xi_{t+1}(w_{t+1}) = \xi'_{t+1}(w_{t+1}),\ \forall w_{t+1} \in \mathcal{W}$ holds.

Knowing that $\xi_{t+1}(w_{t+1}) = \sum_{(x_{t+1},s_{t+1})\in\mathcal{D}(w_{t+1})} \theta_t(x_t,s_t)$, we can derive the expression for $\xi_{t+1}(w_{t+1})$ as shown in (A.6). Meanwhile, $\xi'_{t+1}(w+1)$ can be expressed by Equation (3.47). According to Equation (3.45), we have

$b_t(y_t|w_t)\xi_t(w_t) = \sum_{(x_t,s_t)\in\mathcal{D}(w_t)} a_t(y_t|x_t,s_t)\theta_t(x_t,s_t)$. Since $\xi_t(w_t) = \xi'_t(w_t),\ \forall w_t \in \mathcal{W}$ holds according to our assumption, we can substitute $b_t(y_t|w_t)\xi'_t(w_t)$ by $\sum_{(x_t,s_t)\in\mathcal{D}(w_t)} a_t(y_t|x_t,s_t)\theta_t(x_t,s_t)$ in Equation (3.47), which then leads to the expression as shown in (A.7). On noticing that the operator $\sum_{w_t}\sum_{(x_t,s_t)\in\mathcal{D}(w_t)}$ is equivalent to $\sum_{(x_t,s_t)}$, we can conclude $\xi_{t+1}(w_{t+1}) = \xi'_{t+1}(w_{t+1})$. Thus, according to the principle of induction, there is $\xi_t(w_t) = \xi'_t(w_t),\ \forall w_t \in \mathcal{W}$ at each time step.

A.4 Proof of Theorem 2

At first, $\hat{b}_t$ can be easily verified to be a feasible action which belongs to the set $\mathcal{B}_t$ by the following:

$$\begin{aligned} &P_Y(y)\times\gamma'_t(y+w) \\ &\quad = P_Y(y)\times\gamma_t(y+w) \\ &\quad = P^{\pi}(S_t = y+w, Y_t = y|Y^{t-1} = y^{t-1}, h_0) \\ &\quad = P^{\pi}(W_t = w, Y_t = y|Y^{t-1} = y^{t-1}, h_0). \end{aligned} \tag{A.8}$$

We next show the sufficiency. Since γ'_t and ξ'_t are easily shown to be equivalent, it is sufficient to check if $\gamma'_t = \gamma'_1$ for all t. For a time invariant policy, it is then sufficient to show $\gamma'_2 = \gamma'_1$. Consider a realization s of S_2, y of Y_1, and $w = s - y$. If $y \in \overline{\mathcal{Y}}(w)$ holds, we have:

$$\begin{aligned} P^{\hat{f}}(S_2 = s, Y_1 = y) &= P^{\hat{f}}(W_1 = s-y, Y_1 = y) \\ &= \xi'_1(s-y)\hat{b}_1(y|s-y) \\ &= P_Y(y)\gamma'_1(s). \end{aligned} \tag{A.9}$$

Marginalize over all possible s, we can get $P_{Y_1}(y) = Q_Y(y)$. Divide both sides by $Q_Y(y)$, it follows that $\gamma'_2(s) = P^{\hat{f}}(S_2 = s|Y_1 = y) = \gamma'_1(s)$. Regarding the proof of necessity, we divide it into two parts, where we first show that under the time-invariant policy which leads to the steady state, $P_{Y_t|Y^{t-1}=y^{t-1}},\ \forall y^{t-1}$ remains identical.The joint distribution $P^{\hat{f}}_{W_t,Y_t|Y^{t-1},h_0}$ can be decomposed by the following:

$$\begin{aligned} P^{\hat{f}}_{W_t,Y_t|Y^{t-1},h_0} &= P^{\hat{f}}_{W_t|Y^{t-1},h_0} \times P^{\hat{f}}_{Y_t|W_t,Y^{t-1},h_0} \\ &= \xi'_t(w)b_t(y|w) \\ &\overset{(a)}{=} \xi'_1(w)b_1(y|w) \\ &= P^{\hat{f}}_{W_1,Y_1|h_0}, \end{aligned} \tag{A.10}$$

where (a) holds due to the stationarity of states ξ'_t. Marginalizing over W, we can get $P^{\hat{f}}_{Y_t|Y^{t-1},h_0} = P^{\hat{f}}_{Y_1|h_0},\ \forall y^{t-1}$, which proves the above lemma.

Given the conclusion $P^{\hat{f}}_{Y_t|Y^{t-1},h_0} = P^{\hat{f}}_{Y_1|h_0},\ \forall y^{t-1}$ remains indentical, we further show that the structure of the strategy in (3.50) should always be satisfied with $Q_Y \overset{(\Delta)}{=} P_{Y_t|Y^{t-1}=y^{t-1}},\ \forall y^{t-1}$. Considering a realization s of S_2, y of Y_1, and $w = s - y$. If $y \in \overline{\mathcal{Y}}(w)$ holds, we have,

$$\begin{aligned}\mathbb{P}^{\hat{f}}(S_2 = s, Y_1 = y) &= \mathbb{P}^{\hat{f}}(W_1 = s - y, Y_1 = y) \\ &= \xi_1'(s-y) b_1(y|s-y)\end{aligned} \tag{A.11}$$

Since $\gamma_2'(s) = \gamma_1'(s)$ holds due to the stationarity of the states, divide both sides by $P^{\hat{f}}_{Y_1}(y)$, we can get,

$$\begin{aligned}&\mathbb{P}^{\hat{f}}(S_2 = s|Y_1 = y) = \gamma_1'(s) = \frac{\xi_1'(s-y) b_1(y|s-y)}{P^{\hat{f}}_{Y_1}(y)} \\ \implies\ & b_1(y|w) = P^{\hat{f}}_{Y_1}(y) \frac{\gamma_1'(y+w)}{\xi_1'(w)}\end{aligned} \tag{A.12}$$

Since γ_t' and ξ_t' remain identical over whole time horizon, and there is $P^{\hat{f}}_{Y_t|Y^{t-1}} = P^{\hat{f}}_{Y_1}$, the above equation implies that the action b_t satisfies the structure in (3.50) with $Q_Y \overset{(\Delta)}{=} P_{Y_t|Y^{t-1}=y^{t-1}},\ \forall y^{t-1}$ over whole time horizon.

A.5 Proof of Theorem 3

Note that each step in the multi-step non-cooperative game is independent of the other steps. Therefore, the non-cooperative game can be divided into independent single-step games. The per-step payoff for User i at time step t can be rewritten as

$$\begin{aligned}U_{i,t}(z_{i,t}, \boldsymbol{z}_{-i,t}) = &(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha) z_{i,t}^2 + (\delta_{i,t} - \eta\alpha\bar{z}_{-i,t}) z_{i,t} \\ &+ \delta_{i,t}^{'} + \eta\alpha y_{i,t}\bar{z}_{-i,t},\end{aligned} \tag{A.13}$$

where we introduce the following notations:

$$\delta_{i,t} = \eta(\alpha y_{i,t} - \rho_{base}) - \frac{\nu_i \mu_{i,t}}{\sigma_{i,t}^2}, \tag{A.14}$$

$$\delta_{i,t}^{'} = \eta\rho_{base} y_{i,t} + \frac{\nu_i \mu_{i,t}^2}{2\sigma_{i,t}^2}, \tag{A.15}$$

$$\bar{z}_{-i,t} = \sum_{k \neq i} z_{k,t}. \tag{A.16}$$

One can easily see that the per-step payoff for each user in (A.13) is a quadratic function with respect to $z_{i,t}$ given $\bar{z}_{-i,t}$ induced by the other users. Given the condition $\alpha > \frac{\nu_i}{2\eta\sigma_{i,t}^2}$

for all i, t, i.e., $\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha < 0$. Thus the per-step payoff function for each user at each time step is concave, and all the users will engage in an M-person concave game as defined in [88] at each time step. Notice that the feasible power request set $\overline{\mathcal{Z}}(y_{i,t}, R_{i,t})$ for User i at time step t is a closed convex set. It follows from [88, Th. 1] that the non-cooperative game admits at least a pure-strategy NE at each time step, which further guarantees the existence of pure-strategy NE over the T-time horizon.

According to [88, Th. 2], the non-cooperative game admits a *unique* pure-strategy NE if the per-step payoff functions of all users satisfy the diagonal strict concavity condition at each time step. Let $\boldsymbol{z}_t = (z_{1,t}, z_{2,t}, \ldots, z_{M,t})^{'} \in \mathcal{Z}_t = \overline{\mathcal{Z}}(y_{1,t}, R_{1,t}) \times \overline{\mathcal{Z}}(y_{2,t}, R_{2,t}) \times \cdots \times \overline{\mathcal{Z}}(y_{M,t}, R_{M,t})$. The diagonal strict concavity condition is satisfied if there exists a non-negative vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_M)^{'} \in \mathbb{R}^{M\times 1}$ such that for any $\boldsymbol{z}_t^0$, $\boldsymbol{z}_t^1 \in \mathcal{Z}_t$,

$$(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0)^{'} \boldsymbol{g}(\boldsymbol{z}_t^0, \boldsymbol{r}) + (\boldsymbol{z}_t^0 - \boldsymbol{z}_t^1)^{'} \boldsymbol{g}(\boldsymbol{z}_t^1, \boldsymbol{r}) > 0, \tag{A.17}$$

where the pseudogradient vector is defined as

$$\begin{aligned} &\boldsymbol{g}(\boldsymbol{z}_t, \boldsymbol{r}) \\ &= \left(r_1 \frac{\partial U_{1,t}(z_{1,t}, \boldsymbol{z}_{-1,t})}{\partial z_{1,t}}, \ldots, r_M \frac{\partial U_{M,t}(z_{M,t}, \boldsymbol{z}_{-M,t})}{\partial z_{M,t}} \right)^{'}. \end{aligned} \tag{A.18}$$

The first-order partial derivative of $U_{i,t}(z_{i,t}, \boldsymbol{z}_{-i,t})$ with respect to $z_{i,t}$ is

$$\frac{\partial U_{i,t}(z_{i,t}, \boldsymbol{z}_{-i,t})}{\partial z_{i,t}} = (\frac{\nu_i}{\sigma_{i,t}^2} - 2\eta\alpha) z_{i,t} + \delta_{i,t} - \eta\alpha \bar{z}_{-i,t}. \tag{A.19}$$

Let $\boldsymbol{R}$ be an $M \times M$ diagonal matrix with the i-th diagonal element r_i, $\boldsymbol{D}$ be an $M \times M$ diagonal matrix with the i-th diagonal element $\frac{\nu_i}{\sigma_{i,t}^2}$, $\boldsymbol{I}$ be an $M \times M$ identity matrix, $\boldsymbol{J}$ be an $M \times M$ matrix with all elements 1, $\boldsymbol{K} = \boldsymbol{D} - \eta\alpha(\boldsymbol{I} + \boldsymbol{J})$, and $\boldsymbol{c} = (\delta_{1,t}, \delta_{2,t}, \ldots, \delta_{M,t})^{'}$. Then, $\boldsymbol{g}(\boldsymbol{z}_t, \boldsymbol{r}) = \boldsymbol{R}(\boldsymbol{K}\boldsymbol{z}_t + \boldsymbol{c})$. Substituting it into (A.17), we can obtain

$$\begin{aligned} &(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0)^{'} \boldsymbol{g}(\boldsymbol{z}_t^0, \boldsymbol{r}) + (\boldsymbol{z}_t^0 - \boldsymbol{z}_t^1)^{'} \boldsymbol{g}(\boldsymbol{z}_t^1, \boldsymbol{r}) \\ &= (\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0)^{'} \boldsymbol{R}(\boldsymbol{K}\boldsymbol{z}_t^0 + \boldsymbol{c}) + (\boldsymbol{z}_t^0 - \boldsymbol{z}_t^1)^{'} \boldsymbol{R}(\boldsymbol{K}\boldsymbol{z}_t^1 + \boldsymbol{c}) \\ &= -(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0)^{'} \boldsymbol{R}\boldsymbol{K}(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0). \end{aligned} \tag{A.20}$$

If $\boldsymbol{R}\boldsymbol{K}$ is a negative definite matrix, it follows that $-(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0)^{'} \boldsymbol{R}\boldsymbol{K}(\boldsymbol{z}_t^1 - \boldsymbol{z}_t^0) > 0$ and furthermore the inequality (A.17) always holds. Let $\boldsymbol{r}$ be a positive vector, i.e., $r_i > 0$ for all $1 \le i \le M$. Since $\boldsymbol{R}\boldsymbol{K}$ is a diagonal matrix with eigenvalues $\{\frac{r_i \nu_i}{\sigma_{i,t}^2} - 2\eta\alpha r_i\}_{i=1}^M$, $\boldsymbol{R}\boldsymbol{K}$ is a negative definite matrix if $\frac{\nu_i}{\sigma_{i,t}^2} - 2\eta\alpha < 0$ for all $1 \le i \le M$, i.e., all eigenvalues of $\boldsymbol{R}\boldsymbol{K}$ are negative. Thus, the i-th single-step game admits a unique pure-strategy NE if $\alpha > \frac{\nu_i}{2\eta\sigma_{i,t}^2}$, $\forall 1 \le i \le M$. Since all steps in the game are mutually independent, the non-cooperative game admits a unique pure-strategy NE if $\alpha > \frac{\nu_i}{2\eta\sigma_{i,t}^2}$, $\forall 1 \le i \le M$, $1 \le t \le T$.

A.6 Proof of Theorem 4

To prove the convergence of Algorithm 4, we need to show that the conditions (1)-(6) of [83, Th. 3.1] hold. Based on the previous definitions and derivations, the joint action set $\mathcal{Z}_t$ is a convex compact set, and the best response function defined in (4.19) is single-valued and continuous on $\mathcal{Z}_t$, which verify conditions (1) and (3). It is also easy to verify the step size ξ_s satisfies the following conditions: (i) $\xi_s > 0$; (ii) $\sum_{s=0}^{\infty} \xi_s \to \infty$; (iii) $\xi_s \to 0$ as $s \to \infty$. Thus, condition (6) also holds. In the following, we provide the detailed proof to show show that conditions (2), (4), and (5) hold for our problem. First, with slight abuse of notation, we provide the following definition.

Definition 6. *Let $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is a convex closed subset of the Euclidean space $\mathbb{R}^M$. The function $\psi(\cdot, \cdot)$ is referred to as weakly convex-concave if the following conditions hold for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$ and $0 \leq \gamma \leq 1$:*

$$\begin{aligned} &\gamma\psi(\boldsymbol{x}, \boldsymbol{y}) + (1-\gamma)\psi(\boldsymbol{z}, \boldsymbol{y}) \\ &\quad \geq \psi(\gamma\boldsymbol{x} + (1-\gamma)\boldsymbol{z}, \boldsymbol{y}) + \gamma(1-\gamma) r_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{z}), \end{aligned} \tag{A.21}$$

$$\begin{aligned} &\gamma\psi(\boldsymbol{x}, \boldsymbol{y}) + (1-\gamma)\psi(\boldsymbol{x}, \boldsymbol{z}) \\ &\quad \leq \psi(\boldsymbol{x}, \gamma\boldsymbol{y} + (1-\gamma)\boldsymbol{z}) + \gamma(1-\gamma) \mu_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{z}), \end{aligned} \tag{A.22}$$

where $r_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{z})$ and $\mu_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{z})$ are the residual terms and are defined with respect to a norm as

$$\begin{aligned} &\frac{r_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{z})}{\|\boldsymbol{x} - \boldsymbol{z}\|} \to 0, \ as \ \|\boldsymbol{x} - \boldsymbol{z}\| \to 0, \ \forall \boldsymbol{y} \in \mathcal{X}, \\ &\frac{\mu_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{z})}{\|\boldsymbol{y} - \boldsymbol{z}\|} \to 0, \ as \ \|\boldsymbol{y} - \boldsymbol{z}\| \to 0, \ \forall \boldsymbol{x} \in \mathcal{X}. \end{aligned} \tag{A.23}$$

Now, we show condition (2) holds, i.e., the Nikaido-Isoda function in our problem is weakly convex-concave.

Lemma 7. *For the considered non-cooperative game, the Nikaido-Isoda function $\psi(\boldsymbol{z}_t, \boldsymbol{z}_t')$ defined in* (4.18) *is a weakly convex-concave function.*

Proof. We first check the Nikaido-Isoda function $\psi(\boldsymbol{z}_t, \boldsymbol{z}_t')$ is weakly convex on the joint power request set $\mathcal{Z}_t$ with respect to the first argument. According to Definition 6, we need to find a residual term $r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{w}_t)$ such that the following condition holds for all $\boldsymbol{z}_t, \boldsymbol{w}_t, \boldsymbol{z}_t' \in \mathcal{Z}_t$,

$$\begin{aligned} \gamma(1-\gamma) r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{w}_t) \leq &\gamma\psi(\boldsymbol{z}_t, \boldsymbol{z}_t') + (1-\gamma)\psi(\boldsymbol{w}_t, \boldsymbol{z}_t') \\ &- \psi(\gamma\boldsymbol{z}_t + (1-\gamma)\boldsymbol{w}_t, \boldsymbol{z}_t'), \end{aligned} \tag{A.24}$$

and

$$\frac{r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{w}_t)}{\|\boldsymbol{z}_t, \boldsymbol{w}_t\|} \to 0, \ as \ \|\boldsymbol{z}_t - \boldsymbol{w}_t\| \to 0, \ \forall \boldsymbol{z}_t' \in \mathcal{Z}_t. \tag{A.25}$$

Substituting the definition of the Nikaido-Isoda function $\psi(\cdot,\cdot)$ into the RHS of inequality (A.24), we have

$$\begin{aligned}
&\frac{\gamma\psi(\boldsymbol{z}_t,\boldsymbol{z}_t')+(1-\gamma)\psi(\boldsymbol{w}_t,\boldsymbol{z}_t')-\psi(\gamma\boldsymbol{z}_t+(1-\gamma)\boldsymbol{w}_t,\boldsymbol{z}_t')}{\gamma(1-\gamma)}\\
&=\sum_{i=1}^{M}\big[\eta\alpha(\bar{z}_{-i,t}-\bar{w}_{-i,t})(z_{i,t}-w_{i,t})\\
&\qquad+(\eta\alpha-\frac{\nu_i}{2\sigma_{i,t}^2})(z_{i,t}-w_{i,t})^2\big]\\
&=\sum_{i=1}^{M}\big[\eta\alpha\big(\sum_{\substack{j=1\\j\neq i}}^{M}(z_{j,t}-w_{j,t})\big)(z_{i,t}-w_{i,t})\\
&\qquad+(\eta\alpha-\frac{\nu_i}{2\sigma_{i,t}^2})(z_{i,t}-w_{i,t})^2\big]\\
&=\eta\alpha\big[\sum_{i=1}^{M}(z_{i,t}-w_{i,t})\big]^2-\sum_{i=1}^{M}\frac{\nu_i}{2\sigma_{i,t}^2}(z_{i,t}-w_{i,t})^2\\
&\geq\eta\alpha\big[\sum_{i=1}^{M}(z_{i,t}-w_{i,t})\big]^2-\big(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2}\big)\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2^2,
\end{aligned}\tag{A.26}$$

where $\|\cdot\|_2$ denotes the l_2-norm. Let

$$\begin{aligned}
&r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t,\boldsymbol{w}_t)\\
&=\eta\alpha\big[\sum_{i=1}^{M}(z_{i,t}-w_{i,t})\big]^2-\big(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2}\big)\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2^2.
\end{aligned}\tag{A.27}$$

Based on the above derivation, the residual term satisfies (A.24).

Divide both sides of (A.27) by $\|\boldsymbol{z}_t-\boldsymbol{w}_t\|_2$, we obtain

$$\begin{aligned}
&\frac{r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t,\boldsymbol{w}_t)}{\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2}\\
&=\eta\alpha\frac{\big[\sum_{i=1}^{M}(z_{i,t}-w_{i,t})\big]^2}{\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2}-\big(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2}\big)\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2\\
&\overset{(a)}{\leq}\eta\alpha M\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2-\big(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2}\big)\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2,
\end{aligned}\tag{A.28}$$

where (a) holds due to the Cauchy-Schwarz inequality. On the other hand, there is

$$\frac{r_{\boldsymbol{z}_t'}(\boldsymbol{z}_t,\boldsymbol{w}_t)}{\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2}\geq-\big(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2}\big)\left\|\boldsymbol{z}_t-\boldsymbol{w}_t\right\|_2.\tag{A.29}$$

Therefore, $r_{\boldsymbol{z}'_t}(\boldsymbol{z}_t, \boldsymbol{w}_t)/\left\|\boldsymbol{z}_t - \boldsymbol{w}_t\right\|_2 \to 0$ as $\left\|\boldsymbol{z}_t - \boldsymbol{w}_t\right\|_2 \to 0$ since both of its upper bound (A.28) and lower bound (A.29) go to 0 as $\left\|\boldsymbol{z}_t - \boldsymbol{w}_t\right\|_2 \to 0$. In this case, the Nikaido-Isoda function is weakly convex.

We next show the Nikaido-Isoda function $\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t)$ is weakly concave on the joint power request set $\mathcal{Z}_t$ with respect to the second argument. According to Definition 6, we need to find a residual term $\mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{w}'_t)$ such that the following condition holds for all $\boldsymbol{z}_t, \boldsymbol{w}'_t, \boldsymbol{z}'_t \in \mathcal{Z}_t$

$$\begin{aligned}\gamma(1-\gamma)\mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{w}'_t) \geq \gamma\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t) + (1-\gamma)\psi(\boldsymbol{z}_t, \boldsymbol{w}'_t) \\ - \psi(\boldsymbol{z}_t, \gamma\boldsymbol{z}'_t + (1-\gamma)\boldsymbol{w}'_t),\end{aligned} \tag{A.30}$$

and

$$\frac{\mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{w}'_t)}{\left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\|} \to 0,\ as\ \left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\| \to 0,\ \forall \boldsymbol{z}_t \in \mathcal{Z}_t. \tag{A.31}$$

Substituting the definition of the Nikaido-Isoda function $\psi(\cdot,\cdot)$ into the RHS of inequality (A.30), we have

$$\begin{aligned}&\frac{\gamma\psi(\boldsymbol{z}_t, \boldsymbol{z}'_t) + (1-\gamma)\psi(\boldsymbol{z}_t, \boldsymbol{w}'_t) - \psi(\boldsymbol{z}_t, \gamma\boldsymbol{z}'_t + (1-\gamma)\boldsymbol{w}'_t)}{\gamma(1-\gamma)} \\ &= \sum_{i=1}^{M}\left(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha\right)(z'_{i,t} - w'_{i,t})^2 \\ &\leq \left(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha\right)\left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\|_2^2.\end{aligned} \tag{A.32}$$

Let

$$\mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{w}'_t) = \left(\max_{1\leq i\leq M}\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha\right)\left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\|_2^2. \tag{A.33}$$

Therefore, the residual term satisfies the inequality (A.30) and $\mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{w}'_t)/\left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\|_2 \to 0$ as $\left\|\boldsymbol{z}'_t - \boldsymbol{w}'_t\right\|_2 \to 0$. □

We next show that condition (4) is satisfied, i.e., the residual term $r_{\boldsymbol{z}'_t}(\boldsymbol{z}_t, \boldsymbol{w}_t)$ is uniformly continuous on $\boldsymbol{z}'_t$ over $\mathcal{Z}_t$. Given a power request sequence $\boldsymbol{z}'_{t'} \in \mathcal{Z}_t$ such that $\left\|\boldsymbol{z}'_t - \boldsymbol{z}'_{t'}\right\| \to 0$, then

$$\left\|r_{\boldsymbol{z}'_t}(\boldsymbol{z}_t, \boldsymbol{w}_t) - r_{\boldsymbol{z}'_{t'}}(\boldsymbol{z}_t, \boldsymbol{w}_t)\right\| \to 0,\ \forall \boldsymbol{z}_t, \boldsymbol{w}_t \in \mathcal{Z}_t, \tag{A.34}$$

since the residual term specified in (A.27) actually does not depend on $\boldsymbol{z}'_t$. Thus, according to the definition of uniformly continuity, the residual term $r_{\boldsymbol{z}'_t}(\boldsymbol{z}_t, \boldsymbol{w}_t)$ is uniformly continuous on $\boldsymbol{z}'_t$ over $\mathcal{Z}_t$.

Finally, we show that the above identified residual terms satisfy condition (5), i.e.,

$$r_{\boldsymbol{z}'_t}(\boldsymbol{z}_t, \boldsymbol{z}'_t) - \mu_{\boldsymbol{z}_t}(\boldsymbol{z}'_t, \boldsymbol{z}_t) \geq \beta\left\|\boldsymbol{z}_t - \boldsymbol{z}'_t\right\|,\ \forall \boldsymbol{z}_t, \boldsymbol{z}'_t \in \mathcal{Z}_t, \tag{A.35}$$

where β is a strictly increasing function and $\beta(0) = 0$.

Let the matrix $Q(\boldsymbol{z}_t, \boldsymbol{z}_t) = \psi_{\boldsymbol{z}_t\boldsymbol{z}_t}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t} - \psi_{\boldsymbol{z}_t'\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t}$, where $\psi_{\boldsymbol{z}_t\boldsymbol{z}_t}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t}$ is the Hessian matrix of the Nikaido-Isoda function with respect to the first argument and $\psi_{\boldsymbol{z}_t'\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t}$ is the Hessian matrix of the Nikaido-Isoda function with respect to the second argument, both evaluated at $\boldsymbol{z}_t' = \boldsymbol{z}_t$. According to [83], it is then sufficient to show the matrix $Q(\boldsymbol{z}_t, \boldsymbol{z}_t)$ is positive definite. It is easy to verify that $\psi_{\boldsymbol{z}_t\boldsymbol{z}_t}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t}$ is an $M \times M$ matrix with i-th diagonal element to be $-2(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha)$ and all the off-diagonal elements to be $2\eta\alpha$. Meanwhile, $\psi_{\boldsymbol{z}_t'\boldsymbol{z}_t'}(\boldsymbol{z}_t, \boldsymbol{z}_t')|_{\boldsymbol{z}_t'=\boldsymbol{z}_t}$ is identified as a diagonal matrix with i-th diagonal element to be $2(\frac{\nu_i}{2\sigma_{i,t}^2} - \eta\alpha)$. In this case, $Q(\boldsymbol{z}_t, \boldsymbol{z}_t)$ is an $M \times M$ matrix with i-th diagonal element to be $4\eta\alpha - \frac{2\nu_i}{\sigma_{i,t}^2}$ and all the off-diagonal elements to be $2\eta\alpha$. Let E be an $M \times M$ diagonal matrix with i-th diagonal element to be $2\eta\alpha - \frac{2\nu_i}{\sigma_{i,t}^2}$, and let I be an $M \times M$ identity matrix. Then we have $Q(\boldsymbol{z}_t, \boldsymbol{z}_t) = E + 2\eta\alpha I$. According to Weyl's inequality [89], when $\alpha > \frac{\nu_i}{\eta\sigma_{i,t}^2}, \forall i, t$, we have $\lambda_{min}(Q(\boldsymbol{z}_t, \boldsymbol{z}_t)) \geq \lambda_{min}(E) > 0$, which means $Q(\boldsymbol{z}_t, \boldsymbol{z}_t)$ is positive definite.

In conclusion, if $\alpha > \frac{\nu_i}{\eta\sigma_{i,t}^2}$ for all $1 \leq i \leq M$ and $1 \leq t \leq T$, all six conditions are satisfied and our proposed distributed relaxation algorithm will converge a unique pure-strategy NE.

A.7 Proof of Theorem 5

According to [90], the EM algorithm will have the following convergence rate in case $-Q(\lambda|\lambda')$ is ρ-strongly-convex.

$$\min_{n\in\{1,2,\ldots,k\}} \left\|\lambda_n^p - \lambda_{n-1}^p\right\|^2 \leq \frac{2(\log P(y^p|\lambda^{p^*}) - \log P(y^p|\lambda_0^p))}{\rho k}. \tag{A.36}$$

Let α be any number in the set $[0, 1]$, and $\lambda^\dagger, \lambda^\ddagger \in \Lambda$. By substituting λ with $\alpha\lambda^\dagger + (1 - \alpha)\lambda^\ddagger$ in function $-\langle \ln \lambda, \vec{P}\rangle$, we have the following equation

$$-\langle \ln \lambda, \vec{P}\rangle = -\sum_{l=1}^{L} \vec{P}_l \ln(\alpha\lambda_l^\dagger + (1 - \alpha)\lambda_l^\ddagger), \tag{A.37}$$

where $L = N + N^2 + NM$ is the length of vector λ and $\vec{P}$. Note that the HMM parameter set Λ is composed of different permutations of different probability simplex, which is thus a convex set. In this case, the vector $\alpha\lambda^\dagger + (1 - \alpha)\lambda^\ddagger$ also belongs to the set Λ with each element to be within the range $[0, 1]$. Given the fact that function $-\ln x$ is 1-strongly-

convex in $x \in [0,1]$, the following inequality holds

$$\begin{aligned} -\sum_{l=1}^{L} \vec{P}_l \ln(\alpha\lambda_l^\dagger + (1-\alpha)\lambda_l^\ddagger) \leq \\ -\alpha\sum_{l=1}^{L} \vec{P}_l \ln\lambda_l^\dagger - (1-\alpha)\sum_{l=1}^{L} \vec{P}_l \ln\lambda_l^\ddagger \\ -\frac{\alpha(1-\alpha)}{2}\sum_{l=1]}^{L} \vec{P}_l(\lambda_l^\dagger - \lambda_l^\ddagger)^2. \end{aligned} \tag{A.38}$$

According to Assumption 2, all elements in $\vec{P}$ are lower bounded by η. In this case, we have

$$\begin{aligned} -\langle \ln\left(\alpha\lambda_l^\dagger + (1-\alpha)\lambda_l^\ddagger\right), \vec{P}\rangle \leq -\alpha\langle \ln\lambda^\dagger, \vec{P}\rangle \\ -(1-\alpha)\langle \ln\lambda^\ddagger, \vec{P}\rangle - \frac{\alpha(1-\alpha)\eta}{2}\left\|\lambda^\dagger - \lambda^\ddagger\right\|_2^2. \end{aligned} \tag{A.39}$$

The above inequality shows that function $-\langle \ln\lambda, \vec{P}\rangle$ is η-strongly-convex with respect to vector λ.

Plugging the above results and Assumption 1 into (A.36) finishes the proof.

A.8 Proof of Lemma 6

Note that there are two different probability terms included in (6.12), i.e., $P(y_t^{b_t}|y_{1:t-1}^{1:\bar{N}}, c_{1:t})$ and $P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, y_t^{b_t})$. We further decompose these two terms as following

$$\begin{aligned} &P(y_t^{b_t}|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) \\ &\quad= \sum_{z_t} P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) P(y_t^{b_t}|z_t, y_{1:t-1}^{1:\bar{N}}, c_{1:t}) \\ &\quad\approx \sum_{z_t} P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) P(y_t^{b_t}|z_t, y_{1:t-1}^{1:\bar{N}}, c_{1:t}), \end{aligned} \tag{A.40}$$

$$\begin{aligned} &P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}, y_t^{b_t}) \\ &\quad= \frac{P(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) P(y_t^{b_t}|z_t, y_{1:t-1}^{1:\bar{N}}, c_{1:t})}{P(y_t^{b_t}|y_{1:t-1}^{1:\bar{N}}, c_{1:t})} \\ &\quad\approx \frac{P_{i_t^{1:\bar{N}}}(z_t|y_{1:t-1}^{1:\bar{N}}, c_{1:t}) P(y_t^{b_t}|z_t, y_{1:t-1}^{1:\bar{N}}, c_{1:t})}{P(y_t^{b_t}|y_{1:t-1}^{1:\bar{N}}, c_{1:t})}. \end{aligned} \tag{A.41}$$

At time step t, assume the reference sensor is selected as sensor with index b_t, denote. Given fixed $i_t^{b_t}$ and c_t, $\forall z_t \in \mathcal{Z}$, if there is any $x_t^n \in \mathcal{X}$ that satisfies $z_t = g(x_t^n, i_t^{b_t}, c_t)$,

$P(y_t^{b_t}|z_t)$ can be then calculated according to the following

$$P(y_t^{b_t}|z_t) = P(y_t^{b_t}|x_t^{b_t} = h(z_t, i_t^{b_t}, c_t)), \tag{A.42}$$

where $h(\cdot)$ is the inverse function of $g(\cdot)$ as defined in Remark 2. Otherwise, $P(y_t^{b_t}|z_t) = 0$. Note that $P(y_t^{b_t}|x_t^{b_t})$ is the emission probability of sensor b_t at time t, which is obtained via the supervised learning of the HMM parameters for each single sensor. The equation above indicates that $P(y_t^{b_t}|z_t)$ can be fully specified by the emission probability term of sensor b_t given control action b_t. Also note that $P(y_t^{b_t}|z_t, y_{1:t-1}^{1:\bar{N}}, c_{1:t}) = P(y_t^{b_t}|z_t)$ due to the conditional independent assumptions. In this case, the per-step cost in (6.12) thus can be fully written as a function of any state-action pair (θ_t, b_t) and thus can be approximated by a function of the state-action pair $(\tilde{\theta}_t, b_t)$.

Bibliography

[1] Y. Mo, T. H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, Jan 2012.

[2] G. W. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *IEEE Technology and Society Magazine*, vol. 8, no. 2, pp. 12–16, June 1989.

[3] J. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial intelligence and statistics*, 2012, pp. 1472–1482.

[4] M. Zeifman, "Disaggregation of home energy display data using probabilistic approach," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 23–31, 2012.

[5] Q. Liu, K. M. Kamoto, X. Liu, M. Sun, and N. Linge, "Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 28–37, 2019.

[6] M. Figueiredo, B. Ribeiro, and A. de Almeida, "Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 364–373, 2014.

[7] Y. Kim, E. C. H. Ngai, and M. B. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct 2011, pp. 178–183.

[8] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *2010 First IEEE International Conference on Smart Grid Communications*, Oct 2010, pp. 238–243.

[9] J. M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *2010 IEEE International Conference on Communications Workshops*, May 2010, pp. 1–5.

[10] S. Li, A. Khisti, and A. Mahajan, "Privacy-optimal strategies for smart metering systems with a rechargeable battery," in *2016 American Control Conference (ACC)*, July 2016, pp. 2080–2085.

[11] J. Yao and P. Venkitasubramaniam, "On the privacy-cost tradeoff of an in-home power storage mechanism," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2013, pp. 115–122.

[12] Y. You, Z. Li, and T. J. Oechtering, "Optimal privacy-enhancing and cost-efficient energy management strategies for smart grid consumers," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, 2018, pp. 826–830.

[13] L. Yang, X. Chen, J. Zhang, and H. V. Poor, "Cost-effective and privacy-preserving energy management for smart meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 486–495, Jan 2015.

[14] O. Tan, J. Gómez-Vilardebó, and D. Gündüz, "Privacy-cost trade-offs in demand-side management with storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1458–1469, 2017.

[15] G. Giaconi and D. Gündüz, "Smart meter privacy with renewable energy and a finite capacity battery," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2016, pp. 1–5.

[16] E. Erdemir, P. L. Dragotti, and D. Gündüz, "Privacy-cost trade-off in a smart meter system with a renewable energy source and a rechargeable battery," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2687–2691.

[17] O. Tan, D. Gündüz, and H. V. Poor, "Increasing smart meter privacy through energy harvesting and storage devices," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1331–1341, 2013.

[18] "The EU General Data Protection Regulation," Available online: https://eugdpr.org/.

[19] P. Ferrer-Cid, J. M. Barcelo-Ordinas, J. Garcia-Vidal, A. Ripoll, and M. Viana, "Multisensor data fusion calibration in iot air pollution platforms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3124–3132, 2020.

[20] B. Maag, Z. Zhou, and L. Thiele, "A survey on sensor calibration in air pollution monitoring deployments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4857–4870, 2018.

[21] J. A. Gomes, J. J. P. C. Rodrigues, R. A. L. Rabêlo, N. Kumar, and S. Kozlov, "Iot-enabled gas sensors: Technologies, applications, and opportunities," *Journal of Sensor and Actuator Networks*, vol. 8, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2224-2708/8/4/57

[22] J. A. Gomes, J. J. P. C. Rodrigues, J. Al-Muhtadi, N. Arunkumar, R. A. L. Rabêlo, and V. Furtado, "An iot-based smart solution for preventing domestic co and lpg gas accidents," in *2018 IEEE 10th Latin-American Conference on Communications (LATINCOM)*, 2018, pp. 1–6.

[23] "Why NDIR?" Accessed May. 30, 2022. [Online]. Available: https://senseair.com/knowledge/sensor-technology/technology/why-ndir/

[24] J. Park, H. Cho, and S. Yi, "NDIR CO_2 gas sensor with improved temperature compensation," *Procedia Engineering*, pp. 303–306, 2010.

[25] M. Müller, P. Graf, J. Meyer, A. Pentina, D. Brunner, F. Perez-Cruz, C. Hüglin, and L. Emmenegger, "Integration and calibration of non-dispersive infrared (ndir) CO_2 low-cost sensors and their operation in a sensor network covering switzerland," *Atmospheric Measurement Techniques*, vol. 13, no. 7, pp. 3815–3834, 2020.

[26] C. R. Martin, N. Zeng, A. Karion, R. R. Dickerson, X. Ren, B. N. Turpie, and K. J. Weber, "Evaluation and environmental correction of ambient co 2 measurements from a low-cost ndir sensor," *Atmospheric measurement techniques*, vol. 10, no. 7, pp. 2383–2395, 2017.

[27] "TN-011," SenseAir, Tech. Rep., 2000.

[28] Z. Zhang, Z. Qin, L. Zhu, W. Jiang, C. Xu, and K. Ren, "Toward practical differential privacy in smart grid with capacity-limited rechargeable batteries," 2015. [Online]. Available: https://arxiv.org/abs/1507.03000

[29] M. Backes and S. Meiser, "Differentially private smart metering with battery recharging," in *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 2013, pp. 194–212.

[30] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 837–846, June 2013.

[31] Z. Li, T. J. Oechtering, and M. Skoglund, "Privacy-preserving energy flow control in smart grids," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 2194–2198.

[32] R. R. Avula, T. J. Oechtering, J. Chin, and G. Hug, "Smart meter privacy control strategy including energy storage degradation," in *2019 IEEE Milan PowerTech*, June 2019, pp. 1–6.

[33] Z. Li, T. J. Oechtering, and D. Gündüz, "Smart meter privacy based on adversarial hypothesis testing," in *IEEE International Symposium on Information Theory (ISIT) 2017*, 2017, pp. 774–778.

[34] Z. Li, T. J. Oechtering, and D. Gündüz, "Privacy against a hypothesis testing adversary," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1567–1581, June 2019.

[35] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[36] Z. Guo, Z. J. Wang, and A. Kashani, "Home appliance load modeling from aggregated smart meter data," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 254–262, 2015.

[37] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2016.

[38] W. Kong, Z. Y. Dong, J. Ma, D. J. Hill, J. Zhao, and F. Luo, "An extensible approach for non-intrusive load disaggregation with smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3362–3372, 2018.

[39] X. He, X. Zhang, and C. Kuo, "A distortion-based approach to privacy-preserving metering in smart grids," *IEEE Access*, vol. 1, pp. 67–78, 2013.

[40] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proceedings of the 2011 SIAM international conference on data mining*. SIAM, 2011, pp. 747–758.

[41] D. Mashima and A. Roy, "Privacy preserving disclosure of authenticated energy usage data," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 866–871.

[42] R. R. Avula and T. J. Oechtering, "On design of optimal smart meter privacy control strategy against adversarial map detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5845–5849.

[43] G. Kalogridis, Z. Fan, and S. Basutkar, "Affordable privacy for home smart meters," in *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops*, May 2011, pp. 77–84.

[44] J. X. Chin, T. T. D. Rubira, and G. Hug, "Privacy-protecting energy management unit through model-distribution predictive control," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[45] F. Farokhi and H. Sandberg, "Fisher information as a measure of privacy: Preserving privacy of households with smart meters using batteries," *IEEE Transactions on Smart Grid*, 2017.

[46] W. Saad, Z. Han, H. V. Poor, and T. Basar, "Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 86–105, 2012.

[47] F. Farokhi, A. M. H. Teixeira, and C. Langbort, "Estimation with strategic sensors," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 724–739, 2017.

[48] J. Yao and P. Venkitasubramaniam, "Privacy aware stochastic games for distributed end-user energy storage sharing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 82–95, 2018.

[49] K. Erdayandi, A. Paudel, L. Cordeiro, and M. A. Mustafa, "Privacy-friendly peer-to-peer energy trading: A game theoretical approach," *CoRR*, vol. abs/2201.01810, 2022. [Online]. Available: https://arxiv.org/abs/2201.01810

[50] D. Solomatine, L. M. See, and R. J. Abrahart, "Data-driven modelling: concepts, approaches and experiences," in *Practical hydroinformatics*. Springer, 2009, pp. 17–30.

[51] L. Chen, L. Zhangzhong, W. Zheng, J. Yu, Z. Wang, L. Wang, and C. Huang, "Data-driven calibration of soil moisture sensor considering impacts of temperature: A case study on fdr sensors," *Sensors*, vol. 19, no. 20, p. 4381, 2019.

[52] D. Wang, J. Liu, and R. Srinivasan, "Data-driven soft sensor approach for quality prediction in a refining process," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 11–17, Feb 2010.

[53] L. Dong, Z. Qiao, H. Wang, W. Yang, W. Zhao, K. Xu, G. Wang, L. Zhao, and H. Yan, "The gas leak detection based on a wireless monitoring system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6240–6251, 2019.

[54] T. Wissel, B. Wagner, P. Stüber, A. Schweikard, and F. Ernst, "Data-driven learning for calibrating galvanometric laser scanners," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5709–5717, Oct 2015.

[55] R. Ak, O. Fink, and E. Zio, "Two machine learning approaches for short-term wind speed time-series prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1734–1747, 2016.

[56] B. Chen, W.-A. Zhang, and L. Yu, "Distributed finite-horizon fusion Kalman filtering for bandwidth and energy constrained wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 797–812, 2014.

[57] M. Ghanbari and M. J. Yazdanpanah, "Delay compensation of tilt sensors based on mems accelerometer using data fusion technique," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1959–1966, 2015.

[58] R. Jassemi-Zargani and D. Necsulescu, "Extended Kalman filter-based sensor fusion for operational space control of a robot arm," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 6, pp. 1279–1282, 2002.

[59] T. Denoeux and M. Masson, "Belief functions: theory and applications," in *Proceedings of the 2nd international conference on belief functions*. Springer, 2012, pp. 9–11.

[60] D. Alshamaa, F. Mourad-Chehade, and P. Honeine, "Tracking of mobile sensors using belief functions in indoor wireless networks," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 310–319, 2018.

[61] ——, "Decentralized kernel-based localization in wireless sensor networks using belief functions," *IEEE Sensors Journal*, vol. 19, no. 11, pp. 4149–4159, 2019.

[62] N. Milisavljevic and I. Bloch, "Sensor fusion in anti-personnel mine detection using a two-level belief function model," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 33, no. 2, pp. 269–283, 2003.

[63] E. Lefevre, O. Colot, and P. Vannoorenberghe, "Belief function combination and conflict management," *Information fusion*, vol. 3, no. 2, pp. 149–162, 2002.

[64] C. K. Murphy, "Combining belief functions when evidence conflicts," *Decision support systems*, vol. 29, no. 1, pp. 1–9, 2000.

[65] Y. Deng, W. Shi, Z. Zhu, and Q. Liu, "Combining belief functions based on distance of evidence," *Decision support systems*, vol. 38, no. 3, pp. 489–493, 2004.

[66] Z. Zhang, T. Liu, D. Chen, and W. Zhang, "Novel algorithm for identifying and fusing conflicting data in wireless sensor networks," *Sensors*, vol. 14, no. 6, pp. 9562–9581, 2014.

[67] Y. Yang and D. Han, "A new distance-based total uncertainty measure in the theory of belief functions," *Knowledge-Based Systems*, vol. 94, pp. 114–123, 2016.

[68] A. Jousselme, B. Grenier, and É. Bossé, "A new distance between two bodies of evidence," *Information fusion*, vol. 2, no. 2, pp. 91–101, 2001.

[69] C. Villani, *The Wasserstein distances*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 93–111. [Online]. Available: https://doi.org/10.1007/978-3-540-71050-9_6

[70] V. Krishnamurthy, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge University Press, 2016.

[71] M. Cao, J. Chen, and J. Wang, "A novel vehicle tracking method for cross-area sensor fusion with reinforcement learning based gmm," in *2020 American Control Conference (ACC)*, 2020, pp. 442–447.

[72] R. Sutton and A. Barto, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 2, no. 4.

[73] K. Tan, K. Tan, T. Lee, S. Zhao, and Y. Chen, "Autonomous robot navigation based on fuzzy sensor fusion and reinforcement learning," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, 2002, pp. 182–187.

[74] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[75] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[76] D. F. Swinehart, "The beer-lambert law," *Journal of Chemical Education*, vol. 39, no. 7, p. 333, 1962. [Online]. Available: https://doi.org/10.1021/ed039p333

[77] J. Tsitsiklis, "NP-hardness of checking the unichain condition in average cost mdps," *Operations research letters*, vol. 35, no. 3, pp. 319–323, 2007.

[78] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Springer US, 2015.

[79] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[80] F. Melo and M. Ribeiro, "Q-learning with linear function approximation," in *International Conference on Computational Learning Theory*. Springer, 2007, pp. 308–322.

[81] J. Kolter and M. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, no. Citeseer, 2011, pp. 59–62.

[82] G. E. Rahi, S. R. Etesami, W. Saad, N. B. Mandayam, and H. V. Poor, "Managing price uncertainty in prosumer-centric energy trading: A prospect-theoretic stackelberg game approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 702–713, 2019.

[83] J. Krawczyk and S. Uryasev, "Relaxation algorithms to find nash equilibria with economic applications," *Environmental Modeling & Assessment*, vol. 5, no. 1, pp. 63–73, 2000.

[84] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[85] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[86] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[87] "Drl environment for sensor calibration." Accessed May. 30, 2022. [Online]. Available: https://github.com/tmacyouy/DRL-Environment-Sensor-Calibration.git

[88] J. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, pp. 520–534, 1965.

[89] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[90] R. Kumar and M. Schmidt, "Convergence rate of expectation-maximization," in *10th NIPS Workshop on Optimization for Machine Learning*, 2017, p. 98.